SAPIENZA
UNIVERSITÀ DI ROMA

Abstraction in Ontology-based Data Management

Department of Computer, Control and Management Engineering "Antonio Ruberti"

Philosophiæ Doctor in Engineering in Computer Science – XXXII Cycle

Candidate

Gianluca Cima
ID number 1492955

Thesis Advisor

Prof. Maurizio Lenzerini

Co-Advisor

Prof. Giuseppe De Giacomo

A thesis submitted in partial fulfilment of the requirements for the degree of Philosphiæ Doctor in Engineering in Computer Science

A.Y. 2019/2020

Thesis defended on December 07, 2020
in front of a Board of Examiners composed of:

Prof. Maria Costabile (chairperson)

Prof. Matteo Matteucci

Prof. Magdalena Ortiz (Examiner of a different EU State from where the thesis is discussed as required for the additional certificate/mention of *Doctor Europaeus*)

List of Reviewers:

Prof. Meghyn Bienvenu

Prof. Axel Polleres

Abstraction in Ontology-based Data Management
Philosophiæ Doctor Thesis. Sapienza – University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Version: December 9, 2020

Author's Website: https://sites.google.com/view/gianlucacima

Author's Email: gianlucacima@gmail.com

Acknowledgments

First of all, I would like to express my sincere gratitude to my advisor, Maurizio Lenzerini, for his constant support and enthusiastic encouragement during all my PhD journey. Without his help this work would never have been accomplished.

I feel very luck to have been part of the “Data Management and Service-Oriented Computing” research group at the “Department of Computer, Control and Management Engineering” of “Sapienza University of Rome”. The working environment is very stimulating, and I thank all the members of the group I have worked with. In particular, I am very grateful to Antonella Poggi for her invaluable advices and the several fruitful and stimulating discussions we have had together.

I am also thankful to the members of the “Knowledge Representation and Reasoning” research group at the “Department of Computer Science” of “University of Oxford”, which I have joined for six months as a visiting PhD student. I have really appreciated my staying, and I am pleased with the work we have done together.

Finally, I would like to thank all the people whose company and friendship I have enjoyed over these years, and all my family members for their abiding affection. A special thank goes to my loving girlfriend, Giulia Maria, for her continuous motivation and the way in which she pushes me to give my best in all that I do.

Gianluca Cima
Italy, July 2020

Abstract

In many aspects of our society there is growing awareness and consent on the need for data-driven approaches that are resilient, transparent, and fully accountable. But in order to fulfil the promises and benefits of a data-driven society, it is necessary that the data services exposed by the organisations' information systems are well-documented, and their semantics is clearly specified. Effectively documenting data services is indeed a crucial issue for organisations, not only for governing their own data, but also for interoperation purposes.

In this thesis, we propose a new approach to automatically associate formal semantic descriptions to data services, thus bringing them into compliance with the FAIR guiding principles, i.e., make data services automatically Findable, Accessible, Interoperable, and Reusable (FAIR). We base our proposal on the Ontology-based Data Management (OBDM) paradigm, where a domain ontology is used to provide a semantic layer mapped to the data sources of an organisation, thus abstracting from the technical details of the data layer implementation.

The basic idea is to characterise or explain the semantics of a given data service expressed as query over the source schema in terms of a query over the ontology. Thus, the query over the ontology represents an *abstraction* of the given data service in terms of the domain ontology through the mapping, and, together with the elements in the vocabulary of the ontology, such abstraction forms a basis for annotating the given data service with suitable metadata expressing its semantics.

We illustrate a formal framework for the task of automatically produce a semantic characterisation of a given data service expressed as a query over the source schema. The framework is based on three semantically well-founded notions, namely *perfect*, *sound*, and *complete source-to-ontology rewriting*, and on two associated basic computational problems, namely *verification* and *computation*. The former verifies whether a given query over the ontology is a perfect (respectively, sound, complete) source-to-ontology rewriting of a given data service expressed as a query over the source schema, whereas the latter computes one such rewriting, provided it exists. We provide an in-depth complexity analysis of these two computational problems in a very general scenario which uses languages amongst the most popular considered in the literature of managing data through an ontology. Furthermore, since we study also cases where the target query language for expressing source-to-ontology rewritings allows inequality atoms, we also investigate the problem of answering queries with inequalities over lightweight ontologies, a problem that has been rarely addressed. In another direction, we study and advocate the use of a non-monotonic target query language for expressing source-to-ontology rewritings. Last but not least, we outline a detailed related work, which illustrates how the results achieved in this thesis notably contributes to new results in the Semantic Web context, in the relational database theory, and in view-based query processing.

List of Publications

The results presented in this thesis are part of the findings for my doctoral research work, and most of them are already published in the following scientific publications:

- Gianluca Cima, Maurizio Lenzerini, and Antonella Poggi. Answering Conjunctive Queries with Inequalities in *DL-Lite*$_{\mathcal{R}}$. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2020)*, pages 2782–2789, 2020.

- Gianluca Cima, Domenico Lembo, Riccardo Rosati, and Domenico Fabio Savo. Controlled Query Evaluation in Description Logics through Instance Indistinguishability. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI 2020)*, pages 1791–1797, 2020.

- Gianluca Cima, Maurizio Lenzerini, and Antonella 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)*, pages 243–252, 2020.

- Gianluca Cima, Domenico Lembo, Lorenzo Marconi, Riccardo Rosati, and Domenico Fabio Savo. Controlled Query Evaluation in Ontology-based Data Access. In *Proceedings of the Nineteenth International Semantic Web Conference (ISWC 2020)*, volume 12506 of Lecture Notes in Computer Science, pages 128–146, 2020.

- Federico Croce, Gianluca Cima, Maurizio Lenzerini, and Tiziana Catarci. Ontology-based Explanation of Classifiers. In *Proceedings of the Workshops of the EDBT/ICDT 2020 Joint Conference*, volume 2578 of CEUR Electronic Workshop Proceedings, 2020.

- Gianluca Cima, Domenico Lembo, Riccardo Rosati, and Domenico Fabio Savo. Controlled Query Evaluation in Description Logics through Instance Indistinguishability (Extended Abstract). In *Proceedings of the Thirty-Third International Workshop on Description Logics (DL 2020)*, volume 2663 of CEUR Electronic Workshop Proceedings, 2020.

- Gianluca Cima, Maurizio Lenzerini, and Antonella Poggi. Semantic Characterization of Data Services through Ontologies. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI 2019)*, pages 1647–1653, 2019.

- Gianluca Cima, Charalampos Nikolaou, Egor V. Kostylev, Mark Kaminski, Bernardo Cuenca Grau, and Ian Horrocks. Bag Semantics of *DL-Lite* with Functionality Axioms. In *Proceedings of the Eighteenth International Semantic Web Conference (ISWC 2019)*, volume 11778 of Lecture Notes in Computer Science, pages 128–144, 2019.

- Gianluca Cima, Charalampos Nikolaou, Egor V. Kostylev, Mark Kaminski, Bernardo Cuenca Grau, and Ian Horrocks. Bagging the *DL-Lite* Family Further. In *Proceedings of the Thirty-Second International Workshop on Description Logics (DL 2019)*, volume 2373 of CEUR Electronic Workshop Proceedings, 2019.

- Gianluca Cima, Federico Croce, Maurizio Lenzerini, Antonella Poggi, and Elian Toccacieli. On Queries with Inequalities in $\mathit{DL\text{-}Lite}_{\mathcal{R}}^{\neq}$. In *Proceedings of the Thirty-Second International Workshop on Description Logics (DL 2019)*, volume 2373 of CEUR Electronic Workshop Proceedings, 2019.

- Gianluca Cima, Maurizio Lenzerini, and Antonella Poggi. Reverse Engineering of Data Services. In *Proceedings of the Twenty-Seventh Italian Symposium on Advanced Database Systems (SEBD 2019)*, volume 2400 of CEUR Electronic Workshop Proceedings, 2019.

- Gianluca Cima, Maurizio Lenzerini, and Antonella Poggi. Exploiting Ontologies for Explaining Data Sources Semantics. In *Proceedings of Discussion and Doctoral Consortium Papers of the Eighteenth International Conference of the Italian Association for Artificial Intelligence (DDC@AI*IA 2019)*, volume 2495 of CEUR Electronic Workshop Proceedings, pages 33–35, 2019.

- Gianluca Cima, Giuseppe De Giacomo, Maurizio Lenzerini, and Antonella Poggi. On the `SPARQL` Metamodeling Semantics Entailment Regime for `OWL 2 QL` ontologies. In *Proceedings of the Seventh International Conference on Web Intelligence, Mining and Semantics (WIMS 2017)*, pages 10:1-10:6, 2017.

- Gianluca Cima, Maurizio Lenzerini, and Antonella Poggi. Semantic Technology for Open Data Publishing. In *Proceedings of the Seventh International Conference on Web Intelligence, Mining and Semantics (WIMS 2017)*, pages 1:1, 2017.

- Gianluca Cima. Preliminary results on Ontology-based Open Data Publishing. In *Proceedings of the Thirtieth International Workshop on Description Logics (DL 2017)*, volume 1879 of CEUR Electronic Workshop Proceedings, 2017.

- Gianluca Cima, Giuseppe De Giacomo, Maurizio Lenzerini, and Antonella Poggi. Querying `OWL 2 QL` Ontologies under the `SPARQL` Metamodeling Semantics Entailment Regime. In *Proceedings of the Twenty-Fifth Italian Symposium on Advanced Database Systems (SEBD 2017)*, volume 2037 of CEUR Electronic Workshop Proceedings, page 165, 2017.

Contents

Chapter 1

Introduction

In many aspects of our society there is growing awareness and consent on the need for data-driven approaches that are resilient, transparent, and fully accountable. It is therefore not surprising that the architecture of many modern Information Systems is based on *data services*, i.e., services deployed on top of data stores, other services, and/or applications to encapsulate a wide range of data-centric operations [Carey *et al.*, 2012]. Data services are also used to handle the programming logic for data virtualisation in a cloud-hosted data storage infrastructure, so as to delegate most administrative tasks to the cloud infrastructure, and effectively realising the idea of *data-as-a-service* [Machan, 2009]. Furthermore, since data may be obtuse, disorganised, and may not make much sense to most potential users, in order to get value from them, it is reasonable to resort to data services built on top of massive amount of raw data.

However, in order to fulfil the promises and benefits of a *data-driven society* [Pentland, 2013], it is of vital importance to well document and clearly specify the semantics of data services. Effectively documenting data services is indeed a crucial issue for organisations, not only for governing their own data, that often grow rapidly in the current Big Data era [Chen *et al.*, 2014], but also for interoperation purposes. Most current techniques manually associate *APIs (Application Programming Interfaces)* to data services, and describe their intended meaning with ad-hoc methods, often using natural language or complex metadata[1] [Zheng *et al.*, 2013]. This is clearly insufficient since such description of data services lack support for a formal semantics, and ergo, they are limited to human consumption only. Contrariwise, data service consumers need access to enhanced metadata, which are both machine-readable and human-readable. These metadata are essential to integrate entities returned from different data services and/or to understand the relationships between various data services, so as to be able to formulate queries and navigate between sets of entities.

In this thesis, we propose a new approach, whose goal is to *automatically* associate formal semantic descriptions to data services, thus bringing them into compliance with the FAIR guiding principles [Wilkinson *et al.*, 2016], i.e., make data services automatically Findable, Accessible, Interoperable, and Reusable (FAIR).

The proposal of the thesis is based on the *Ontology-based Data Management*

[1] As defined in [Duval *et al.*, 2002], metadata are “structured data about data”.

(OBDM[2]) paradigm [Lenzerini, 2011], which is an advanced approach to semantic data integration [De Giacomo *et al.*, 2018] experimented and used in the practice in the last years (see, e.g., [Antonioli *et al.*, 2014; Kharlamov *et al.*, 2017]). OBDM is a promising attempt to give principles and techniques to effectively govern even modern, complex information systems by providing a unified access to data. An OBDM specification consists of a triple $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, where $\mathcal{O}$ is an *ontology* expressed in a specific Description Logic language, $\mathcal{S}$, called *source schema*, is the schema of the data sources forming the data layer of an information system, and $\mathcal{M}$ is a *mapping* between the source schema and the ontology, i.e., an explicit representation of the correspondence between the data sources and the elements of the ontology. The ontology is a formal logic-based representation of the underlying domain that gives a high-level view of the information contained in the data sources. Thus, the OBDM paradigm provides a means for managing data through the lens of an ontology [Lenzerini, 2018], and enables the application of Knowledge Representation and Reasoning principles and techniques to various data management tasks.

But how can we automatically produce a semantic characterisation of a data service, having an OBDM specification available? The idea is to exploit a novel reasoning task over the OBDM specification, which we call *abstraction*, that works as follows: given a data service expressed as a query $q_{\mathcal{S}}$ over the source schema, automatically derive a query $q_{\mathcal{O}}$ over the ontology that describes "at best" the data service $q_{\mathcal{S}}$ with respect to the underlying OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$. Thus, $q_{\mathcal{O}}$ represents an abstraction of the data service represented by $q_{\mathcal{S}}$ in terms of the domain ontology $\mathcal{O}$ through the mapping $\mathcal{M}$. In this way, the query expression $q_{\mathcal{O}}$, together with the elements in the vocabulary of the ontology $\mathcal{O}$, form a basis for annotating the data service represented by the query $q_{\mathcal{S}}$ with suitable metadata expressing its semantics. The next example illustrates this idea.

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

- $\mathcal{O} = \{\ \mathsf{IndependentPolitician} \sqsubseteq \mathsf{Politician}, \exists \mathsf{HasTutor}^- \sqsubseteq \mathsf{Professor}\ \}$
- $\mathcal{S} = \{\ s_1, s_2, s_3\ \}$
- $\mathcal{M} = \{\ m_1, m_2, m_3, m_4\ \}$, where:

$$\begin{array}{lrcl} m_1: & s_1(x) & \rightarrow & \mathsf{Professor}(x), \\ m_2: & s_1(x) & \rightarrow & \mathsf{Politician}(x), \\ m_3: & s_2(x_1, x_2) & \rightarrow & \mathsf{HasTutor}(x_1, x_2), \\ m_4: & s_3(x) & \rightarrow & \mathsf{IndependentPolitician}(x). \end{array}$$

Let the data service be expressed as the logical query $q_{\mathcal{S}} = \{(x) \mid (s_1(x)) \vee (\exists y. s_2(y, x) \wedge s_3(x))\}$ over the source schema $\mathcal{S}$. Conceivably, by inspecting the mapping assertions in $\mathcal{M}$ and the ontology assertions in $\mathcal{O}$, one can argue that the query $q_{\mathcal{O}}$ over the ontology $\mathcal{O}$ that describes at best the data service $q_{\mathcal{S}}$ with respect to the OBDM specification Σ is $q_{\mathcal{O}} = \{(x) \mid \mathsf{Professor}(x) \wedge \mathsf{Politician}(x)\}$. $\square$

[2]Throughout the thesis, it is preferred the usage of the acronym OBDM rather than its similar *OBDA*, which stands for *Ontology-based Data Access* [Poggi *et al.*, 2008], because data access is just one aspect, although one of the most important, of the more general notion of data management.

As testified by [Poggi *et al.*, 2008; Calvanese *et al.*, 2009; Bienvenu, 2016; Xiao *et al.*, 2018; Ortiz, 2018] (and references therein), most of, if not all, the literature about managing data sources through an ontology deals with users' queries expressed over the ontology, and studies the problem of finding a so-called *ontology-to-source rewriting*, i.e., a query over the source schema that, once executed over the data, provides the answers to the original query. Here, the problem is reversed, because we start with a source query $q_{\mathcal{S}}$ over the source schema, and we aim at deriving a corresponding query $q_{\mathcal{O}}$ over the ontology $\mathcal{O}$, which we call a *source-to ontology rewriting*, that is as much close as possible to $q_{\mathcal{S}}$, taking into account the ontology and the mapping. Thus, we deal with a sort of reverse engineering problem, which is novel in the investigation of both OBDM and data integration.

This new notion of source-to-ontology rewriting is also useful in a context strictly related to data services, namely *open data publishing*. In recent years, both public and private organisations have been faced with the issue of publishing *open data*[3], in particular with the goal of providing data consumers with suitable information to capture the semantics of their published datasets. Current practices for publishing open data, however, focus essentially on providing extensional information (often in very simple forms, such as CSV[4] files), and they carry out the task of documenting data mostly by using metadata expressed in natural languages, or in terms of record structures. As a consequence, the semantics of datasets is not formally expressed in a machine-readable form. Only few recent proposals (see, e.g., [Rashid *et al.*, 2020]) provide methodologies to associate formal semantics to datasets by means of machine-readable metadata.

When an OBDM specification is available in an organisation there is an obvious method, called *top-down*, to publish high-quality, semantically annotated open data that are compliant with the W3C[5] Linked Open Data (LOD) principles [Bizer *et al.*, 2009]: (*i*) express the dataset to be published in terms of a `SPARQL` query over the ontology, (*ii*) compute the certain answers to the query, and (*iii*) publish the result of the certain answer computation, using the query expression and the elements in the vocabulary of the ontology as a basis for annotating the dataset with suitable metadata expressing its semantics. Unfortunately, in many organisations (for instance, in Public Administration) IT employees are not yet ready to formulate `SPARQL` queries, rather they may be tempted to directly publish a dataset as the result of the evaluation of a query $q_{\mathcal{S}}$ (over the source schema) over the data of the information system, in any structured form representing it. In order to publish publish both the content and the semantics of the dataset, it is possible to follow a method, called *bottom-up*, that first derive a source-to-ontology rewriting $q_{\mathcal{O}}$ of $q_{\mathcal{S}}$, and then, using $q_{\mathcal{O}}$, continues with the steps (*ii*) and (*iii*) of the top-down approach.

Besides semantic characterisations of data services and open data publishing, we point out that the reasoning task of abstraction is relevant in other plethora of application scenarios, as for example:

[3]According to the Organisation for Economic Co-operation and Development (OECD), *open data* are "data that can be used by anyone without technical or legal restrictions. The use encompasses both access and reuse" [OECD, 2015].

[4]Comma-Separated Values: `https://tools.ietf.org/html/rfc4180`

[5]World Wide Web Consortium: https://www.w3.org/

- *Source profiling*: Source profiling [Abedjan *et al.*, 2017; Abedjan *et al.*, 2018] is a very general term that has to do with the analysis of raw data for the purpose of understanding the source contents. The task of abstraction arguably provides a semantic-based approach to source profiling, in particular for describing the structure and the content of a data source in terms of the business vocabulary.

- *Updating*: As noted in [Lutz *et al.*, 2018], the concept of *realization* of source queries, similar to one of the notions studied here, can be used to check whether the mapping provides the right coverage for expressing the relevant data services at the ontology level. If this is not the case, then, probably, the existing data sources and/or the ontology need to be updated [Lembo *et al.*, 2017].

- *Explanation of classifiers*: Understanding and explaining the decisions made by *machine learning* algorithms is widely recognized as a very important task for wide and safe adoption of machine learning and data mining technologies (see, e.g., [European Union, Parliament and Council, 2016; Goodman and Flaxman, 2017]), especially in high-risk domains, and in dealing with bias. The task of abstraction also moves towards this direction. As an example of its potential usefulness in the *explainable machine learning* field, suppose to acquire the outcome of a *binary classifier* over tuples of data sources in the information system, and that an OBDM specification is also available. Then, it is possible to semantically describe the choices taken by such a classifier by means of a query over the domain ontology, and therefore in terms of the elements in the vocabulary of this latter. For instance, as a naive criteria for the semantic description of the classifier, one may require that the answers to the query include all the tuples classified positively, and none of the tuples classified negatively. For a more detailed discussion on this topic, the reader is referred to [Croce *et al.*, 2020].

1.1 Contributions of the Thesis

This thesis mainly addresses the topic of abstraction in OBDM. The principal contributions can be summarised as follows:

I. We present a formal framework for the reasoning task of abstraction in OBDM. In particular, three semantically well-founded notions are introduced, namely *perfect*, *sound*, and *complete source-to-ontology rewriting*, and two basic computational problems are defined, namely *verification* and *computation*. The former verifies whether a given query $q_{\mathcal{O}}$ over the ontology is a perfect (respectively, sound, complete) source-to-ontology rewriting of a data service expressed as a query $q_{\mathcal{S}}$ over the source schema, whereas the latter computes one such source-to-ontology rewriting, provided it exists.

II. Although the ideal notion is the one of perfect source-to-ontology rewriting, we show that there are cases where, with the current OBDM specification, no query over the ontology can precisely characterise the data service at hand. Thus, two further notions are introduced, namely *maximally sound* and *minimally* *complete source-to-ontology rewriting*, which intuitively aim at approximating the perfect source-to-ontology rewriting of a data service at best, with the goal of either precision (sound rewriting), or recall (complete rewriting).

III. Before of delving into the computational problems introduced in the framework, we provide a thorough analysis of the implications that the presence of inequalities in queries has in the context of lightweight ontologies, which is a problem that has been rarely addressed. This is necessary because we do study also cases where the target query language for expressing source-to-ontology rewritings allows inequality atoms. In particular, we concentrate on the problem of answering conjunctive queries with inequalities (CQ$^{\neq}$s) and unions thereof (UCQ$^{\neq}$s) over *DL-Lite*$_{\mathcal{R}}$ *knowledge bases* (i.e., pairs of ontology and ABox assertions), both with and without the unique name assumption (UNA). Since it is known that the problem is in general undecidable, we explore two alternative strategies for recovering decidability, and especially tractability:

- The first strategy is to weaken the query language by restricting the application of the inequality predicate to either individuals or distinguished variables (variables representing output values) only. The resulting query language is called “UCQs with bounded inequalities” (UCQ$^{\neq,b}$s).
- The second strategy is to weaken the ontology language, so as to eliminate all the constructs introducing incomplete information resulting from existentially quantified assertions in the ontology. The outcome is a sublanguage of *DL-Lite*$_{\mathcal{R}}$, called *DL-Lite*$^{\neg}_{\mathsf{RDFS}}$.

When the UNA is adopted, we prove that both the problems of answering UCQ$^{\neq,b}$s over *DL-Lite*$_{\mathcal{R}}$ knowledge bases and answering UCQ$^{\neq}$s over *DL-Lite*$^{\neg}_{\mathsf{RDFS}}$ knowledge bases are a straightforward generalisation of the well-known problem of answering union of conjunctive queries (UCQs) over *DL-Lite*$_{\mathcal{R}}$ knowledge bases, i.e., the problems are still in AC^0 in *data complexity* (i.e., with respect to the size of the ABox only) and NP-complete in *combined complexity* (i.e., with respect to the size of the whole input, including the query). Afterwards, we concentrate on the case when the UNA is not adopted.

For the case of (U)CQ$^{\neq,b}$s, we show that answering CQ$^{\neq,b}$s over *DL-Lite*$^{\neq}_{\mathcal{R}}$ knowledge bases has the same computational complexity of the UCQ case, i.e., it is in AC^0 in data complexity and NP-complete in combined complexity. However, perhaps surprisingly, answering UCQ$^{\neq,b}$s over *DL-Lite*$_{\mathcal{R}}$ knowledge bases is Π^p_2-complete in combined complexity. Thus, unless NP = coNP, the presence of union makes the problem of answering queries with inequalities over *DL-Lite*$_{\mathcal{R}}$ knowledge bases significantly different from the UCQ case.

For the case of *DL-Lite*$^{\neg}_{\mathsf{RDFS}}$, we show that answering UCQ$^{\neq}$s is decidable, and in particular coNP-complete in data complexity, and Π^p_2-complete in combined complexity. We also investigate if the number of inequalities in each disjunct plays a role in falling into intractability. We answer positively to this question, by showing that if the query has at most one inequality per disjunct, answering UCQ$^{\neq}$s is PTime-complete in data complexity, and NP-complete in combined complexity, while it is coNP-hard in data complexity if the query is conjunctive and has at most two inequalities. We also show that going from one to two inequalities causes the jump from NP-hardness to Π_2^p-hardness in combined complexity for UCQ$^{\neq}$s, and we conjecture that this holds already for CQ$^{\neq}$s.

We argue that the above results considerably improve our understanding of the implication that the presence of inequalities in queries has in the context of lightweight ontologies. In particular, to the best of our knowledge, our investigation on $DL\text{-}Lite_{\mathsf{RDFS}}^{\neg}$ provides the first results on reasoning with inequalities when querying $DL\text{-}Lite_{\mathsf{RDFS}}$ knowledge bases.

IV. The mentioned results on answering CQ$^{\neq,b}$s over $DL\text{-}Lite_{\mathcal{R}}$ knowledge bases allows us to improve the current state-of-the-art on the problem of answering UCQ$^{\neq}$s posed over `OWL 2 QL` knowledge bases interpreted under the *Direct Semantics Entailment Regime* (DSER), i.e., the regime usually adopted in the Semantic Web scenarios that slightly differs from the classical *First Order Logic* (FOL) semantics. In particular, we prove that answering UCQ$^{\neq}$s over `OWL 2 QL` knowledge bases interpreted under DSER has the same computational complexity of the UCQ case under FOL, i.e., the problem is in AC^0 in data complexity and NP-complete in combined complexity.

V. The results on reasoning with inequalities over $DL\text{-}Lite_{\mathsf{RDFS}}^{\neg}$ knowledge bases implies new results on containment of queries with inequalities in the database theory setting. Specifically, we prove that the containment problem for UCQ$^{\neq}$s is still Π_2^p-hard in general (and therefore Π_2^p-complete) and coNP-hard (and therefore coNP-complete) when the containing query is assumed to be fixed, even if the contained query is a CQ and the containing query is a UCQ$^{\neq}$ with at most two inequality atoms. Furthermore, we prove that by allowing at most one inequality to occur in UCQ$^{\neq}$s makes the computational complexity of the containment problem for UCQ$^{\neq}$s falling from Π_2^p-complete down to NP-complete and, when the containing query is assumed to be fixed, from coNP-complete down to PTime-complete. To the best of our knowledge, this is the first investigation on how the number of inequality atoms affects the computational complexity of the containment problem for UCQ$^{\neq}$s.

VI. We study both the verification, and the computation problem for complete, sound, and perfect source-to-ontology rewritings in a very general scenario which uses languages amongst the most popular considered in the literature: (*i*) the setting for OBDM specifications is such that the ontology language is $DL\text{-}Lite_{\mathcal{R}}$, the source schemas do not have integrity constraints, and each mapping assertion maps a conjunctive query (CQ) over the source to a CQ over the ontology (GLAV assertion), and (*ii*) the query language for expressing both the data service and the source-to-ontology rewriting is the one of UCQs.

We show that the verification problem for complete source-to-ontology rewritings is NP-complete, whereas it is Π_2^p-complete for both sound and perfect source-to-ontology rewritings.

As for the computation problem, we illustrate an algorithm to compute minimally complete source-to-ontology rewritings of given queries over the source schema (thus proving that they always exist), and an algorithm that, given a query over the source schema, it computes a perfect source-to-ontology rewriting if it exists and can be expressed as a UCQ, otherwise it reports that no such UCQ-perfect source-to-ontology rewriting exists. For the case of sound source-to-ontology rewritings, instead, we precisely determine the cases where a maximally sound source-to-ontology rewriting is not guaranteed to exists.

VII. For the case of complete source-to-ontology rewritings, we extend the general scenario by allowing inequalities to occur in the target query language for expressing source-to-ontology rewritings. We first show that UCQ$^{\neq}$s provide better approximated complete source-to-ontology rewritings compared to UCQs, and then we present an algorithm to compute UCQ$^{\neq}$-minimally complete source-to-ontology rewritings of given UCQs over the source schema, thus proving that they always exist. In this extended scenario, we also study both the verification, and the computation problem for complete source-to-ontology rewritings when the UNA is dropped.

VIII. We single out two restricted scenarios that are still meaningful from the point of view of expressive power, and guarantees the existence of maximally sound source-to-ontology rewritings. In both such restrained scenarios, we consider the setting for OBDM specifications obtained from the general one by limiting the ontology language to $\mathit{DL\text{-}Lite}_{\mathsf{RDFS}}$ rather than $\mathit{DL\text{-}Lite}_{\mathcal{R}}$, and limiting the mapping language to GAV assertions rather than GLAV assertions. The difference between the two restricted scenarios is in the query language allowed for expressing data services, where in the first one, called *restricted scenario for CQJFEs*, is the class of *conjunctive queries with join-free existential variables* (CQJFEs), whereas in the second one, called *restricted scenario for UCQJFEs*, is the class of unions of CQJFEs (UCQJFEs). For both the restricted scenarios, we study both the verification, and the computation problem for sound source-to-ontology rewritings.

As for the verification problem, we show that it falls from Π_2^p-complete down to coNP-complete in the restricted scenario for UCQJFEs, and even further down to tractability (i.e., in PTIME) in the restricted scenario for CQJFEs.

As for the computation problem, we first provide an algorithm to compute maximally sound source-to-ontology rewritings of given UCQJFEs over the source schema (thus proving that they are guaranteed to exists in these restricted scenarios), and then we specialise it for the case of CQJFEs.

IX. We provide a detailed relationship between the notions introduced in this thesis and the usual notions of ontology-to-source rewriting and *view-based query rewriting* (i.e., rewriting given queries using view definitions). For this latter long-established notion, we also present new interesting results when dealing with UCQ views, also referred as *disjunctive views*. Specifically, we delineate the precise dividing line between the existence and the non-existence cases of UCQ-maximally sound rewritings of UCQs with respect to disjunctive views, along the dimension of join existential variables occurring in the bodies of the various disjuncts of the given UCQ to be rewritten.

X. Finally, we carry out an investigation of source-to-ontology rewritings expressed in a non-monotonic query language. In this endeavour, the choice of the non-monotonic query language is the class of *EQL-Lite*(UCQ) queries, a particularly well-behaved fragment of *EQL* queries. Such a language incorporates a single modal knowledge operator **K**, which is used to formalise the epistemic state of the OBDM system. As a first contribution, we show how queries of such a non-monotone query language can be rewritten as FOL queries over the source schema to compute certain answers. We then show that *EQL-Lite*(UCQ) queries provides a better means to compute abstractions of data services compared to the language of UCQs. In particular, there are cases where the perfect source-to-ontology rewriting of a query over the source schema is expressible as a *EQL-Lite*(UCQ) query, but not as a UCQ. Also, there are cases where a maximally sound source-to-ontology rewriting exists in the class of *EQL-Lite*(UCQ) queries, but not in the class of UCQs, and cases where a maximally sound (respectively, minimally complete) source-to-ontology rewriting of a query over the source schema is a better approximation than the analogous in the class of UCQs.

   On the other hand, similarly to UCQs, we prove that there are cases where no maximally sound source-to-ontology rewriting exists in the class of *EQL-Lite*(UCQ) queries. Quite surprisingly, we prove that the same holds for minimally complete source-to-ontology rewritings. In order to address the issue of non-expressibility, we explore two special scenarios. In the first one, we limit the query language, and consider a fragment, still non-monotonic, of *EQL-Lite*(UCQ) queries, called *EQL-Lite*$^{-}$(UCQ) queries, where both nested negation and union are not allowed. In the second one, we limit the mapping language, and consider the so-called *One-To-One* mapping, where each mapping assertion links one source relation to one ontology element. For both scenarios, we address the problem of computing minimally complete, and maximally sound source-to-ontology rewritings of source queries, presenting algorithms whenever possible.

1.2 Structure of the Thesis

The thesis is organised in ten chapters, whose content is briefly summarised below:

- Chapter 1 is the current introduction.
- Chapter 2 introduces the relevant theoretical background needed to understand the thesis.
- Chapter 3 illustrates a formal framework for the task of abstraction in OBDM. Here, the various notions of source-to-ontology rewritings are introduced, some associated computational problems are defined, and a detailed comparison with related work is outlined.
- Chapter 4 deals with the problem of answering queries with inequalities over lightweight ontologies. In addition, it presents some interesting implications

that our technical results have on the problem of answering queries with inequalities in the Semantic Web context, and on the containment problem for queries with inequalities in the relational database theory.

- Chapter 5 studies both the verification, and the computation problem for complete source-to-ontology rewritings by providing algorithms and characterising the complexity of both tasks. For this notion, it further studies the computation problem when the target query language allows inequality atoms.
- Chapter 6 studies both the verification, and the computation problem for sound source-to-ontology rewritings. It first analyses the computational complexity of the verification problem, and then precisely determine the cases where a maximally sound source-to-ontology rewriting is not guaranteed to exists.
- Chapter 7 studies both the verification, and the computation problem for perfect source-to-ontology rewritings by providing algorithms and characterising the complexity of both tasks.
- Chapter 8 studies both the verification, and the computation problem for sound source-to-ontology rewritings in two restricted scenarios. For both scenarios, it provides algorithms and complexity results for both the verification, and the computation problem for sound source-to-ontology rewritings. Furthermore, it illustrates new interesting results on view-based query processing, particularly on the problem of rewriting queries using disjunctive views.
- Chapter 9 carries out an investigation of source-to-ontology rewritings when the target query language is non-monotonic. It first argues that non-monotonicity is an important feature when providing abstractions of data services. Then, it determines cases where both maximally sound, and minimally complete source-to-ontology rewritings are not guaranteed to exist in the chosen non-monotonic query language. Lastly, the chapter studies the computation problem for both maximally sound, and minimally complete source-to-ontology rewritings in two special scenarios.
- Finally, Chapter 10 concludes the thesis with a brief discussion and possible directions for future work.

Chapter 2

Theoretical Background

In this chapter, we recall some basic theoretical notions and related results which will be used in following discussions.

2.1 Relational Databases

A *relational database schema* (or simply *schema*) $\mathcal{S}$ is a finite set of predicate symbols, each with a specific arity, and a set of integrity constraints. Given a schema $\mathcal{S}$, an $\mathcal{S}$*-database* D is a finite set of *facts* $s(\vec{c})$ satisfying all integrity constraints in $\mathcal{S}$, 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.

An *incomplete* $\mathcal{S}$*-database* $\mathcal{W}$ is like an $\mathcal{S}$-database, but where also *variables* are allowed as *terms*. Formally, it is a finite set of *atoms* $s(\vec{t})$ over $\mathcal{S}$, where s is an n-ary predicate symbol of $\mathcal{S}$, and $\vec{t} = (t_1, \ldots, t_n)$ is a n-tuple of terms in which t_i is either a constant or a variable, for each $i \in [1, n]$. As usual, each variable is taken from a countable infinite set of symbols denoted by Var, where $\mathsf{Const} \cap \mathsf{Var} = \emptyset$. We will use variables to represent *unknown values* [Imielinski and Lipski Jr., 1984], rather than *non-existent values* [Zaniolo, 1982]. For an incomplete $\mathcal{S}$-database $\mathcal{W}$, we denote by $\mathsf{dom}(\mathcal{W})$ the set of all terms (i.e., constants and variables) occurring in $\mathcal{W}$.

2.2 Query Languages and Homomorphism

In its general form, an $\mathcal{L}_\mathcal{S}$ *query* q over a schema $\mathcal{S}$ is a function in a certain query language $\mathcal{L}_\mathcal{S}$ that can be *evaluated* over an $\mathcal{S}$-database D to return a set of *answers* q^D, each answer being a tuple of constants.

We assume to deal with databases supporting queries in *First-Order Logic (FOL)*. An FOL query q over a schema $\mathcal{S}$ is a query of the form $q = \{\vec{t} \mid \phi(\vec{x})\}$, also denoted $q(\vec{t})$, where $\vec{t}$, called the *target list* of q, is an n-tuple of terms ($ar(q) = n$, where $ar(q)$ denotes the *arity* of q), and $\phi(\vec{x})$, called the *body* of q, is an FOL formula over the predicates of $\mathcal{S}$ in which all the free variables (i.e., the variables in the tuple $\vec{x}$), called the *distinguished variables* of q, occur in $\vec{t}$. As usual, we impose that each variable x occurring in $\vec{t}$ must also appear in some atom of $\phi(\vec{x})$. When $ar(q) = 0$, the query is called *boolean*. Given an FOL query $q = \{(t_1, \ldots, t_n) \mid \phi(\vec{x})\}$ of arity n and an n-tuple of constants $\vec{c} = (c_1, \ldots, c_n)$ such that $c_j = t_j$ for each $j \in [1, n]$ in which t_j is a constant, we denote by $q(\vec{c}) = \{() \mid \phi(\vec{x}/\vec{c})\}$ the boolean FOL query without free variables, in which $\phi(\vec{x}/\vec{c})$ is the FOL sentence obtained from $\phi(\vec{x})$ by replacing all the occurrences of the term t_i with the constant c_i, for each $i \in [1, n]$.

A *conjunctive query (CQ)* q over a schema $\mathcal{S}$ is an FOL query of the form $q = \{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$, where $\vec{y}$ is the tuple of *existential variables* of q, each variable occurring in $\vec{y}$ appears in $\phi(\vec{x}, \vec{y})$, $\vec{x}$ is the tuple of distinguished variables of q, and $\phi(\vec{x}, \vec{y})$ is either $\bot(\vec{x}, \vec{y})$, or a finite conjunction of atoms of the form $s(t'_1, \ldots, t'_n)$, where s is an n-ary predicate of $\mathcal{S}$, and term t'_i is either a constant or a variable occurring in $\vec{x}$ or $\vec{y}$, for each $i = [1, n]$. Given a CQ $q = \{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$, we say that an existential variable y occurring in $\vec{y}$ is a *join existential variable of* q if it occurs more than once in the atoms of $\phi(\vec{x}, \vec{y})$. In what follows, we also consider a subclass of CQs, namely *conjunctive queries with join-free existential variables (CQJFEs)*. A CQ q is also a CQJFE if there is no join existential variable occurring in q.

A *conjunctive query with inequalities (CQ$^{\neq}$)* over a schema $\mathcal{S}$ is an expression of the form $q = \{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$ similar to a CQ, but where $\phi(\vec{x}, \vec{y})$ may additionally contains *inequality atoms*, i.e., atoms of the form $\neq (z_1, z_k)$, also denoted by $z_i \neq z_k$, where both z_i and z_k are either constants or variables in $\vec{x}$ or $\vec{y}$. As usual, we impose safeness [Abiteboul *et al.*, 1995], i.e., each variable z occurring in an inequality atom also occurs in an atom that is not an inequality atom.

When convenient, we treat tuples as sets, in those cases we implicitly refer to the set of all the terms occurring in the tuple. For a CQ$^{\neq}$ $q = \{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$, notice that $\vec{x} \subseteq \vec{t}$ and $\vec{t} \cap \vec{y} = \emptyset$. Furthermore, when convenient, we treat CQ$^{\neq}$s q and their bodies $\phi(\vec{x}, \vec{y})$ as a set of atoms, in those cases we implicitly refer to the set of all the atoms occurring in $\phi(\vec{x}, \vec{y})$ that are not inequality atoms. In particular, for a CQ$^{\neq}$ $q = \{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$ over a schema $\mathcal{S}$, we denote (i) by $\mathcal{W}_q$ the incomplete $\mathcal{S}$-database associated to q, i.e., the set of all atoms over $\mathcal{S}$ occurring in $\phi(\vec{x}, \vec{y})$ that are not inequality atoms, and (ii) by D_q (called the *freezing* of q) the $\mathcal{S}$-database associated to q, i.e., the set of facts over $\mathcal{S}$ obtained from $\mathcal{W}_q$ by replacing each variable v occurring in $\mathcal{W}_q$ with a different fresh constant denoted by c_v.

A class of queries laying between CQs and CQ$^{\neq}$s is the class of *conjunctive queries with bound inequalities (CQ$^{\neq,b}$s)*. A CQ$^{\neq,b}$ $q = \{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$ is a CQ$^{\neq}$ whose inequality atoms involve only constants or distinguished variables, i.e., for each inequality atom $z_i \neq z_k$ appearing in $\phi(\vec{x}, \vec{y})$, we have both $z_i \notin \vec{y}$ and $z_k \notin \vec{y}$.

Another class between CQs and CQ$^{\neq}$s is the class of *conjunctive queries with at most k inequalities (CQ$^{k,\neq}$)*. A CQ$^{k,\neq}$ is a CQ$^{\neq}$ having at most k inequality atoms.

Finally, a *UCQ$^{\neq}$* (respectively, *UCQ$^{\neq,b}$*, *UCQ$^{k,\neq}$*, *UCQ*, *UCQJFE*) is a union of a finite set of CQ$^{\neq}$s (respectively, CQ$^{\neq,b}$s, CQ$^{k,\neq}$s, CQs, CQJFEs) with same arity, called its disjuncts.

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

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

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

Given a $\mathrm{CQ}^{\neq}$ $q = \{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$, the *evaluation* of q over a (possibly infinite) set of atoms $\mathcal{W}$ is the set $q^{\mathcal{W}}$ of tuples of terms $\vec{c}$ such that there exists a homomorphism h from $\mathcal{W}_q$ (i.e., the set of all atoms occurring in $\phi(\vec{x}, \vec{y})$ that are not inequality atoms) to $\mathcal{W}$ for which (*i*) $h(z_i) \neq h(z_k)$ for each inequality atom $z_i \neq z_k$ occurring in $\phi(\vec{x}, \vec{y})$, and (*ii*) $h(\vec{t}) = \vec{c}$, where $h(\vec{t}) = (h(t_1), \ldots, h(t_n))$ for a tuple of terms $\vec{t} = (t_1, \ldots, t_n)$. In what follows, we also say that this is a homomorphism from q to $\mathcal{W}$ (or also a homomorphism from $\phi(\vec{x}, \vec{y})$ to $\mathcal{W}$) with $h(\vec{t}) = \vec{c}$, and write $h(q)$ (or also $h(\phi(\vec{x}, \vec{y}))$), to actually denote $h(\mathcal{W}_q)$. Finally, the evaluation of a $\mathrm{UCQ}^{\neq}$s over a set of atoms $\mathcal{W}$ is simply the union of the evaluation of its disjuncts over $\mathcal{W}$. As an usual convention, for a boolean $\mathrm{CQ}^{\neq}$ q, the evaluation of q over a set of atoms $\mathcal{W}$ amounts to $q^{\mathcal{W}} = \{()\}$ (also denoted by $\mathcal{W} \models q$) if and only if there is a homomorphism from $\mathcal{W}_q$ to $\mathcal{W}$.

Furthermore, given two $\mathrm{CQ}^{\neq}$s $q_1 = \{\vec{t_1} \mid \exists \vec{y_1}.\phi_1(\vec{x_1}, \vec{y_1})\}$ and $q_2 = \{\vec{t_2} \mid \exists \vec{y_2}.\phi_2(\vec{x_2}, \vec{y_2})\}$, we say that h is a homomorphism from q_2 to q_1 if h is a homomorphism from $\mathcal{W}_{q_2}$ to $\mathcal{W}_{q_1}$ for which (*i*) $h(z_i) \neq h(z_k)$ for each inequality atom occurring in $\phi_2(\vec{x}, \vec{y})$, and (*ii*) $h(\vec{t_2}) = \vec{t_1}$.

Given a schema $\mathcal{S}$ and two queries of the same arity q_1 and q_2 over $\mathcal{S}$, we write $q_1 \sqsubseteq_{\mathcal{S}} q_2$ (or simply $q_1 \sqsubseteq q_2$ when $\mathcal{S}$ is clear from the context) if $q_1^D \subseteq q_2^D$ for every $\mathcal{S}$-database D. Furthermore, we write $q_1 \equiv_{\mathcal{S}} q_2$ (or simply $q_1 \equiv q_2$ when $\mathcal{S}$ is clear from the context) if both $q_1 \sqsubseteq q_2$ and $q_2 \sqsubseteq q_1$ hold, that is, if $q_1^D = q_2^D$ for every $\mathcal{S}$-database D. When $\mathcal{S}$ is a database schema without integrity constraints, it is well-known that, if both $q_1 = \{\vec{t_1} \mid \exists \vec{y_1}.\phi_1(\vec{x_1}, \vec{y_1})\}$ and $q_2 = \{\vec{t_2} \mid \exists \vec{y_2}.\phi_2(\vec{x_2}, \vec{y_2})\}$ are CQs over $\mathcal{S}$, then $q_1 \sqsubseteq q_2$ if and only if $\vec{t_1} \in q_2^{\mathcal{W}_{q_1}}$, i.e., if and only if there is a homomorphism h from q_2 to $\mathcal{W}_{q_1}$ with $h(\vec{t_2}) = \vec{t_1}$ [Chandra and Merlin, 1977], and if both q_1 and q_2 are UCQs over $\mathcal{S}$, then $q_1 \sqsubseteq q_2$ if and only if for each disjunct q of q_1 there is a disjunct q' of q_2 such that $q \sqsubseteq q'$ [Sagiv and Yannakakis, 1980].

Given a $\mathrm{CQ}^{\neq}$ $q = \{(t_1, \ldots, t_n) \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$ of arity n and an n-tuple of constants $\vec{c} = (c_1, \ldots, c_n)$, we denote by $q(\vec{c}) = \{() \mid \exists \vec{y}.\phi(\vec{x}/\vec{c}, \vec{y})\}$ the boolean CQ in which the formula $\phi(\vec{x}/\vec{c}, \vec{y})$ corresponds to $\bot(\vec{y})$ in the case that there is some $i \in [1, n]$ for which $t_i \neq c_i$ and t_i is a constant, otherwise $\phi(\vec{x}/\vec{c}, \vec{y})$ is obtained from $\phi(\vec{x}, \vec{y})$ by replacing all the occurrences of the term t_i with the constant c_i, for each $i \in [1, n]$.

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

2.3 Computational Complexity

We assume familiarity with basic notions about computational complexity, as defined in standard textbooks [Garey and Johnson, 1979; Papadimitriou, 1994; Arora and Barak, 2009]. In particular, we consider the following *complexity classes*:

$$\mathrm{AC}^0 \subsetneq \textsc{LogSpace} \subseteq \textsc{NLogSpace} \subseteq \textsc{PTime} \subseteq$$

$$\begin{bmatrix} \mathrm{NP} \\ \mathrm{coNP} \end{bmatrix} \subseteq \mathrm{DP} \subseteq \textsc{PTime}^{\mathrm{NP}} \subseteq \begin{bmatrix} \Sigma_2^p \\ \Pi_2^p \end{bmatrix} \subseteq \textsc{ExpTime} \subseteq \begin{bmatrix} \textsc{NExpTime} \\ \textsc{coNExpTime} \end{bmatrix}$$

It is known that $\mathrm{AC}^0 \subsetneq \textsc{LogSpace}$, for example the undirected graph reachability problem is in LogSpace [Reingold, 2008] but not in AC^0. The strictness of all the other inclusions, as well as whether complexity classes in square brackets coincide, are still open problems. By the time hierarchy theorems [Hartmanis and Stearns, 1965; Cook, 1973], however, it is known that PTime $\subsetneq$ ExpTime and NP $\subsetneq$ NExpTime (respectively, coNP $\subsetneq$ coNExpTime).

Since readers might be less familiar with the complexity class AC^0, we briefly provide a basic intuition about it, and refer to [Vollmer, 1999] for its formal definition which is based on the circuit model. Intuitively, a *decision problem* belongs to AC^0 if it can be decided in constant time using a number of processors that is polynomial in the size of the input. A typical decision problem belonging to AC^0 is the evaluation of FOL queries over relational databases, where only the database is regarded as the input, and the query is assumed to be fixed [Abiteboul *et al.*, 1995].

A decision problem $\mathcal{P}$ is said to be $\mathcal{C}$*-hard* for a complexity class $\mathcal{C}$, if any decision problem $\mathcal{P}' \in \mathcal{C}$ can be reduced to $\mathcal{P}$, and it is said to be $\mathcal{C}$*-complete* if in addition $\mathcal{P} \in \mathcal{C}$. Most of the reductions presented in this thesis are LogSpace reductions.

A LogSpace *reduction* is a reduction computable by a three-tape *Turing machine* [Turing, 1937] that, with an input written on the read-only input tape, writes its output on the write-only output tape using a number of cells of the (initially-blank) read/write work tape that is logarithmic in the size of the input.

2.4 View-based Query Processing

View-based query processing is a general term denoting several tasks related to the presence of views in database systems. In particular, two notions have been subject to extensive investigations in literature, namely *view-based query answering* and *view-based query rewriting*, whose relationship, although widely discussed (see, e.g., [Calvanese *et al.*, 2000; Calvanese *et al.*, 2007c]), seems to be often ignored.

In view-based query answering, a notion originated with [Duschka and Genesereth, 1997], we are given a query, a set of view definitions, and a set of view extensions, and the goal is to compute the so-called *certain answers*, i.e., the set of tuples satisfying the query in all databases consistent with the views.

In view-based query rewriting, a notion originated with [Levy *et al.*, 1995],we are given a query and a set of view definitions, and the goal is to *reformulate* the query into an expression over the alphabet of the view names that satisfies certain conditions. In this thesis, we are mainly interested to view-based query rewriting.

We denote by $\mathcal{V} = \{V_1, \ldots, V_n\}$ and $\mathcal{E} = \{E_1, \ldots, E_n\}$ a set of *view definitions* and *view extensions* over a schema $\mathcal{S}$, respectively, where, for each $i \in [1, n]$, to the symbol V_i it is implicitly associated a query over a schema $\mathcal{S}$, and E_i is the view extension of V_i (i.e., a set of facts having V_i as a predicate with arity the one of the corresponding query). When dealing with (U)CQ view definitions $\mathcal{V} = \{V_1, \ldots, V_n\}$, we implicitly assume that the target list (of each disjunct) of the (U)CQ associated to V_i does not have repeated variables or constants, for each $i \in [1, n]$.

2.4.1 Exact View Assumption

Given a schema $\mathcal{S}$, a set of view definitions $\mathcal{V} = \{V_1, \ldots, V_n\}$ over $\mathcal{S}$, and a query $q_{\mathcal{S}}$ over $\mathcal{S}$, the goal is to reformulate $q_{\mathcal{S}}$ into an expression over the view alphabet (called *rewriting*) that, when evaluated over the set of view extensions $\mathcal{V}(D) = \{V_1^D, \ldots, V_n^D\}$

(for a view symbol V_i, V_i^D denotes the evaluation of the query associated to V_i over D), coincide with the evaluation of $q_{\mathcal{S}}$ over D (i.e., $q_{\mathcal{S}}^D$), for each $\mathcal{S}$-database D.

Formally, for a set of view definitions $\mathcal{V}$ and a query $q_{\mathcal{S}}$ over a schema $\mathcal{S}$, we say that a query $q_{\mathcal{V}}$ is an *exact rewriting* of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$, if $q_{\mathcal{S}}^D = q_{\mathcal{V}}^{\mathcal{V}(D)}$ for each $\mathcal{S}$-database D. The underlying decision problem associated to exact rewritings is the *expressibility problem*: given a set of view definitions $\mathcal{V}$ and a query $q_{\mathcal{S}}$ over a schema $\mathcal{S}$, does there exist an exact rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$?[1] This happens to be a difficult problem, it is indeed in general undecidable already when both view definitions in $\mathcal{V}$ and queries $q_{\mathcal{S}}$ are CQs [Gogacz and Marcinkowski, 2016].

Obviously, one might be interested not in arbitrary exact rewritings, but in those belonging to a certain query language $\mathcal{L}$. Formally, we say that $q_{\mathcal{V}}$ is an $\mathcal{L}$-exact rewriting of a query $q_{\mathcal{S}}$ with respect to a set of view definitions $\mathcal{V}$, if $q_{\mathcal{V}}$ is an exact rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$ and $q_{\mathcal{V}} \in \mathcal{L}$. When view definitions in $\mathcal{V}$ are CQs and queries $q_{\mathcal{S}}$ are (U)CQs, the problem of deciding whether there exists a (U)CQ-exact rewriting is NP-complete [Levy *et al.*, 1995].

2.4.2 Sound View Assumption

The setup we have considered so far is that of *exact views*, in the sense that the extension of each view is precisely the result of evaluating the corresponding view expression over the database. There is, however, at least one other notion of a database being coherent with the view extensions,[2] namely that of *sound views*. Specifically, an $\mathcal{S}$-database D is consistent with the set of view extensions $\mathcal{E} = \{E_1, \dots, E_n\}$ with respect to the set of view definitions $\mathcal{V} = \{V_1, \dots, V_n\}$ under the sound view assumption, if $\mathcal{E} \subseteq \mathcal{V}(D)$, i.e., if $E_i \subseteq V_i^D$ for each $i \in [1, n]$.

Given a set of view definitions $\mathcal{V}$ over a schema $\mathcal{S}$, a set of view extensions $\mathcal{E}$, and a query $q_{\mathcal{S}}$ over $\mathcal{S}$, we denote by $cert_{q_{\mathcal{S}},\mathcal{V}}^{\mathcal{E}}$ the *certain answers* of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$ and $\mathcal{E}$, i.e., the set of tuples of constants that are in all the evaluations of $q_{\mathcal{S}}$ over $\mathcal{S}$-databases D consistent with $\mathcal{E}$ with respect to $\mathcal{V}$. Formally:

$$cert_{q_{\mathcal{S}},\mathcal{V}}^{\mathcal{E}} := \bigcap_{D\colon \mathcal{E} \subseteq \mathcal{V}(D)} q_{\mathcal{S}}^D$$

Given a set of view definitions $\mathcal{V}$ over a schema $\mathcal{S}$ and a query $q_{\mathcal{S}}$ over $\mathcal{S}$, following the literature terminology, we say that a query $q_{\mathcal{V}}$ over the view alphabet $\mathcal{V}$ is a *perfect rewriting* of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$, if $q_{\mathcal{V}}^{\mathcal{E}} = cert_{q_{\mathcal{S}},\mathcal{V}}^{\mathcal{E}}$ for each view extension $\mathcal{E}$.

Furthermore, we say that a query $q_{\mathcal{V}}$ over the view alphabet $\mathcal{V}$ is a *sound rewriting* of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$, if $q_{\mathcal{V}}^{\mathcal{E}} \subseteq q_{\mathcal{S}}^D$ for each view extension $\mathcal{E}$ and $\mathcal{S}$-database D such that $\mathcal{E} \subseteq \mathcal{V}(D)$. Obviously, the interest is in computing sound rewritings that capture the original query at best. This is formalized by the following notion, where $\mathcal{L}$ is a query language. A query $q_{\mathcal{V}} \in \mathcal{L}$ over the view alphabet $\mathcal{V}$ is an *$\mathcal{L}$-maximally sound rewriting* of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$, if $q_{\mathcal{V}}$ is a sound rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$ and there exists no query $q'_{\mathcal{V}} \in \mathcal{L}$ such that (i) $q'_{\mathcal{V}}$ is a sound rewriting of $q_{\mathcal{S}}$ with

[1] In literature, this problem is also referred as *determinacy* [Nash *et al.*, 2010] and *losslessness under the exact view assumption* [Calvanese *et al.*, 2007c].

[2] In fact, a third notion considered in literature is that of *complete views*, which, however, we do not consider it here.

respect to $\mathcal{V}$, (*ii*) $q_{\mathcal{V}}^{\mathcal{E}} \subseteq {q'_{\mathcal{V}}}^{\mathcal{E}}$ for each set of view extensions $\mathcal{E}$, and (*iii*) $q_{\mathcal{V}}^{\mathcal{E}'} \subsetneq {q'_{\mathcal{V}}}^{\mathcal{E}'}$ for a set of view extensions $\mathcal{E}'$.

We conclude this section with the following renowned positive result for CQ view definitions and UCQ queries.

Theorem 2.1. [Levy *et al.*, 1995] *Let $\mathcal{V}$ be a set of CQ view definitions over a schema $\mathcal{S}$. For a UCQ $q_{\mathcal{S}}$ over $\mathcal{S}$, it is always possible to compute the UCQ $q_{\mathcal{V}}$ which is the union of all CQ-maximally sound rewritings of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$. Moreover:*

- *$q_{\mathcal{V}}$ is a UCQ-maximally sound rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$;*
- *$q_{\mathcal{V}}$ is a perfect rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$;*
- *$q_{\mathcal{V}}$ is a UCQ-exact rewriting of $q_{\mathcal{S}}$ wih respect to $\mathcal{V}$, if this latter exists.*

Notice, however, that the above theorem is no longer true when view definitions $\mathcal{V}$ are expressed as UCQs rather than CQs [Duschka and Genesereth, 1998; Afrati and Chirkova, 2019]. Specifically, there are cases where a UCQ-maximally sound rewriting of a CQ $q_{\mathcal{S}}$ with respect to a set of UCQ view definitions $\mathcal{V}$ does not exist.

2.5 Description Logic Ontologies and Knowledge Bases

Description Logics (DLs) are fragments of FOL languages using only unary and binary predicates, called *atomic concepts* and *atomic roles*, respectively [Baader *et al.*, 2003; Baader *et al.*, 2017]. According to [Gruber, 1993; Gruber, 2018], an *ontology* is a formal explicit specification of a shared conceptualisation of a domain of interest. Since DLs are logics specifically designed to represent structure knowledge and to reason about it, they are arguably well-suited to represent ontologies.

In this thesis, 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 an ontology $\mathcal{O}$ as a schema. In such cases, we implicitly refer to the finite set of unary and binary predicates corresponding to atomic concepts and atomic roles, respectively, which constitute the *alphabet* of $\mathcal{O}$.

2.5.1 *DL-Lite*$_{\mathcal{R}}$ Ontologies: Syntax

We are interested in DL ontologies expressed in *DL-Lite*$_{\mathcal{R}}$, a member of the *DL-Lite* family[3] [Calvanese *et al.*, 2004a; Calvanese *et al.*, 2005; Calvanese *et al.*, 2007b] of DLs. Notably, *DL-Lite*$_{\mathcal{R}}$ is the logic underpinning `OWL 2 QL`[4], i.e., one of the three `OWL 2`[5] profiles [Cuenca Grau *et al.*, 2008; Motik *et al.*, 2012], specifically the one especially designed for efficient query answering. Notice that, in the Semantic Web [Berners-Lee *et al.*, 2001] context, `OWL 2` is the current W3C recommended standard ontology language.

[3]Not to be confused with the *DL-Lite*$_{\mathsf{bool}}$ family studied in [Artale *et al.*, 2009], a supremum of the *DL-Lite* family in the lattice of DLs.

[4]`OWL 2 Query Language`: https://www.w3.org/TR/owl2-profiles/#OWL_2_QL

[5]`OWL 2 Web Ontology Language`: https://www.w3.org/TR/owl2-profiles/

In *DL-Lite*$_\mathcal{R}$, concepts and roles obey to the following syntax:

$$B \longrightarrow A \mid \exists R \qquad\qquad R \longrightarrow P \mid P^-$$

where A, P, and P^- denote an atomic concept (atomic concepts include the *universal concept* $\top$ and the *bottom concept* $\bot$), an atomic role, and the *inverse* of an atomic role, respectively. R denotes a *basic role*, i.e., a role that is either an atomic role or its inverse. B denotes a *basic concept*, i.e., a concept that is either an atomic concept or $\exists R$, where this latter is the standard DL construct of *unqualified existential quantification* on basic roles.

A *DL-Lite*$_\mathcal{R}$ ontology $\mathcal{O}$ is a finite set of *assertions* of the form:

$$\begin{array}{lll} B_1 \sqsubseteq B_2 & R_1 \sqsubseteq R_2 & \text{(concept/role inclusion assertion)} \\ B_1 \sqsubseteq \neg B_2 & R_1 \sqsubseteq \neg R_2 & \text{(concept/role disjointness assertion)} \end{array}$$

Without loss of generality, we assume that each *DL-Lite*$_\mathcal{R}$ ontology $\mathcal{O}$ contains an inclusion assertion of the form $A \sqsubseteq \top$ (respectively, $\exists P \sqsubseteq \top$ and $\exists P^- \sqsubseteq \top$), for each possible atomic concept A (respectively, atomic role P) in the alphabet of $\mathcal{O}$.

Let C (respectively, E) denote a *general concept* (respectively, *general role*), i.e., a concept (respectively, role) that is either a basic concept (respectively, basic role) or its negation. In principle, one might also include $B_1 \sqcup B_2$ and $R_1 \sqcup R_2$ (respectively, $C_1 \sqcap C_2$ and $E_1 \sqcap E_2$) in the constructs for the left-hand side (respectively, right-hand side) of assertions, where $\sqcup$ (respectively, $\sqcap$) denotes union (respectively, intersection). Notice, however, that the expressive capabilities of the language would remain the same, since in fact assertions of the form $B_1 \sqcup B_2 \sqsubseteq C$ and $R_1 \sqcup R_2 \sqsubseteq E$ (respectively, $B \sqsubseteq C_1 \sqcap C_2$ and $R \sqsubseteq E_1 \sqcap E_2$) are equivalent to the pair of assertions $B_1 \sqsubseteq C, B_2 \sqsubseteq C$ and $R_1 \sqsubseteq E, R_2 \sqsubseteq E$ (respectively, $B \sqsubseteq C_1, B \sqsubseteq C_2$ and $R \sqsubseteq E_1, R \sqsubseteq E_2$).

Similar arguments hold for the additional assertions included in `OWL 2 QL`, namely *symmetry*, *asymmetry*, *qualified existential quantification*, *reflexivity*, and *irreflexivity*. It is immediate to see that symmetry (respectively, asymmetry) of an atomic role P can be expressed by means of the inclusion assertion $P \sqsubseteq P^-$ (respectively, $P \sqsubseteq \neg P^-$). Likewise, a qualified existential quantification of the form $B_1 \sqsubseteq \exists R.B_2$ is equivalent to the assertions $B_1 \sqsubseteq \exists P_{\text{new}}$, $P_{\text{new}} \sqsubseteq R$, and $\exists P^-_{\text{new}} \sqsubseteq B_2$, where P_{new} is a fresh atomic role. Furthermore, the addition of reflexivity and irreflexivity assertions (which include a form of *second-order constructs*) does not affect the computational complexity of the basic inference problems (including query answering) [Corona *et al.*, 2009; Artale *et al.*, 2009], and such constructs could be easily included with only minor changes in reasoning algorithms.

Observe that *DL-Lite*$_\mathcal{R}$ is an extension of (the DL-like part of) the ontology language `RDFS`[6] [Cuenca Grau, 2004], the schema language for `RDF`[7]. We will refer to this latter DL ontology language as *DL-Lite*$_{\mathsf{RDFS}}$. Specifically, a *DL-Lite*$_{\mathsf{RDFS}}$ is a finite set of assertions of the form:

$$B \sqsubseteq A \qquad\qquad R_1 \sqsubseteq R_2$$

[6] `Resource Description Framework Schema` [Brickley and Guha, 2014].

[7] `Resource Description Framework`: https://www.w3.org/RDF/

We will also consider a slight extension of the DL ontology language $DL\text{-}Lite_{\mathsf{RDFS}}$, namely $DL\text{-}Lite^{\neg}_{\mathsf{RDFS}}$, which also allows for the concept/role disjointness assertions expressible in $DL\text{-}Lite_{\mathcal{R}}$.

We further note that $DL\text{-}Lite_{\mathcal{R}}$ is an extension of $DL\text{-}Lite_{core}$ i.e., the basic logic of the *DL-Lite* family. A $DL\text{-}Lite_{core}$ ontology is a finite set of assertions of the form:

$$B_1 \sqsubseteq B_2 \qquad B_1 \sqsubseteq \neg B_2$$

Observe that $DL\text{-}Lite_{core}$ and $DL\text{-}Lite_{\mathsf{RDFS}}$ (respectively, $DL\text{-}Lite^{\neg}_{\mathsf{RDFS}}$) are incomparable fragments of $DL\text{-}Lite_{\mathcal{R}}$.

2.5.2 *DL-Lite*$_{\mathcal{R}}$ Ontologies: Semantics

The semantics of DL ontologies is specified through the notion of interpretation: an FOL *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* $\Delta^{\mathcal{I}} \subseteq \mathsf{Const}$ is a non-empty, possibly infinite set of objects, and the *interpretation function* $\cdot^{\mathcal{I}}$ assigns to each atomic concept A a set of domain objects $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, 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 $DL\text{-}Lite_{\mathcal{R}}$, the interpretation function extends to other basic concepts, basic roles, and the additional binary predicate $\neq$ as follows:

- $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$
- $\bot^{\mathcal{I}} = \emptyset$
- $\neq^{\mathcal{I}} = \{(o, o') \mid o, o' \in \Delta^{\mathcal{I}} \wedge o \neq o'\}$. We often write $(o, o') \in \neq^{\mathcal{I}}$ as $o \neq^{\mathcal{I}} o'$.
- $(\exists P)^{\mathcal{I}} = \{o \mid \exists o'. \ (o, o') \in P^{\mathcal{I}}\}$
- $(P^{-})^{\mathcal{I}} = \{(o, o') \mid (o', o) \in P^{\mathcal{I}}\}$

When convenient, we treat interpretations $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ for $\mathcal{O}$ as a (possibly infinite) set of facts over $\mathcal{O}$, in those cases we implicitly refer to the set of facts including: for each unary predicate U and for each $e \in \Delta^{\mathcal{I}}$ (respectively, for each binary predicate B and for each pair $e_1, e_2 \in \Delta^{\mathcal{I}}$), the fact $U(e)$ (respectively, $B(e_1, e_2)$) if and only if $e \in U^{\mathcal{I}}$ (respectively, $(e_1, e_2) \in B^{\mathcal{I}}$).

We say that an interpretation $\mathcal{I}$ for an ontology $\mathcal{O}$ *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, $R_1^{\mathcal{I}} \cap R_2^{\mathcal{I}} = \emptyset$). Finally, 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 $\mathcal{O}$.

2.5.3 *DL-Lite*$_{\mathcal{R}}$ Knowledge Bases

An $\mathcal{L}$ *knowledge base* $\mathcal{K}$ is a pair $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$, where $\mathcal{O}$ is a DL ontology expressed in $\mathcal{L}$, and $\mathcal{A}$ is an ABox ("Assertional Box") for $\mathcal{O}$, i.e., a finite set of *membership assertions* (or equivalently, a finite set of facts) of the form:

$$A(a) \qquad P(a, b)$$

where a, b are constants (also known as *individuals*) in Const, and A and P is an atomic concept and an atomic role, respectively, in the alphabet of $\mathcal{O}$.

The semantics of an $\mathcal{L}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ is given in terms of *interpretations for* $\mathcal{K}$, i.e., FOL interpretations $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ for $\mathcal{O}$ such that the interpretation function $\cdot^{\mathcal{I}}$ further assigns to each individual a occurring in $\mathcal{A}$ a domain object $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. Unless otherwise stated, we adopt the so-called *unique name assumption (UNA)*, i.e., we consider only those interpretations $\mathcal{I}$ for which $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ for each pair of individuals a, b occurring in $\mathcal{A}$ with $a \neq b$

An interpretation $\mathcal{I}$ for an $\mathcal{L}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ is a model of $\mathcal{K}$, denoted by $\mathcal{I} \models \mathcal{K}$, if $\mathcal{I} \models \mathcal{O}$ and $\mathcal{I} \models \mathcal{A}$, where $\mathcal{I} \models \mathcal{A}$ if, for each membership assertion $A(a)$ (respectively, $P(a, b)$) in $\mathcal{A}$, we have that $a^{\mathcal{I}} \in A^{\mathcal{I}}$ (respectively, $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in P^{\mathcal{I}}$).

The set of *models* of an $\mathcal{L}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$, denoted by $Mod_{\mathcal{A}}(\mathcal{O})$, is the set of interpretations $\mathcal{I}$ for $\mathcal{K}$ such that $\mathcal{I} \models \mathcal{K}$. An $\mathcal{L}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ is said to be *satisfiable* if $Mod_{\mathcal{A}}(\mathcal{O}) \neq \emptyset$, *unsatisfiable* otherwise.

For an $\mathcal{L}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$, an interpretation $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ for $\mathcal{K}$, and a UCQ$^{\neq}$ query $q_{\mathcal{O}}$ over $\mathcal{O}$, with a slight abuse of notation, we denote by $q_{\mathcal{O}}^{\mathcal{I}}$ the evaluation of the query $q'_{\mathcal{O}}$ over $\mathcal{I}$, where $q'_{\mathcal{O}}$ is obtained from $q_{\mathcal{O}}$ by replacing each individual c occurring in $q_{\mathcal{O}}$ with the domain object $c^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ (if defined, i.e., if c occurs in $\mathcal{A}$). Furthermore, for a CQ$^{\neq}$ $q_{\mathcal{O}}$ and an interpretation $\mathcal{I}$, we say that h is a homomorphism from $q_{\mathcal{O}}$ to $\mathcal{I}$, if h is a homomorphism from $\mathcal{W}_{q'_{\mathcal{O}}}$ (i.e., the set of all atoms occurring in the body of $q'_{\mathcal{O}}$ that are not inequality atoms) to $\mathcal{I}$.

Finally, given an $\mathcal{L}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ and a UCQ$^{\neq}$ $q_{\mathcal{O}}$ over $\mathcal{O}$, we denote by $cert^{\mathcal{A}}_{q_{\mathcal{O}},\mathcal{O}}$ the set of *certain answers* of $q_{\mathcal{O}}$ with respect to $\mathcal{O}$ and $\mathcal{A}$, i.e., the set of tuples of constants $(c_1, \ldots, c_n)$ such that $(c_1^{\mathcal{I}}, \ldots, c_n^{\mathcal{I}}) \in q_{\mathcal{O}}^{\mathcal{I}}$ for each $\mathcal{I} \in Mod_{\mathcal{A}}(\mathcal{O})$. For a boolean FOL query $q_{\mathcal{O}}$ over $\mathcal{O}$, we denote by $\mathcal{I} \models q_{\mathcal{O}}$ (also by $q_{\mathcal{O}}^{\mathcal{I}} = \{()\}$) the fact that the body of $q_{\mathcal{O}}$, which is a FOL sentence, is true in $\mathcal{I}$, and by $\mathcal{K} \models q_{\mathcal{O}}$ (also by $cert^{\mathcal{A}}_{q_{\mathcal{O}},\mathcal{O}} = \{()\}$) the fact that $\mathcal{I} \models q_{\mathcal{O}}$ for each model $\mathcal{I}$ of $\mathcal{K}$.

Observe that, if $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ is unsatisfiable, then the set of certain answers of any query $q_{\mathcal{O}}$ (over $\mathcal{O}$) with respect to $\mathcal{O}$ and $\mathcal{A}$ is trivially the set of all possible tuples of constants occurring in $\mathcal{A}$ whose arity is the one of the query (*ex falso sequitur quodlibet*), i.e., $cert^{\mathcal{A}}_{q_{\mathcal{O}},\mathcal{O}} = \{(x_1, \ldots, x_{ar(q_{\mathcal{O}})}) \mid \top(x_1, \ldots, x_{ar(q_{\mathcal{O}})})\}^{\mathcal{I}_{\mathcal{A}}}$, where $\top(x_1, \ldots, x_{ar(q_{\mathcal{O}})})$ (similarly, $\bot(x_1, \ldots, x_{ar(q_{\mathcal{O}})})$) is a shortcut denoting the conjunction of atoms $\top(x_1) \wedge \ldots \wedge \top(x_{ar(q_{\mathcal{O}})})$ (similarly, $\bot(x_1) \wedge \ldots \wedge \bot(x_{ar(q_{\mathcal{O}})})$), and $\mathcal{I}_{\mathcal{A}}$ is the interpretation for $\mathcal{K}$ obtained as follows:

- $\Delta^{\mathcal{I}_{\mathcal{A}}}$ is composed of all individuals occurring in $\mathcal{A}$;
- $a^{\mathcal{I}_{\mathcal{A}}} = a$, for each individual a occurring in $\mathcal{A}$;
- $A^{\mathcal{I}_{\mathcal{A}}} = \{a \mid A(a) \in \mathcal{A}\}$ for each atomic concept $\mathcal{A}$;
- $P^{\mathcal{I}_{\mathcal{A}}} = \{(a, b) \mid P(a, b) \in \mathcal{A}\}$ for each atomic role P.

When we talk about the problem of answering queries belonging to a query language $\mathcal{Q}$ over $\mathcal{L}$ knowledge bases, we implicitly refer to the following decision problem: Given a query $q \in \mathcal{Q}$, an $\mathcal{L}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$, and a tuple of constants $\vec{c}$, check whether $\vec{c} \in cert^{\mathcal{A}}_{q_{\mathcal{O}},\mathcal{O}}$.

From [Calvanese *et al.*, 2007b], it is well-known that answering UCQs over *DL-Lite*$_{\mathcal{R}}$ knowledge bases is *FOL-rewritable*, i.e., for every UCQ q and every

DL-Lite$_{\mathcal{R}}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$, it is possible to compute the set of certain answers $cert^{\mathcal{A}}_{q_{\mathcal{O}},\mathcal{O}}$ by first reformulating $q_{\mathcal{O}}$ with respect to $\mathcal{O}$ (obtaining a FOL query over $\mathcal{O}$), and then by evaluating the reformulated query over $\mathcal{I}_{\mathcal{A}}$.

Algorithm 2.1 PerfectRef

Input:
DL-Lite$_{\mathcal{R}}$ ontology $\mathcal{O}$;
UCQ $q_{\mathcal{O}} = q^1_{\mathcal{O}} \cup \ldots \cup q^n_{\mathcal{O}}$ over $\mathcal{O}$
Output:
UCQ q_r over $\mathcal{O}$

$q_r := \emptyset$
for $i \leftarrow 1$ to n **do**
  PR $:= \{\tau(q^i_{\mathcal{O}})\}$
  repeat
    PR$'$ $:=$ PR
    for each $q \in$ PR$'$ **do**
      for each atom g in q **do**
        for each inclusion assertion I in $\mathcal{O}$ that is applicable to g **do**
          PR $:=$ PR $\cup \{q[g/gr(g, I)]\}$
        end for
        for each pair of atoms g_1, g_2 in q **do**
          if g_1 and g_2 may unify **then**
            PR $:=$ PR $\cup \{\tau(reduce(q, g_1, g_2))\}$
          end if
        end for
      end for
    end for
  until PR$'$ $=$ PR
  $q_r := q_r \cup \{q \mid q \in \text{PR}\}$
end for
return q_r

More specifically, for satisfiable *DL-Lite*$_{\mathcal{R}}$ knowledge bases $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ we have that, if $q_{\mathcal{O}}$ is a UCQ over $\mathcal{O}$, then PerfectRef$(\mathcal{O}, q_{\mathcal{O}})^{\mathcal{I}_{\mathcal{A}}} = cert^{\mathcal{A}}_{q_{\mathcal{O}},\mathcal{O}}$, where PerfectRef$(\mathcal{O}, q_{\mathcal{O}})$ is the UCQ obtained by executing the algorithm PerfectRef described in [Calvanese *et al.*, 2007b] and reported above on $\mathcal{O}$ and $q_{\mathcal{O}}$. Note that PerfectRef ignores the disjointness assertions of the input *DL-Lite*$_{\mathcal{R}}$ ontology. Let "_" represents an existential variable that is not a join existential variable. In the algorithm, τ is a function that, given a CQ q, it returns a CQ obtained by replacing each existential variable that is not a join existential variable with the symbol _. The algorithm uses the notion of applicability of an inclusion assertion to an atom. Specifically, an inclusion assertion I is *applicable to an atom* $A(x)$ (respectively, $P(x_1, x_2)$), if the right-hand side of I is A (respectively, if (i) $x_2 = \_$ and the right-hand side of I is $\exists P$; or (ii) $x_1 = \_$ and the right-hand side of I is $\exists P^-$; or (iii) the right-hand side of I is either P or P^-). Furthermore, $q[g/gr(g, I)]$ denotes the CQ obtained from q by replacing the atom g with a new atom $gr(g, I)$, where, for an inclusion assertion I applicable to g, $gr(g, I)$ is the atom defined as follows:

- if $g = A(x)$ and $I = A_1 \sqsubseteq A$, then $gr(g, I) = A_1(x)$;
- if $g = A(x)$ and $I = \exists P \sqsubseteq A$, then $gr(g, I) = P(x, \_)$;
- if $g = A(x)$ and $I = \exists P^- \sqsubseteq A$, then $gr(g, I) = P(\_, x)$;
- if $g = P(x, \_)$ and $I = A \sqsubseteq \exists P$, then $gr(g, I) = A(x)$;
- if $g = P(x, \_)$ and $I = \exists P_1 \sqsubseteq \exists P$, then $gr(g, I) = P_1(x, \_)$;
- if $g = P(x, \_)$ and $I = \exists P_1^- \sqsubseteq \exists P$, then $gr(g, I) = P_1(\_, x)$;
- if $g = P(\_, x)$ and $I = A \sqsubseteq \exists P^-$, then $gr(g, I) = A(x)$;
- if $g = P(\_, x)$ and $I = \exists P_1 \sqsubseteq \exists P^-$, then $gr(g, I) = P_1(x, \_)$;
- if $g = P(\_, x)$ and $I = \exists P_1^- \sqsubseteq \exists P^-$, then $gr(g, I) = P_1(\_, x)$;
- if $g = P(x_1, x_2)$ and either $I = P_1 \sqsubseteq P$ or $I = P_1^- \sqsubseteq P^-$, then $gr(g, I) = P_1(x_1, x_2)$;
- if $g = P(x_1, x_2)$ and either $I = P_1 \sqsubseteq P^-$ or $P_1^- \sqsubseteq P$, then $gr(g, I) = P_1(x_2, x_1)$.

Finally, *reduce* is a function that, given a CQ q and two atoms g_1 and g_2, it returns a new CQ obtained by applying the *most general unifier* between g_1 and g_2. In unifying g_1 and g_2, each occurrence of the symbol $\_$ is considered a different existential variable. The most general unifier substitutes each symbol in g_1 with the corresponding argument in g_2, and vice versa (obviously, if both arguments are $\_$, the resulting argument is still $\_$).

Furthermore, a *DL-Lite*$_\mathcal{R}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ is satisfiable if and only if $cert^{\mathcal{A}}_{\mathcal{V}_\mathcal{O}, \mathcal{O}^p} = \emptyset$, where again $cert^{\mathcal{A}}_{\mathcal{V}_\mathcal{O}, \mathcal{O}^p} = \mathsf{PerfectRef}(\mathcal{O}, \mathcal{V}_\mathcal{O})^{\mathcal{I}_\mathcal{A}}$. Here, $\mathcal{O}^p$ is the *DL-Lite*$_\mathcal{R}$ ontology obtained from $\mathcal{O}$ by removing the disjointness assertions in $\mathcal{O}$, and $\mathcal{V}_\mathcal{O}$ is the *violation query for* $\mathcal{O}$, i.e., the boolean UCQ obtained by including 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 $R_1 \sqsubseteq \neg R_2$), 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., $R = P^-$. Furthermore, we denote by $\mathcal{V}^n_\mathcal{O}$ the UCQ over $\mathcal{O}$ of arity n obtained by adding the target list $(x_1, \ldots, x_n)$ and $\top(x_1, \ldots, x_n)$ in the body to each disjunct of $\mathcal{V}_\mathcal{O}$, where x_i is a fresh distinguished variable for each $i \in [1, n]$.

Observe that *DL-Lite*$_\mathcal{R}$ is insensitive to the adoption of the UNA for UCQ answering [Artale *et al.*, 2009]. Specifically, a *DL-Lite*$_\mathcal{R}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ is satisfiable if and only if it is so without the UNA, and the set of certain answers of an UCQ q with respect to $\mathcal{O}$ and $\mathcal{A}$ does not depend on the adoption of the UNA.

We conclude this subsection with a consideration on the computational complexity of the query answering problem. Both with and without the UNA, the problem of answering UCQs over *DL-Lite*$_\mathcal{R}$ knowledge bases is FOL-rewritable (and therefore in AC^0 in *data complexity*, i.e., the complexity where only the ABox is considered to be the input [Vardi, 1982]), and NP-complete in combined complexity, i.e., the complexity with respect to all inputs of the problem [Calvanese *et al.*, 2007b].

2.6 Ontology-based Data Management

According to [Lenzerini, 2018], *Ontology-based Data Management (OBDM)* can be seen as a sophisticated form of *Information Integration* [Lenzerini, 2002; Calvanese and De Giacomo, 2005; Doan *et al.*, 2012], where the usual global schema is replaced by the conceptual model of an application domain, formulated as an ontology.

The OBDM paradigm resorts to a three-level architecture, consisting of the ontology, some existing data sources relevant for an organization, and the mapping between the two. In all this thesis, we assume that data sources are expressed as a unique relational database schema. Note that this is a realistic assumption, since many, nowadays available, off-the-shelf *Data Federation/Virtualisation* tools can be used to wrap multiple, possibly non-relational, sources, and present them as they were structured according to a single schema.

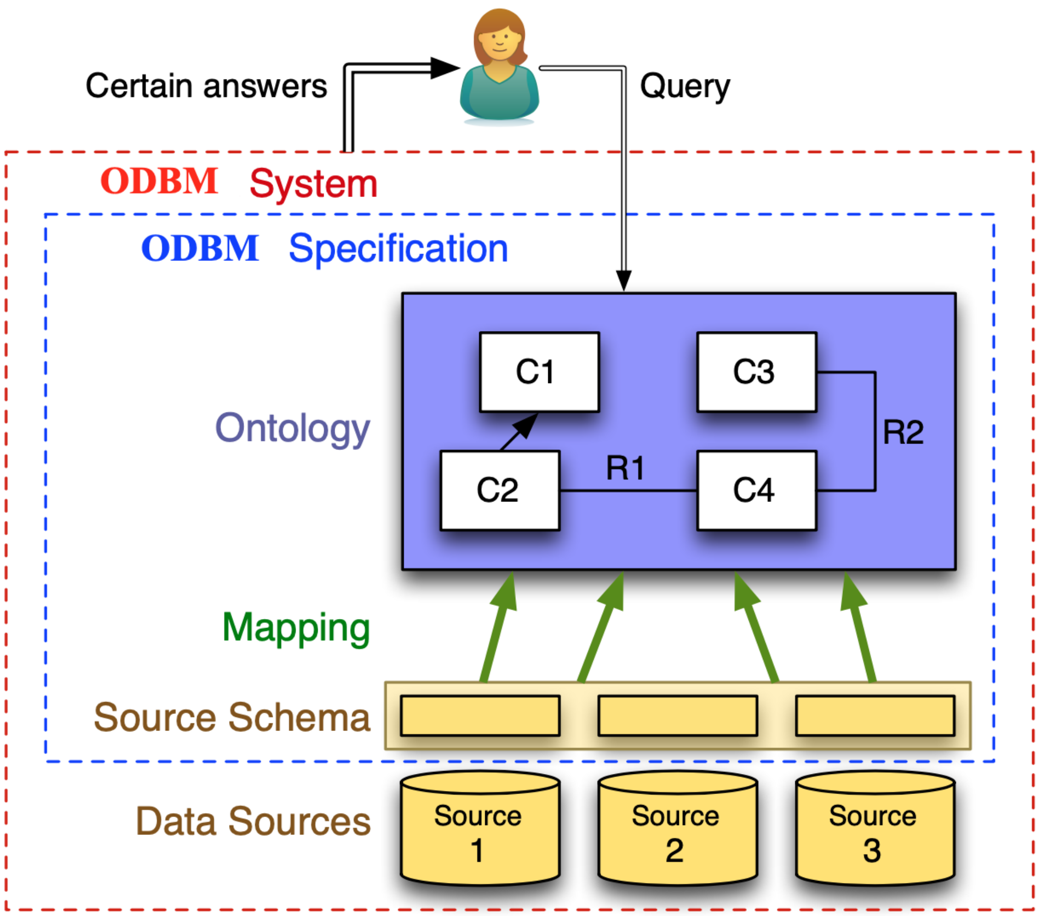

Figure 2.1. OBDM Specification and System

We distinguish between the specification of an OBDM system, and the OBDM system itself (cf. Figure 2.1). From a more formal perspective, an OBDM specification Σ determines the intensional level of the system, and it is expressed as a triple $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, where:

- $\mathcal{O}$ is a DL ontology;

- $\mathcal{S}$ is a relational database schema, also called *source schema*;

- $\mathcal{M}$ is a *mapping*, i.e., a finite set of mapping assertions relating $\mathcal{S}$ to $\mathcal{O}$.

An OBDM system is a pair $\langle \Sigma, D\rangle$, where $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M}\rangle$ is an OBDM specification, and D is an $\mathcal{S}$-database, also called *source database* for Σ.

2.6.1 Mapping Specifications

When mapping data sources to ontologies, one should take into account that sources store values (data), whereas instances of atomic concepts (respectively, atomic roles) are objects (respectively, pair of objects), where each object should be denoted by an ad-hoc identifier (a semantical object), not to be confused with any data item. Dealing with this problem, however, would uselessly complicate our technical treatment. Therefore, for ease of exposition, we ignore this problem (known as the *impedance mismatch problem* [Meseguer and Qian, 1993]), and refer to [Poggi *et al.*, 2008] for technical details on how to solve it. We notice, however, that all our results can easily carry over in a setting that take into account also such issue.

Commonly, mapping assertions are assumed to be *sound*, which means that certain patterns over the source schema implies certain patterns at the ontology level. Typically, from a logical point of view, sound mapping assertions constituting a mapping $\mathcal{M}$ relating a schema $\mathcal{S}$ to an ontology $\mathcal{O}$ are of the form $\forall \vec{x}.(\exists \vec{y}.\phi_{\mathcal{S}}(\vec{x},\vec{y}) \rightarrow \exists \vec{z}.\varphi_{\mathcal{O}}(\vec{x},\vec{z}))$, where $\phi_{\mathcal{S}}(\vec{x},\vec{y})$ and $\varphi_{\mathcal{O}}(\vec{x},\vec{z})$ are body of CQs, i.e., finite conjunction of atoms over $\mathcal{S}$ and $\mathcal{O}$, respectively [Lenzerini, 2002; Doan *et al.*, 2012]. Mapping assertions of the above form are also called *GLAV (Global-and-Local-as-View)* mapping assertions. Special cases of GLAV mapping assertions are *GAV (Global-as-View)* and *LAV (Local-as-View)* mapping assertions.

A GAV mapping assertion is a GLAV mapping assertion in which the right-hand side of the implication does not make use of existential variables, i.e., it is an assertion of the form $\forall \vec{x}.(\exists \vec{y}.\phi_{\mathcal{S}}(\vec{x},\vec{y}) \rightarrow \varphi_{\mathcal{O}}(\vec{x}))$. Furthermore, a GAV mapping assertion is called *pure* if $\varphi_{\mathcal{O}}(\vec{x})$ is a conjunction of atoms without constants or variables that are repeated more than once in the body of $\varphi_{\mathcal{O}}(\vec{x})$.

A LAV mapping assertion is a GLAV mapping assertion in which the left-hand side of the implication is simply an atom without constants or repeated variables, and all universally quantified variables appear in the right-hand side of the implication, i.e., it is an assertion of the form $\forall \vec{x}.(\phi_{\mathcal{S}}(\vec{x}) \rightarrow \exists \vec{z}.\varphi_{\mathcal{O}}(\vec{x},\vec{z}))$, where $\phi_{\mathcal{S}}(\vec{x})$ corresponds to $s(x_1,\ldots,x_n)$ with s being an n-ary predicate symbol of $\mathcal{S}$, and $x_1,\ldots,x_n$ being pairwise different variables.

For readability purposes, from now on we will drop universal quantifiers in front of mapping assertions. Finally, we say that a mapping $\mathcal{M}$ is a GLAV (respectively, LAV, GAV, pure GAV) mapping if it consists of a finite set of GLAV (respectively, LAV, GAV, pure GAV) mapping assertions.

Due to the assumption that ontologies contain the atomic concept $\top$ in their alphabet (with its obvious semantics), in each mapping $\mathcal{M}$ relating a source schema $\mathcal{S}$ to an ontology $\mathcal{O}$, we tacitly assume that, for each n-ary predicate $s \in \mathcal{S}$, the mapping $\mathcal{M}$ contains the following both LAV and pure GAV mapping assertion:

$$s(x_1, x_2, \ldots, x_n) \rightarrow \top(x_1) \wedge \top(x_2) \wedge \ldots \wedge \top(x_n).$$

2.6.2 The Chase

The *chase* is a fixpoint algorithm typically used to reason about data dependencies [Beeri and Vardi, 1984]. It has been introduced in [Maier *et al.*, 1979], and now it plays a central role in various contexts, e.g., database schema design [Bernstein, 1976; Fagin, 1977], query containment [Kolaitis and Vardi, 2000], data exchange [Fagin *et al.*, 2005a; Fagin *et al.*, 2005b; Arenas *et al.*, 2014], etc.

Here, with chase we implicitly refer to the so-called *oblivious chase* [Calì *et al.*, 2013] (also known as the *naive chase* [ten Cate *et al.*, 2009]) rather than to the standard chase [Fagin *et al.*, 2005a]. Formally, given a set of atoms $\mathcal{W}$ over a schema $\mathcal{S}$ and a mapping $\mathcal{M}$ relating $\mathcal{S}$ to an ontology $\mathcal{O}$, the chase of $\mathcal{W}$ with respect to $\mathcal{M}$, denoted by $\mathcal{M}(\mathcal{W})$, is computed as follows: (i) we start with an empty set of atoms $J := \emptyset$ over $\mathcal{O}$, then (ii) for every GLAV assertion $\exists \vec{y}.\phi_{\mathcal{S}}(\vec{x}, \vec{y}) \rightarrow \exists \vec{z}.\varphi_{\mathcal{O}}(\vec{x}, \vec{z})$ in $\mathcal{M}$ and for every homomorphism h from the set of all atoms occurring in $\phi_{\mathcal{S}}(\vec{x}, \vec{y})$ to $\mathcal{W}$, we add to J the image of the set of all atoms occurring in $\varphi_{\mathcal{O}}(\vec{x}, \vec{z})$ under h', that is, $J := J \cup h'(\varphi_{\mathcal{O}}(\vec{x}, \vec{z}))$, where h' extends h by assigning to each variable $z \in \vec{z}$ a different fresh variable in Var still not present in J.

With a slight abuse of notation, given a CQ$^{\neq}$ q over a schema $\mathcal{S}$ and a mapping $\mathcal{M}$ relating schema $\mathcal{S}$ to an ontology $\mathcal{O}$, we denote by $\mathcal{M}(q)$ the conjunction of all the atoms in $\mathcal{M}(\mathcal{W}_q)$, i.e., the conjunction of all the atoms obtained by chasing the incomplete $\mathcal{S}$-database associated to q with respect to $\mathcal{M}$.

2.6.3 Semantics and Query Answering

The semantics of an OBDM system $\langle \Sigma, D \rangle$, with $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ an OBDM specification and D an $\mathcal{S}$-database, is given in terms of interpretations $\mathcal{I}$ for $\langle \Sigma, D \rangle$, i.e., 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 $a \in \mathsf{dom}(D) \cup \mathsf{con}_{\mathcal{M}}$ a domain object $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, where $\mathsf{con}_{\mathcal{M}}$ denotes the set of all constants occurring in $\mathcal{M}$. Unless otherwise stated, we adopt the UNA, i.e., we consider only those interpretations $\mathcal{I}$ for which $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ for each pair of constants $a, b \in \mathsf{dom}(D) \cup \mathsf{con}_{\mathcal{M}}$ with $a \neq b$.

Given an OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, an $\mathcal{S}$-database D, an interpretation $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ for $\langle \Sigma, D \rangle$, and a GLAV mapping assertion $m = \exists \vec{y}.\phi_{\mathcal{S}}(\vec{x}, \vec{y}) \rightarrow \exists \vec{z}.\varphi_{\mathcal{O}}(\vec{x}, \vec{z})$ belonging to a mapping $\mathcal{M}$ relating schema $\mathcal{S}$ to $\mathcal{O}$, we say that the pair $\langle D, \mathcal{I} \rangle$ satisfies m if the following holds: for each homomorphism h from $\phi_{\mathcal{S}}(\vec{x}, \vec{y})$ to D, there exists a homomorphism h' from $\varphi_{\mathcal{O}}^{h_{\mathcal{I}}}(\vec{x}, \vec{z})$ to $\mathcal{I}$, where $\varphi_{\mathcal{O}}^{h_{\mathcal{I}}}(\vec{x}, \vec{z})$ denotes the set of all atoms occurring in $\varphi_{\mathcal{O}}(\vec{x}, \vec{z})$ obtained by replacing each term t, that is either a constant or $t \in \vec{x}$, with $h(t)^{\mathcal{I}}$ (by definition, $h(t)$ is either a constant occurring in D or a constant occurring in $\mathcal{M}$, and therefore $h(t)^{\mathcal{I}}$ is a domain object belonging to $\Delta^{\mathcal{I}}$). Furthermore, we say that $\langle D, \mathcal{I} \rangle$ satisfies a mapping $\mathcal{M}$ relating schema $\mathcal{S}$ to $\mathcal{O}$, denoted by $\langle D, \mathcal{I} \rangle \models \mathcal{M}$, if $\langle D, \mathcal{I} \rangle$ satisfies every assertion $m \in \mathcal{M}$.

We are now ready to formalise the notion of model of an OBDM system. An interpretation $\mathcal{I}$ for an OBDM system $\langle \Sigma, D \rangle$ is a model of $\langle \Sigma, D \rangle$ (also called a *model of* Σ *relative to* D), denoted by $\mathcal{I} \models \langle \Sigma, D \rangle$, if (i) $\mathcal{I} \models \mathcal{O}$ and (ii) $\langle D, \mathcal{I} \rangle \models \mathcal{M}$.

The set of *models* of an OBDM system $\langle \Sigma, D \rangle$, denoted by $Mod_D(\Sigma)$, is the set of interpretations $\mathcal{I}$ for $\langle \Sigma, D \rangle$ such that $\mathcal{I} \models \langle \Sigma, D \rangle$. An $\mathcal{S}$-database D is said to be *consistent with* $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ if $Mod_D(\Sigma) \neq \emptyset$, *inconsistent* otherwise.

In OBDM one of the main service of interest is query answering, i.e., computing the certain answers to queries posed over the ontology (cf. Figure 2.1). Given an OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, an $\mathcal{S}$-database D, and an UCQ$^{\neq}$ query $q_{\mathcal{O}}$ over $\mathcal{O}$, we denote by $cert^{D}_{q_{\mathcal{O}},\Sigma}$ the set of *certain answers* of $q_{\mathcal{O}}$ with respect to Σ and D, i.e., the set of tuples of constants $(c_1, \ldots, c_n)$ such that $(c_1^{\mathcal{I}}, \ldots, c_n^{\mathcal{I}}) \in q_{\mathcal{O}}^{\mathcal{I}}$ for each $\mathcal{I} \in Mod_D(\Sigma)$. Observe that, if $\langle \Sigma, D \rangle$ is unsatisfiable, then the set of certain answers of any query $q_{\mathcal{O}}$ (over $\mathcal{O}$) with respect to Σ and D is trivially the set of all possible tuples of constants occurring in D whose arity is the one of the query.

Given an OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and two queries $q^1_{\mathcal{O}}, q^2_{\mathcal{O}}$ over $\mathcal{O}$, we write $cert_{q^1_{\mathcal{O}},\Sigma} \sqsubseteq cert_{q^2_{\mathcal{O}},\Sigma}$ if $cert^{D}_{q^1_{\mathcal{O}},\Sigma} \subseteq cert^{D}_{q^2_{\mathcal{O}},\Sigma}$ for each $\mathcal{S}$-database D. We also write $cert_{q^1_{\mathcal{O}},\Sigma} \sqsubset cert_{q^2_{\mathcal{O}},\Sigma}$ if (i) $cert_{q^1_{\mathcal{O}},\Sigma} \sqsubseteq cert_{q^2_{\mathcal{O}},\Sigma}$, and in addition (ii) $cert^{D}_{q^1_{\mathcal{O}},\Sigma} \subsetneq cert^{D}_{q^2_{\mathcal{O}},\Sigma}$ for at least an $\mathcal{S}$-database D. Finally, we say that $q^1_{\mathcal{O}}$ and $q^2_{\mathcal{O}}$ are *equivalent with respect to* Σ, denoted by $cert_{q^1_{\mathcal{O}},\Sigma} \equiv cert_{q^2_{\mathcal{O}},\Sigma}$, if both $cert_{q^1_{\mathcal{O}},\Sigma} \sqsubseteq cert_{q^2_{\mathcal{O}},\Sigma}$ and $cert_{q^2_{\mathcal{O}},\Sigma} \sqsubseteq cert_{q^1_{\mathcal{O}},\Sigma}$ hold, that is, $cert^{D}_{q^1_{\mathcal{O}},\Sigma} = cert^{D}_{q^2_{\mathcal{O}},\Sigma}$ for each $\mathcal{S}$-database D.

For an OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and a query $q_{\mathcal{O}}$ over $\mathcal{O}$, following the literature terminology, we say that a query $q_{\mathcal{S}}$ over $\mathcal{S}$ is a *perfect* (respectively, *sound*) $\mathcal{O}$*-to-*$\mathcal{S}$ Σ*-rewriting* of $q_{\mathcal{O}}$ if $q^D_{\mathcal{S}} = cert^{D}_{q_{\mathcal{O}},\Sigma}$ (respectively, $q^D_{\mathcal{S}} \subseteq cert^{D}_{q_{\mathcal{O}},\Sigma}$) for each $\mathcal{S}$-database D. The perfect $\mathcal{O}$-to-$\mathcal{S}$ Σ-rewriting of $q_{\mathcal{O}}$ is denoted by $\mathsf{REW}_{q_{\mathcal{O}},\Sigma}$. Observe that, by definition, $\mathsf{REW}^{D}_{q_{\mathcal{O}},\Sigma} = cert^{D}_{q_{\mathcal{O}},\Sigma}$ for each $\mathcal{S}$-database D.

Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification where $\mathcal{O} = \emptyset$, i.e., $\mathcal{O}$ has no assertions, and $\mathcal{M}$ is a GLAV mapping. From result of [Friedman *et al.*, 1999; Calvanese *et al.*, 2012], 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 $\mathsf{MapRef}(q_{\mathcal{O}}, \mathcal{M})$, such that $\mathsf{MapRef}(q_{\mathcal{O}}, \mathcal{M}) \equiv \mathsf{REW}_{q_{\mathcal{O}},\Sigma}$.

We conclude this chapter with the following observations for OBDM specifications $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ where $\mathcal{O}$ is a *DL-Lite*$_{\mathcal{R}}$ ontology and $\mathcal{M}$ is a GLAV mapping:

- Given a UCQ $q_{\mathcal{O}}$ over $\mathcal{O}$, we denote by $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ the UCQ obtained by first executing the algorithm $\mathsf{PerfectRef}$ on $\mathcal{O}$ and $q_{\mathcal{O}}$, and then by rewriting the obtained UCQ with respect to mapping $\mathcal{M}$, i.e., $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} := \mathsf{MapRef}(\mathsf{PerfectRef}(\mathcal{O}, q_{\mathcal{O}}), \mathcal{M})$. From results of Subsection 2.5.3 and the above observation, it is easy to see that, if $q_{\mathcal{O}}$ is a UCQ over $\mathcal{O}$ of arity n, then $\mathsf{REW}_{q_{\mathcal{O}},\Sigma} \equiv \mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},\Sigma}$. In other words, if $q_{\mathcal{O}}$ is a UCQ over $\mathcal{O}$, then the UCQ $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},\Sigma}$ is the perfect $\mathcal{O}$-to-$\mathcal{S}$ Σ-rewriting of $q_{\mathcal{O}}$, i.e, $(\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},\Sigma})^D = cert^{D}_{q_{\mathcal{O}},\Sigma}$ for every $\mathcal{S}$-database D.

- For an $\mathcal{S}$-database D, we denote by $\mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}}$ the *canonical structure of* $\mathcal{O}$ *with respect to* $\mathcal{M}$ *and* D, i.e., the (possibly infinite) set of atoms over $\mathcal{O}$ obtained by first chasing D with respect to $\mathcal{M}$, and then by chasing, possibly ad infinitum, the resulting set of atoms $\mathcal{M}(D)$ with respect to $\mathcal{O}$ as described in [Calvanese *et al.*, 2007b, Definition 5] but using the alphabet Var of variables whenever a new element is needed in the chase. By combining results of [Fagin *et al.*, 2005a, Proposition 4.2] with [Calvanese *et al.*, 2007b, Theorem 29], it is well-known that, if $q_{\mathcal{O}}$ is a UCQ over $\mathcal{O}$ and $\vec{c}$ is a tuple of constants, then $\vec{c} \in cert^{D}_{q_{\mathcal{O}},\Sigma}$ if and only if $\vec{c} \in q_{\mathcal{O}}^{\mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}}}$, i.e., if and only if $\mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}} \models q_{\mathcal{O}}(\vec{c})$.

Chapter 3

Abstraction in Ontology-based Data Management: Framework

In this chapter, we illustrate a formal framework for abstraction in OBDM. Informally, given a data service expressed as a query $q_{\mathcal{S}}$ over the source schema, the goal is in finding a query $q_{\mathcal{O}}$ over the ontology that represents an abstraction of the data service represented by $q_{\mathcal{S}}$ in terms of the domain ontology through the mapping.

Specifically, we introduce three semantically well-founded notions, namely *perfect*, *sound*, and *complete source-to-ontology rewriting*, and two basic computational problems, namely *verification* and *computation*. In what follows, $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ refers to an OBDM specification, and $q_{\mathcal{S}}$ and $q_{\mathcal{O}}$ to queries over the source schema $\mathcal{S}$ and over the ontology $\mathcal{O}$, respectively, of the same arity.

3.1 The notion of Source-to-Ontology Rewriting

Intuitively, given a data service expressed as a query $q_{\mathcal{S}}$ over $\mathcal{S}$, we aim at finding the query $q_{\mathcal{O}}$ over $\mathcal{O}$ that semantically characterises $q_{\mathcal{S}}$ with respect to the OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ of the information system. Since the evaluation of queries over $\mathcal{O}$ is based on certain answers, this means that we aim at finding a query over $\mathcal{O}$ whose certain answers with respect to Σ and D precisely capture the evaluation of $q_{\mathcal{S}}$ over D, for every $\mathcal{S}$-database D. Therefore, we are naturally led to the notion of perfect source-to-ontology rewriting.

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

Example 3.1. Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, $q_{\mathcal{S}}$, and $q_{\mathcal{O}}$ be the OBDM specification, the query over $\mathcal{S}$, and the query over $\mathcal{O}$, respectively, illustrated in Example 1.1. One can easily verify that $q_{\mathcal{O}}$ is a CQ-perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. □

The next proposition states that perfect source-to-ontology rewritings are always unique (up to equivalence with respect to the underlying OBDM specification Σ).

Proposition 3.1. *If q_1 and q_2 are perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewritings of $q_{\mathcal{S}}$, then they are equivalent with respect to Σ, i.e., $cert_{q_1,\Sigma} \equiv cert_{q_2,\Sigma}$.*

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

The next example shows that perfect source-to-ontology rewritings are not guaranteed to exists, even in trivial cases.

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

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

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

Let the data service be expressed as the query $q_{\mathcal{S}} = \{(x) \mid s_1(x)\}$ over the source schema $\mathcal{S}$. By inspecting the mapping $\mathcal{M}$ and the ontology $\mathcal{O}$ one can see that, since the certain answers of $q^1_{\mathcal{O}} = \{(x) \mid \mathsf{Worker}(x)\}$ include also the values stored in the projection on the first component of s_2, and since the certain answers of $q^2_{\mathcal{O}} = \{(x) \mid \mathsf{Student}(x)\}$ include also the values stored in s_3, both queries are too general for exactly characterising $q_{\mathcal{S}}$. On the other hand, queries $q^3_{\mathcal{O}} = \{(x) \mid \mathsf{Engineer}(x)\}$ and $q^4_{\mathcal{O}} = \{(x) \mid \exists y.\mathsf{PlaysSport}(x, y)\}$ are too specific, and therefore we conclude that no perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ exists. □

In order to cope with the situations illustrated above, we introduce the notions of sound and complete source-to-ontology rewritings, which, intuitively, provide sound and complete approximations of perfect source-to-ontology rewritings, respectively.

Definition 3.2. We say that $q_{\mathcal{O}}$ is a *sound* $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, if for every $\mathcal{S}$-database D, $Mod_D(\Sigma) \neq \emptyset$ implies $cert^D_{q_{\mathcal{O}},\Sigma} \subseteq q^D_{\mathcal{S}}$.

Definition 3.3. We say that $q_{\mathcal{O}}$ is a *complete* $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, if for every $\mathcal{S}$-database D, $Mod_D(\Sigma) \neq \emptyset$ implies $q^D_{\mathcal{S}} \subseteq cert^D_{q_{\mathcal{O}},\Sigma}$.

We illustrate these notions continuing on the previous example.

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

Obviously, $q_{\mathcal{O}}$ is a perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ if and only if $q_{\mathcal{O}}$ is both a sound, and a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

As Example 3.3 shows, different sound or complete source-to-ontology rewritings of a query $q_{\mathcal{S}}$ may exist, and therefore it is reasonable to look for the "best" approximations of $q_{\mathcal{S}}$, at least relative to a certain query language $\mathcal{L}_{\mathcal{O}}$ over $\mathcal{O}$.

Definition 3.4. We say that $q_{\mathcal{O}} \in \mathcal{L}_{\mathcal{O}}$ is an $\mathcal{L}_{\mathcal{O}}$*-maximally sound* $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, if $q_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ and there exists no $q' \in \mathcal{L}_{\mathcal{O}}$ such that (i) q' is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, (ii) $cert_{q_{\mathcal{O}},\Sigma} \sqsubseteq cert_{q',\Sigma}$, and (iii) $cert^D_{q_{\mathcal{O}},\Sigma} \subsetneq cert^D_{q',\Sigma}$ for an $\mathcal{S}$-database D.

Definition 3.5. We say that $q_{\mathcal{O}} \in \mathcal{L}_{\mathcal{O}}$ is an $\mathcal{L}_{\mathcal{O}}$*-minimally complete* $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, if $q_{\mathcal{O}}$ is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ and there exists no $q' \in \mathcal{L}_{\mathcal{O}}$ such that (i) q' is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, (ii) $cert_{q',\Sigma} \sqsubseteq cert_{q_{\mathcal{O}},\Sigma}$, and (iii) $cert^D_{q',\Sigma} \subsetneq cert^D_{q_{\mathcal{O}},\Sigma}$ for an $\mathcal{S}$-database D.

We illustrate the above notions by developing on Example 3.2.

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

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

Definition 3.6. We say that $q_{\mathcal{O}} \in \mathcal{L}_{\mathcal{O}}$ is the *unique (up to equivalence w.r.t.* Σ*)* $\mathcal{L}_{\mathcal{O}}$*-maximally sound* $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, if (i) $q_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, and (ii) every $q' \in \mathcal{L}_{\mathcal{O}}$ that is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ is such that $cert_{q',\Sigma} \sqsubseteq cert_{q_{\mathcal{O}},\Sigma}$.

Definition 3.7. We say that $q_{\mathcal{O}} \in \mathcal{L}_{\mathcal{O}}$ is the *unique (up to equivalence w.r.t.* Σ*)* $\mathcal{L}_{\mathcal{O}}$*-minimally complete* $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, if (i) $q_{\mathcal{O}}$ is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, and (ii) every $q' \in \mathcal{L}_{\mathcal{O}}$ that is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ is such that $cert_{q_{\mathcal{O}},\Sigma} \sqsubseteq cert_{q',\Sigma}$.

Let us continue on the running example of this section.

Example 3.5. Refer again to Example 3.2, and consider also the queries $q^5_{\mathcal{O}}$ and $q^6_{\mathcal{O}}$ defined in Example 3.4. It is straightforward to verify that $q^5_{\mathcal{O}}$ (respectively, $q^6_{\mathcal{O}}$) is the unique (up to equivalence w.r.t. Σ) UCQ-minimally complete (respectively, UCQ-maximally sound) $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. Furthermore, observe that: (i) since $q^5_{\mathcal{O}}$ is a CQ, it is also the unique (up to equivalence w.r.t. Σ) CQ-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, and (ii) since both $q^3_{\mathcal{O}}$ and $q^4_{\mathcal{O}}$ are CQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewritings of $q_{\mathcal{S}}$ and they are not equivalent w.r.t. Σ, we conclude that the unique (up to equivalence w.r.t. Σ) CQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ does not exists. $\square$

3.2 Computational Problems

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

- *Verification*: given $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, $q_{\mathcal{S}}$ over $\mathcal{S}$ such that $q_{\mathcal{S}} \in \mathcal{L}_{\mathcal{S}}$, and $q_{\mathcal{O}}$ over $\mathcal{O}$ such that $q_{\mathcal{O}} \in \mathcal{L}_{\mathcal{O}}$, verify whether $q_{\mathcal{O}}$ is a perfect (respectively, sound, complete) $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.
- *Computation*: given $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and $q_{\mathcal{S}}$ over $\mathcal{S}$ such that $q_{\mathcal{S}} \in \mathcal{L}_{\mathcal{S}}$, compute any perfect (respectively, $\mathcal{L}_{\mathcal{O}}$-perfect, $\mathcal{L}_{\mathcal{O}}$-maximally sound, $\mathcal{L}_{\mathcal{O}}$-minimally complete) $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, if it exists.

In what follows, if not otherwise stated, we silently refer to the following scenario which use languages amongst the most popular considered in the literature: (i) the setting for OBDM specifications is such that the DL ontology language is *DL-Lite*$_{\mathcal{R}}$, the source schemas do not have integrity constraints, and the mapping language follows the GLAV approach; and (ii) both $\mathcal{L}_{\mathcal{S}}$ and $\mathcal{L}_{\mathcal{O}}$ denote the class of UCQs.

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

Proposition 3.2. *If q_1 and q_2 are UCQ-minimally complete (respectively, UCQ-maximally sound) $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewritings of $q_{\mathcal{S}}$, 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 $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewritings of $q_{\mathcal{S}}$ and suppose, for the sake of contradiction, that they are not equivalent w.r.t. Σ. This implies the existence of two $\mathcal{S}$-databases D_1 and D_2 such that (i) $\vec{c_1} \notin cert_{q_1,\Sigma}^{D_1}$ and $\vec{c_1} \in cert_{q_2,\Sigma}^{D_1}$ for a tuple of constants $\vec{c_1}$, and (ii) $\vec{c_2} \notin cert_{q_2,\Sigma}^{D_2}$ and $\vec{c_2} \in cert_{q_1,\Sigma}^{D_2}$ for a tuple of constants $\vec{c_2}$. But then, it can be readily seen that the UCQ $Q = q_1 \cup q_2$ is such that (i) since both q_1 and q_2 are sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewritings of $q_{\mathcal{S}}$, Q is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, (ii) for both $i = 1$ and $i = 2$, we have $cert_{q_i,\Sigma} \sqsubseteq cert_{Q,\Sigma}$, and ($iii$) for both $i = 1$ and $i = 2$, the $\mathcal{S}$-database D_i is such that $cert_{q_i,\Sigma}^{D_i} \subsetneq cert_{Q,\Sigma}^{D_i}$. Therefore, according to Definition 3.4 we have a contradiction on the fact that q_1 and q_2 are UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewritings of $q_{\mathcal{S}}$, as required.

Assume that q_1 and q_2 are UCQ-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewritings of $q_{\mathcal{S}}$ and suppose, for the sake of contradiction, that they are not equivalent w.r.t. Σ. This implies the existence of two $\mathcal{S}$-databases D_1 and D_2 such that (i) $\vec{c_1} \in cert_{q_1,\Sigma}^{D_1}$ and $\vec{c_1} \notin cert_{q_2,\Sigma}^{D_1}$ for a tuple of constants $\vec{c_1}$, and (ii) $\vec{c_2} \in cert_{q_2,\Sigma}^{D_2}$ and $\vec{c_2} \notin cert_{q_1,\Sigma}^{D_2}$ for a tuple of constants $\vec{c_2}$. But then, consider the query q where $cert_{q,\Sigma}^{D} = cert_{q_1,\Sigma}^{D} \bigcap cert_{q_2,\Sigma}^{D}$ for every $\mathcal{S}$-database D. Obviously, since q_1 and q_2 are UCQs, q always exists and can be expressed as a UCQ, too. It can be readily seen that (i) since q_1 and q_2 are complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewritings of $q_{\mathcal{S}}$, q is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ as well, (ii) for both $i = 1$ and $i = 2$, we have $cert_{q,\Sigma} \sqsubseteq cert_{q_i,\Sigma}$, and ($iii$) for both $i = 1$ and $i = 2$, the $\mathcal{S}$-database D_i is such that $cert_{q,\Sigma}^{D_i} \subsetneq cert_{q_i,\Sigma}^{D_i}$. Therefore, according to Definition 3.5 we have a contradiction on the fact that q_1 and q_2 are UCQ-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewritings of $q_{\mathcal{S}}$, as required. $\square$

3.3 Related Work

In this section, we provide a detailed relationship between the new definitions introduced in Section 3.1 and some literature notions about pertinent subjects that we argue are worth comparing to.

3.3.1 A slightly different Semantics

A similar, but not equivalent, notion of perfect source-to-ontology rewriting given in Definition 3.1 is the notion of *realization* provided in [Lutz *et al.*, 2018]. Specifically, $q_{\mathcal{O}}$ is a *realization* of $q_{\mathcal{S}}$ in Σ if $cert^{D}_{q_{\mathcal{O}},\Sigma} = q^{D}_{\mathcal{S}}$ for every $\mathcal{S}$-database D. Observe that, while the latter sanctions that $cert^{D}_{q_{\mathcal{O}},\Sigma} = q^{D}_{\mathcal{S}}$ for *all* $\mathcal{S}$-databases D, in Definition 3.1 the condition is limited only to the $\mathcal{S}$-databases D that are consistent with Σ. The different behaviour between the two notions is highlighted by the following example.

Example 3.6. Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the following OBDM specification:

- $\mathcal{O} = \{ \ \mathsf{Professor} \sqsubseteq \neg\mathsf{Student} \ \}$
- $\mathcal{S} = \{ \ s_1, s_2, s_3 \ \}$
- $\mathcal{M} = \{ \ m_1, m_2, m_3 \ \}$, where:

$$\begin{array}{llll} m_1: & s_1(x) & \rightarrow & \mathsf{Worker}(x), \\ m_2: & s_2(x) & \rightarrow & \mathsf{Professor}(x), \\ m_3: & s_3(x) & \rightarrow & \mathsf{Student}(x). \end{array}$$

Let the query over $\mathcal{S}$ be $q_{\mathcal{S}} = \{(x) \mid s_1(x)\}$, and let the query over $\mathcal{O}$ be $q_{\mathcal{O}} = \{(x) \mid \mathsf{Worker}(x)\}$. Note that (i) $\mathsf{PerfectRef}(\mathcal{O}, q_{\mathcal{O}}) = q_{\mathcal{O}}$, and therefore $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} = \mathsf{MapRef}(\mathsf{PerfectRef}(\mathcal{O}, q_{\mathcal{O}}), \mathcal{M}) = \{(x) \mid s_1(x)\}$, and (ii) $\mathcal{V}_{\mathcal{O}} = \{() \mid \exists y.\mathsf{Professor}(y) \wedge \mathsf{Student}(y)\}$, and therefore $\mathsf{PerfRef}_{\mathcal{V}_{\mathcal{O}},\Sigma} = \mathsf{MapRef}(\mathsf{PerfectRef}(\mathcal{O}, \mathcal{V}_{\mathcal{O}}), \mathcal{M}) = \{() \mid \exists y.s_2(y) \wedge s_3(y)\}$. Finally, we know that $\mathsf{REW}_{q_{\mathcal{O}},\Sigma} \equiv \mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \cup \mathsf{PerfRef}_{\mathcal{V}^1_{\mathcal{O}},\Sigma}$, where $\mathcal{V}^1_{\mathcal{O}} = \{(x) \mid \exists y.\mathsf{Professor}(y) \wedge \mathsf{Student}(y) \wedge \top(x)\}$.

Consider now the $\mathcal{S}$-database $D = \{s_1(c_1), s_2(c_2), s_3(c_2)\}$. We have that $q^{D}_{\mathcal{S}} = \{(c_1)\}$, whereas, since D is inconsistent with Σ (i.e., $Mod_D(\Sigma) = \emptyset$), $cert^{D}_{q_{\mathcal{O}},\Sigma} = \mathsf{PerfRef}^{D}_{\mathcal{V}^1_{\mathcal{O}},\Sigma}$, and therefore the set of certain answers of $q_{\mathcal{O}}$ with respect to Σ and D contains *all* the 1-tuples of constants in $\mathsf{dom}(D) = \{c_1, c_2\}$ (hence including the tuple (c_2)). It follows that, according to the semantics proposed in [Lutz *et al.*, 2018] (which ranges over all $\mathcal{S}$-databases), $q_{\mathcal{O}}$ is not a *realization* of $q_{\mathcal{S}}$ in Σ, whereas, according to Definition 3.1, $q_{\mathcal{O}}$ is a perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ because $q^{D}_{\mathcal{S}} = cert^{D}_{q_{\mathcal{O}},\Sigma}$ for each $\mathcal{S}$-database D consistent with Σ. □

Notice, however, that the two notions turn out to be in fact equivalent when dealing with OBDM specifications where inconsistencies can not arise. For instance, OBDM specifications of our setting where the DL ontology language *DL-Lite*$_{\mathcal{R}}$ is replaced with an ontology language not able to express inconsistencies, such as the DL *DL-Lite*$_{\mathsf{RDFS}}$ and the DLs $\mathcal{EL}$ and $\mathcal{ELHI}$ [Baader *et al.*, 2005] considered in [Lutz *et al.*, 2018].

3.3.2 Relationship with Ontology-to-Source Rewritings

As argued in the introduction, most of (if not all) the literature about managing data sources through an ontology, or more generally, about data integration, assume that the user query is expressed over a global domain schema, and the goal is to find an ontology-to-source rewriting (i.e., a query over the source schema) that captures the original query in the best way, independently from the current source database. The framework just introduced can be therefore seen as a sort of reverse engineering problem, because we start with a source query and we aim at deriving a corresponding query over the ontology, called a source-to-ontology rewriting. Here, we make explicit the relationships between these new notions of source-to-ontology rewritings and the usual notions of ontology-to-source rewritings studied in OBDM.

Theorem 3.1. *Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification, and let $q_{\mathcal{S}}$ and $q_{\mathcal{O}}$ be queries over $\mathcal{S}$ and over $\mathcal{O}$, respectively. We have that:*

1. *$q_{\mathcal{S}}$ is a sound $\mathcal{O}$-to-$\mathcal{S}$ Σ-rewriting of $q_{\mathcal{O}}$ if and only if $q_{\mathcal{O}}$ is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$;*

2. *If $q_{\mathcal{S}}$ is a perfect $\mathcal{O}$-to-$\mathcal{S}$ Σ-rewriting of $q_{\mathcal{O}}$, then $q_{\mathcal{O}}$ is a perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. The converse does not necessarily hold.*

Proof. As for 1, by definition $q_{\mathcal{S}}$ is a sound $\mathcal{O}$-to-$\mathcal{S}$ Σ-rewriting of $q_{\mathcal{O}}$ if and only if $q_{\mathcal{S}}^D \subseteq cert_{q_{\mathcal{O}},\Sigma}^D$ for every $\mathcal{S}$-database D. Since for all the $\mathcal{S}$-databases D that are inconsistent with Σ the above inclusion trivially holds, this is equivalent to the condition $q_{\mathcal{S}}^D \subseteq cert_{q_{\mathcal{O}},\Sigma}^D$ for every $\mathcal{S}$-database D consistent with Σ, which is exactly the definition of $q_{\mathcal{O}}$ being a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ (cf. Definition 3.3).

As for the implication part of 2, by definition $q_{\mathcal{S}}$ is a perfect $\mathcal{O}$-to-$\mathcal{S}$ Σ-rewriting of $q_{\mathcal{O}}$ if and only if $q_{\mathcal{S}}^D = cert_{q_{\mathcal{O}},\Sigma}^D$ for every $\mathcal{S}$-database D, which obviously implies that $q_{\mathcal{O}}$ is a perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

To show that the converse does not necessarily hold, consider $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, $q_{\mathcal{S}}$, and $q_{\mathcal{O}}$ as described in Example 3.6. In particular, $q_{\mathcal{O}}$ is a perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, whereas $q_{\mathcal{S}}$ is not a perfect $\mathcal{O}$-to-$\mathcal{S}$ Σ-rewriting of $q_{\mathcal{O}}$. □

Once again, it is easy to see that the converse statement of point 2 of the above theorem becomes in fact true when dealing with ontologies expressed in those DLs where inconsistencies can not arise.

3.3.3 Relationship with View-based Query Processing

In order to establish a relationship between the notion of source-to-ontology rewritings and the view-based query processing approach, here we consider OBDM specifications $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ where $\mathcal{O} = \emptyset$ and mapping $\mathcal{M}$ is a pure GAV mapping.

As a first consideration, observe that every pure GAV mapping $\mathcal{M}$ can be equivalently rewritten as a set of pure GAV mapping assertions of the form $\exists \vec{y}. \phi_{\mathcal{S}}(\vec{x}, \vec{y}) \rightarrow \varphi_{\mathcal{O}}(\vec{x})$, where $\varphi_{\mathcal{O}}(\vec{x})$ is simply an atom without constants or repeated variables, i.e., $\varphi_{\mathcal{O}}(\vec{x})$ is either of the form $A(x)$ for an atomic concept A in the alphabet of $\mathcal{O}$, or of the form $P(x_1, x_2)$ for an atomic role P in the alphabet of $\mathcal{O}$ and with x_1 and x_2 being different variables.

Example 3.7. Let $\mathcal{M}$ be composed of the following pure GAV mapping assertions:

$$\begin{array}{llcl} m_1: & \exists y_1, y_2 . s_1(y_1, x_1, x_3) \wedge s_2(x_3, x_2, y_2) & \rightarrow & P_1(x_1, x_2) \wedge P_2(x_1, x_3), \\ m_2: & \exists y_1 . s_3(x_1, x_2, y_1) & \rightarrow & P_2(x_1, x_2). \end{array}$$

Then, $\mathcal{M}$ is equivalent to the following set of pure GAV mapping assertions:

$$\begin{array}{llcl} m_1: & \exists y_1, y_2, y_3 . s_1(y_1, x_1, y_3) \wedge s_2(y_3, x_2, y_2) & \rightarrow & P_1(x_1, x_2), \\ m_2: & \exists y_1, y_2, y_3 . s_1(y_1, x_1, x_2) \wedge s_2(x_2, y_3, y_2) & \rightarrow & P_2(x_1, x_2), \\ m_3: & \exists y_1 . s_3(x_1, x_2, y_1) & \rightarrow & P_2(x_1, x_2). \end{array}$$

□

Let $\mathcal{M}$ be a pure GAV mapping relating a schema $\mathcal{S}$ to an ontology $\mathcal{O}$, and let A (respectively, P) be an atomic concept (respectively, atomic role) in the alphabet of $\mathcal{O}$. We denote by V_A (respectively, V_P) the following UCQ over $\mathcal{S}$:

$$V_A = \{(x) \mid \exists \vec{y_1} . \phi^1_{\mathcal{S}}(x, \vec{y_1})\} \cup \ldots \cup \{(x) \mid \exists \vec{y_{l_A}} . \phi^{l_A}_{\mathcal{S}}(x, \vec{y_{l_A}})\}$$

$$V_P = \{(x_1, x_2) \mid \exists \vec{y_1} . \phi^1_{\mathcal{S}}(x_1, x_2, \vec{y_1})\} \cup \ldots \cup \{(x_1, x_2) \mid \exists \vec{y_{l_P}} . \phi^{l_P}_{\mathcal{S}}(x_1, x_2, \vec{y_{l_P}})\},$$

where $\exists \vec{y_i} . \phi^i_{\mathcal{S}}(x, \vec{y_i}) \rightarrow A(x)$ (respectively, $\exists \vec{y_i} . \phi^i_{\mathcal{S}}(x_1, x_2, \vec{y_i}) \rightarrow P(x_1, x_2)$) is a mapping assertion in $\mathcal{M}$ for each $i \in [1, l_A]$ (respectively, for each $i \in [1, l_P]$). Furthermore, we denote by $\mathcal{V}_{\mathcal{M}} = \{V_{A_1}, \ldots, V_{A_n}, V_{P_1}, \ldots, V_{P_m}\}$ the set of UCQ view definitions over schema $\mathcal{S}$ obtained by associating the UCQ V_A (respectively, V_P) to each atomic concept A (respectively, atomic role P) in the alphabet of $\mathcal{O}$. Finally, given a query $q_{\mathcal{O}}$ over $\mathcal{O}$, we denote by $q_{\mathcal{O}}^{\mathcal{V}_{\mathcal{M}}}$ the query over the view alphabet $\mathcal{V}_{\mathcal{M}}$ obtained by replacing each predicate name A (respectively, P) with V_A (respectively, V_P).

Example 3.8. Let $\mathcal{M} = \{m_1, m_2, m_3\}$ be the pure GAV mapping defined in Example 3.7, and let $q_{\mathcal{O}} = \{(x) \mid \exists y . P_2(x, y)\}$. We have that $\mathcal{V}_{\mathcal{M}} = \{V_{P_1}, V_{P_2}\}$, where $V_{P_1} = \{(x) \mid \exists y_1, y_2, y_3 . s_1(y_1, x_1, y_3) \wedge s_2(y_3, x_2, y_2)\}$ and $V_{P_2} = \{(x_1, x_2) \mid \exists y_1, y_2, y_3 . s_1(y_1, x_1, x_2) \wedge s_2(x_2, y_3, y_2)\} \cup \{(x_1, x_2) \mid \exists y_1 . s_3(x_1, x_2, y_1)\}$, and $q_{\mathcal{O}}^{\mathcal{V}_{\mathcal{M}}} = \{(x) \mid \exists y . V_{P_2}(x, y)\}$.

We are now ready to draw the correspondence between the notions of source-to-ontology rewritings introduced in Section 3.1 and the usual notions of rewritings with respect to view definitions (cf. Section 2.4).

Theorem 3.2. *Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification where $\mathcal{O} = \emptyset$ and $\mathcal{M}$ is a pure GAV mapping, and let $q_{\mathcal{S}}$ and $q_{\mathcal{O}}$ be two UCQs over $\mathcal{S}$ and $\mathcal{O}$, respectively. We have that $q_{\mathcal{O}}$ is a perfect (respectively, sound) $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ if and only if $q_{\mathcal{O}}^{\mathcal{V}_{\mathcal{M}}}$ is an exact (respectively, a sound) rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}_{\mathcal{M}}$.*

Proof. The proof is based on the following three observations:

1. By [Levy *et al.*, 1995], a UCQ q is an exact (respectively, a sound) rewriting of a UCQ $q_{\mathcal{S}}$ with respect to a set of UCQ view definitions $\mathcal{V}$ if and only if $exp_{\mathcal{V}}(q) \equiv q_{\mathcal{S}}$ (respectively, $exp_{\mathcal{V}}(q) \sqsubseteq q_{\mathcal{S}}$), where $exp_{\mathcal{V}}(\cdot)$ is the function that, given a UCQ q over the view alphabet, replace each atom occurring in q by the definition of the views (being careful to use unique variables in place of

those variables that appear in the bodies of the view but not in the heads of those), and then turning the resulting formula into an equivalent UCQ (e.g., $exp_{\mathcal{V}_{\mathcal{M}}}(q_{\mathcal{O}}^{\mathcal{V}_{\mathcal{M}}}) = \{(x) \mid \exists y, y_1, y_2, y_3. s_1(y_1, x, y) \wedge s_2(y, y_3, y_2)\} \cup \{(x) \mid \exists y, y_1. s_3(x, y, y_1)\}$, where $q_{\mathcal{O}}^{\mathcal{V}_{\mathcal{M}}}$ and $\mathcal{V}_{\mathcal{M}}$ are the query and the set of view definitions, respectively, of Example 3.8).

2. For OBDM specifications $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ with $\mathcal{O} = \emptyset$ and $\mathcal{M}$ a pure GAV mapping, a UCQ $q_{\mathcal{O}}$ is a perfect (respectively, sound) $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of a UCQ $q_{\mathcal{S}}$ if and only if $\mathsf{MapRef}(q_{\mathcal{O}}, \mathcal{M}) \equiv q_{\mathcal{S}}$ (respectively, $\mathsf{MapRef}(q_{\mathcal{O}}, \mathcal{M}) \sqsubseteq q_{\mathcal{S}}$), where $\mathsf{MapRef}(q_{\mathcal{O}}, \mathcal{M})$ in this case is equivalent to *unfolding* the query $q_{\mathcal{O}}$ with respect to $\mathcal{M}$ [Poggi *et al.*, 2008], i.e., replacing each atom α occurring in $q_{\mathcal{O}}$ by the logical disjunction of all the left-hand sides of mapping assertions in $\mathcal{M}$ having the predicate name α in the right-hand side (being careful to use unique variables in place of those variables that appear in the left-hand side of the mapping assertions but not in the right-hand side of those), and then turning the resulting formula into an equivalent UCQ (e.g., $\mathsf{MapRef}(q_{\mathcal{O}}, \mathcal{M}) = \{(x) \mid \exists y, y_1, y_2, y_3. s_1(y_1, x, y) \wedge s_2(y, y_3, y_2)\} \cup \{(x) \mid \exists y, y_1. s_3(x, y, y_1)\}$, where $q_{\mathcal{O}}$ and $\mathcal{M}$ are the query and the pure GAV mapping, respectively, of Example 3.8).

3. By construction, $\mathsf{MapRef}(q_{\mathcal{O}}, \mathcal{M}) = exp_{\mathcal{V}_{\mathcal{M}}}(q_{\mathcal{O}}^{\mathcal{V}_{\mathcal{M}}})$ for any UCQ $q_{\mathcal{O}}$ over an ontology $\mathcal{O}$ and for any pure GAV mapping $\mathcal{M}$ relating a schema $\mathcal{S}$ to $\mathcal{O}$.

Thus, $q_{\mathcal{O}}$ is a perfect (respectively, sound) $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ if and only if $\mathsf{MapRef}(q_{\mathcal{O}}, \mathcal{M}) \equiv q_{\mathcal{S}}$ (respectively, $\mathsf{MapRef}(q_{\mathcal{O}}, \mathcal{M}) \sqsubseteq q_{\mathcal{S}}$), which, since $\mathsf{MapRef}(q_{\mathcal{O}}, \mathcal{M}) = exp_{\mathcal{V}_{\mathcal{M}}}(q_{\mathcal{O}}^{\mathcal{V}_{\mathcal{M}}})$, it is so if and only if $exp_{\mathcal{V}_{\mathcal{M}}}(q_{\mathcal{O}}^{\mathcal{V}_{\mathcal{M}}}) \equiv q_{\mathcal{S}}$ (respectively, $exp_{\mathcal{V}_{\mathcal{M}}}(q_{\mathcal{O}}^{\mathcal{V}_{\mathcal{M}}}) \sqsubseteq q_{\mathcal{S}}$), and therefore if and only if $q_{\mathcal{O}}^{\mathcal{V}_{\mathcal{M}}}$ is an exact (respectively, a sound) rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}_{\mathcal{M}}$, as required. □

From the sound part of the above theorem, we derive the following corollary.

Corollary 3.1. *Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification where $\mathcal{O} = \emptyset$ and $\mathcal{M}$ is a pure GAV mapping, and let $q_{\mathcal{S}}$ and $q_{\mathcal{O}}$ be two UCQs over $\mathcal{S}$ and $\mathcal{O}$, respectively. We have that $q_{\mathcal{O}}$ is the unique (up to equivalence w.r.t. Σ) UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ if and only if $q_{\mathcal{O}}^{\mathcal{V}_{\mathcal{M}}}$ is a UCQ-maximally sound rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}_{\mathcal{M}}$.*

In order to continue deriving further connections, we explore the following result.

Theorem 3.3. [Abiteboul and Duschka, 1998; Duschka and Genesereth, 1998] *Let $\mathcal{V}$ be a set of UCQ view definitions over a schema $\mathcal{S}$, and let $q_{\mathcal{S}}$ be a UCQ. If query $q_{\mathcal{V}}$ over the view alphabet $\mathcal{V}$ is a UCQ-maximally sound rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$, then $q_{\mathcal{V}}$ is a perfect rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$.*

The assumption that the target list of each disjunct of the various view definitions in $\mathcal{V}$ does not have repeated variables or constants is essential for the above theorem to hold. In fact, as shown in [Afrati and Chirkova, 2019] and also in the following example, the above theorem does not carry over when this assumption is removed.

Example 3.9. Consider the set $\mathcal{V} = \{V\}$ of a CQ view definition over schema $\mathcal{S} = \{s\}$ and the CQ $q_{\mathcal{S}}$ over $\mathcal{S}$, where $V = \{(x, x) \mid s(x)\}$ and $q_{\mathcal{S}} = \{(x) \mid s(x)\}$.

It is clear that $q_{\mathcal{V}} = \{(x) \mid V(x,x)\}$ is a UCQ-maximally sound rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$. Consider, however, the view extension $\mathcal{E} = \{E\}$ with $E = \{V(c_1,c_2), V(c_3,c_3)\}$. By construction of the view definition V, we have $(c_1,c_2) \notin V^D$ for each $\mathcal{S}$-database D. But then, since $V(c_1,c_2) \in \mathcal{E}$, there is no $\mathcal{S}$-database D for which $\mathcal{E} \subseteq \mathcal{V}(D)$. By definition this implies that $cert^{\mathcal{E}}_{q_{\mathcal{S}},\mathcal{V}} = \emptyset$, whereas $q^{\mathcal{E}}_{\mathcal{V}} = \{(c_3)\}$. So, $q_{\mathcal{V}}$ is not a perfect rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$. $\square$

From Corollary 3.1 and the above theorem, it is immediate to derive an interesting correspondence between the two notions under considerations.

Corollary 3.2. *Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification where $\mathcal{O} = \emptyset$ and $\mathcal{M}$ is a pure GAV mapping, and let $q_{\mathcal{S}}$ and $q_{\mathcal{O}}$ be two UCQs over $\mathcal{S}$ and $\mathcal{O}$, respectively. We have that $q_{\mathcal{O}}$ is the unique (up to equivalence w.r.t. Σ) UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ if and only if $q_{\mathcal{O}}^{\mathcal{V}_{\mathcal{M}}}$ is a perfect rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}_{\mathcal{M}}$.*

However, as shown in [Duschka and Genesereth, 1998; Afrati and Chirkova, 2019], UCQ-maximally sound rewritings of CQs $q_{\mathcal{S}}$ with respect to UCQ view definitions $\mathcal{V}$ are not guaranteed to exists. The following example gives an intuitive reason of why a UCQ-maximally sound rewriting may not exists for certain choices of $q_{\mathcal{S}}$ and $\mathcal{V}$.

Example 3.10. Consider the set $\mathcal{V} = \{V_1, V_2\}$ of UCQ view definitions over schema $\mathcal{S} = \{s_c, s_e\}$ and the CQ $q_{\mathcal{S}}$, where:

$$\begin{aligned} V_1 &= \{(x) \mid s_c(x, \mathsf{Red})\} \ \cup \ \{(x) \mid s_c(x, \mathsf{Green})\} \\ V_2 &= \{(x_1,x_2) \mid s_e(x_1,x_2)\} \\ q_{\mathcal{S}} &= \{() \mid \exists y_1, y_2, y_3. s_e(y_1,y_2) \wedge s_c(y_1,y_3) \wedge s_c(y_2,y_3)\} \end{aligned}$$

Intuitively, source predicate s_e contains the edges of a graph and mirror them into V_2, whereas V_1 will contain those vertices in s_c colored by either Red or Green. Finally, query $q_{\mathcal{S}}$ asks whether there is a pair of vertices connected by an edge and colored with the same color.

The fact that s_e may contain the edges of a non-2-colourable graph can be easily detected by certain queries issued over the view alphabet. For instance, the query $q^3_{\mathcal{V}} = \{() \mid \exists y_1, y_2, y_3. V_2(y_1,y_2) \wedge V_2(y_2,y_3) \wedge V_2(y_3,y_1) \wedge V_1(y_1) \wedge V_1(y_2) \wedge V_1(y_3)\}$ if true implies the existence of a triangle (and therefore a non-2-colourable pattern) in the graph represented by source predicates s_e and s_c.

More generally, one can verify that each query $q_{\mathcal{V}}$ over the view alphabet is a sound rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$ if and only if the graph described by $q_{\mathcal{V}}$ is not 2-colourable. Since a graph is not 2-colourable if and only if it contains an odd cycle [Asratian *et al.*, 1998], we conclude that the query $q^i_{\mathcal{V}} = \{() \mid \exists y_1, \ldots, y_i. V_2(y_1,y_2) \wedge V_2(y_2,y_3) \wedge \ldots \wedge V_2(y_{i-1},y_i) \wedge V_2(y_i,y_1) \wedge V_1(y_1) \wedge V_1(y_2) \wedge \ldots \wedge V_1(y_i)\}$ is a sound rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$, for each odd $i \geq 3$. Furthermore, the following query $q_{\mathcal{V}}$ over the view alphabet is a perfect rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$:

$$q_{\mathcal{V}} = \bigcup_{\text{odd } i \geq 3} q^i_{\mathcal{V}}$$

Obviously, such infinite union is not representable by a UCQ, and therefore we conclude that no UCQ-maximally sound rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$ exists. $\square$

This result, together with Corollary 3.1, allows us to derive the first negative result on UCQ-maximally sound source-to-ontology rewritings.

Corollary 3.3. *UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewritings of CQs $q_{\mathcal{S}}$ over $\mathcal{S}$ are not guaranteed to exists, even when OBDM specifications $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ are such that $\mathcal{O} = \emptyset$ and $\mathcal{M}$ is a pure GAV mapping.*

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

- $\mathcal{O} = \emptyset$
- $\mathcal{S} = \{ \ s_e, s_c \ \}$
- $\mathcal{M} = \{ \ m_1, m_2, m_3 \ \}$, where:

$$\begin{array}{rrcl} m_1: & s_c(x, \mathsf{Red}) & \rightarrow & A(x), \\ m_2: & s_c(x, \mathsf{Green}) & \rightarrow & A(x), \\ m_3: & s_e(x_1, x_2) & \rightarrow & P(x_1, x_2). \end{array}$$

Similarly to Example 3.10, for the CQ $q_{\mathcal{S}} = \{() \mid \exists y_1, y_2, y_3. s_e(y_1, y_2) \wedge s_c(y_1, y_3) \wedge s_c(y_2, y_3)\}$ no UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ exists. $\square$

In Example 3.10, a maximally sound rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$ does exist in the class of *Datalog* queries, non-2-colourability is indeed expressible in Datalog (see, e.g., [Kolaitis and Vardi, 2008]). Actually, if we modify the above example by considering $\mathcal{V}' = \{V_1', V_2'\}$ where $V_1' = \{(x) \mid s_c(x, \mathsf{Red})\} \cup \{(x) \mid s_c(x, \mathsf{Green})\} \cup \{(x) \mid s_c(x, \mathsf{Blue})\}$ and $V_2' = V_2$, we have that a maximally sound rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}'$ does not exist even in the class of Datalog queries. This is due to the following facts: (i) Given as input a set of view extensions $\mathcal{E} = \{E_1, E_2\}$ where E_i is the view extension of V_i' for both $i = 1$ and $i = 2$, as shown in [van der Meyden, 1993] via a reduction from non-3-colourability, the problem of checking whether $cert^{\mathcal{E}}_{q_{\mathcal{S}}, \mathcal{V}'}$ is true is coNP-hard in the size of $\mathcal{E}$;[1] (ii) Analogously to Theorem 3.3, every maximally sound rewriting of a UCQ $q_{\mathcal{S}}$ with respect to a set of UCQ view definitions $\mathcal{V}$ in the class of Datalog queries is also a perfect rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$; (iii) Answering Datalog queries over relational databases is in PTime in data complexity [Vardi, 1982]. Thus, unless PTime = NP, a maximally sound rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}'$ can not exists in the class of Datalog queries.[2]

Lastly, we observe that Example 3.10, as well as the non-existence results provided in [Duschka and Genesereth, 1998; Afrati and Chirkova, 2019], makes use of CQs $q_{\mathcal{S}}$ having more than one join existential variable in their body. On the one hand, in Section 6.2 we will strengthen such a result, by showing that it holds even for CQs $q_{\mathcal{S}}$ with only a single join existential variable. On the other hand, in Chapter 8 we will prove that having no join existential variables in the body of UCQs $q_{\mathcal{S}}$ is a sufficient condition that guarantees the existence of UCQ-maximally sound rewritings of $q_{\mathcal{S}}$ with respect to UCQ view definitions $\mathcal{V}$. Thus, due to Theorem 3.3, in such cases the

[1] More precisely, given a set of UCQ view definitions $\mathcal{V}$ over a schema $\mathcal{S}$, a set of view extensions $\mathcal{E}$, a UCQ $q_{\mathcal{S}}$ over $\mathcal{S}$, and a tuple of constants $\vec{c}$, the problem of checking whether $\vec{c} \in cert^{\mathcal{E}}_{q_{\mathcal{S}}, \mathcal{V}}$ is coNP-complete in the size of $\mathcal{E}$ [Abiteboul and Duschka, 1998].

[2] Conversely, as shown in [Duschka and Genesereth, 1998; Afrati *et al.*, 1999], maximally sound rewritings (which are again perfect rewritings) of UCQs $q_{\mathcal{S}}$ with respect to UCQ view definitions $\mathcal{V}$ always exist in the class of *Disjunctive Datalog* queries [Lobo *et al.*, 1992; Eiter *et al.*, 1997].

perfect rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$ can be always expressed as a UCQ, exactly as in the case of CQ view definitions [Levy *et al.*, 1995]. We refer to Section 8.3 for a complete discussion on this topic.

3.3.4 Inverse Mapping and/or Reversing the Arrows (Not!)

An important concept studied in the data exchange literature is the one of *inverse mapping.* In a nutshell, given a mapping $\mathcal{M}$ from a schema $\mathcal{S}$ to a schema $\mathcal{O}$, the inverse of $\mathcal{M}$, denoted by $\mathcal{M}'$, is a mapping from $\mathcal{O}$ to $\widehat{\mathcal{S}}$ that recovers *as much information as possible.* Here, $\widehat{\mathcal{S}} = \{\widehat{s} \mid s \in \mathcal{S}\}$, i.e., $\widehat{\mathcal{S}}$ is simply a *copy* of $\mathcal{S}$. According to [Fagin, 2007], $\mathcal{M}'$ is an *inverse of* $\mathcal{M}$ if and only if $\mathcal{M} \circ \mathcal{M}' \equiv \widehat{Id}$, where $\mathcal{M} \circ \mathcal{M}'$ is the composition formula [Fagin *et al.*, 2005c] and $\widehat{Id}$ is the identity mapping from $\mathcal{S}$ to $\widehat{\mathcal{S}}$, i.e., $\widehat{Id} = \{s(x_1, \dots, x_n) \rightarrow \widehat{s}(x_1, \dots, x_n) \mid s \in \mathcal{S}\}$. In other words, $\mathcal{M}'$ is an inverse of $\mathcal{M}$ if and only if $Mod_{\mathcal{M} \circ \mathcal{M}'}(D) = Mod_{\widehat{Id}}(D) = \{D' \mid \widehat{D} \subseteq D'\}$ for each $\mathcal{S}$-database D, where $\widehat{D} = \{\widehat{s}(c_1, \dots, c_n) \mid s(c_1, \dots, c_n) \in D\}$, and, for a mapping $\mathcal{M}$ relating schema $\mathcal{S}$ to schema $\widehat{\mathcal{S}}$ and an $\mathcal{S}$-database D, $Mod_{\mathcal{M}}(D)$ denotes the set of all $\widehat{S}$-databases D' for which $\langle D, D' \rangle \models \mathcal{M}$. This notion turns out to be rather restrictive, as it is rare that a schema mapping possesses an inverse.

Example 3.11. Let $\mathcal{M} = \{m_1, m_2, m_3\}$, where:

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

It is straightforward to verify that $\mathcal{M}$ does not have an inverse mapping. □

To remedy this situation, (at least) two other notions have been introduced in literature: (i) the notion of quasi-inverse which is a principled relaxation of the notion of inverse. According to [Fagin *et al.*, 2008], a mapping $\mathcal{M}'$ is a *quasi-inverse of* $\mathcal{M}$ if and only if for each pair of $\mathcal{S}$-databases D_1 and D_2 there are two $\mathcal{S}$-databases D_3 and D_4 for which $Mod_{\mathcal{M}}(D_1) = Mod_{\mathcal{M}}(D_3)$, $Mod_{\mathcal{M}}(D_2) = Mod_{\mathcal{M}}(D_4)$, and $Mod_{\mathcal{M} \circ \mathcal{M}'}(D_3) = \{D' \mid \widehat{D_4} \subseteq D'\}$; ($ii$) the notion of maximum recovery which is a strict generalisation of the notion of inverse. According to [Arenas *et al.*, 2009], $\mathcal{M}'$ is a *maximum recovery of* $\mathcal{M}$ if and only if $\langle D, \widehat{D} \rangle \models \mathcal{M} \circ \mathcal{M}'$ for each $\mathcal{S}$-database D, and for every mapping $\mathcal{M}''$ either $\langle D, \widehat{D} \rangle \not\models \mathcal{M} \circ \mathcal{M}''$ for at least an $\mathcal{S}$-database D, or $Mod_{\mathcal{M} \circ \mathcal{M}''}(D) \subseteq Mod_{\mathcal{M} \circ \mathcal{M}'}(D)$ for each $\mathcal{S}$-database D.

Example 3.12. Let $\mathcal{M}$ be the mapping as defined in Example 3.11, and consider the mapping $\mathcal{M}' = \{m'_1, m'_2, m'_3\}$, where:

$$
\begin{array}{llcl}
m'_1: & A_1(x) & \rightarrow & \widehat{s_1}(x), \\
m'_2: & A_2(x) & \rightarrow & \widehat{s_1}(x) \wedge \widehat{s_3}(x), \\
m'_3: & A_2(x) & \rightarrow & \widehat{s_2}(x) \wedge \widehat{s_4}(x).
\end{array}
$$

One can see that $\mathcal{M}'$ is both a quasi-inverse and a maximum recovery of $\mathcal{M}$. □

To figure out a possible correspondence between the notion of source-to-ontology rewriting and the notions of quasi-inverse and maximum recovery, here we consider OBDM specifications $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ where $\mathcal{O} = \emptyset$.

Due to Proposition 1, one might think that in order to compute a perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ it is sufficient to compute a mapping $\mathcal{M}'$ that is either a quasi-inverse or a maximum recovery of $\mathcal{M}$, and then computing the perfect $\mathcal{O}$-to-$\widehat{S}$ Σ'-rewriting of $\widehat{q_{\mathcal{S}}}$ using the well-known techniques of data integration (cf. Section 2.6). Here, $\widehat{q_{\mathcal{S}}}$ denotes the query obtained from $q_{\mathcal{S}}$ by replacing each predicate name $s \in \mathcal{S}$ with $\widehat{s} \in \widehat{\mathcal{S}}$, and $\Sigma' = \langle \widehat{S}, \mathcal{M}', \mathcal{O} \rangle$ denotes an OBDM specification where $\widehat{S}$ plays the role of an (empty) ontology and $\mathcal{O}$ plays the role of a source schema. We now prove that this is an absolutely wrong belief.

Example 3.13. Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the following OBDM specification:

- $\mathcal{O} = \emptyset$
- $\mathcal{S} = \{\ s_1, s_2, s_3, s_4\ \}$
- $\mathcal{M} = \{\ m_1, m_2, m_3\ \}$ is the mapping defined in Example 3.11.

Let, moreover, $\Sigma' = \langle \widehat{S}, \mathcal{M}', \mathcal{O} \rangle$ be the following OBDM specification:

- $\widehat{\mathcal{S}} = \emptyset$
- $\mathcal{O} = \{\ A_1, A_2\ \}$
- $\mathcal{M}' = \{\ m'_1, m'_2, m'_3\ \}$ is as defined in Example 3.12, i.e., it is both the quasi-inverse and the maximum recovery of $\mathcal{M}$.

Let the query over $\mathcal{S}$ be $q_{\mathcal{S}} = \{(x) \mid s_1(x)\}$. We have that $q_{\mathcal{O}} = \{(x) \mid A_1(x)\}$ is the perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, whereas one can verify that the perfect $\mathcal{O}$-to-$\widehat{S}$ Σ'-rewriting of $\widehat{q_{\mathcal{S}}}$ is the query $q'_{\mathcal{O}} = \{(x) \mid A_1(x)\} \cup \{(x) \mid A_2(x)\}$, which is a complete, but not a sound, $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

Furthermore, one might think that the perfect $\mathcal{O}$-to-$\widehat{S}$ Σ'-rewriting of $\widehat{q_{\mathcal{S}}}$ always produces a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. We now prove that this a wrong belief as well. Let the query over $\mathcal{S}$ be $q'_{\mathcal{S}} = \{(x) \mid s_1(x) \wedge s_2(x)\}$. We have that the query $q''_{\mathcal{O}} = \{(x) \mid A_1(x) \wedge A_2(x)\} \cup \{(x) \mid A_2(x)\}$ is the perfect $\mathcal{O}$-to-$\widehat{S}$ Σ'-rewriting of $\widehat{q'_{\mathcal{S}}}$, but not a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q'_{\mathcal{S}}$. In this case, one can verify that the unique (up to equivalence w.r.t. Σ) UCQ-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q'_{\mathcal{S}}$ is the query $q_{\mathcal{O}}$ defined above. $\square$

We conclude with a very elementary but useful observation. Notice that, in the above example, the quasi-inverse and maximum recovery mapping $\mathcal{M}'$ of $\mathcal{M}$ is obtained by simply *reversing the arrows* of $\mathcal{M}$. Therefore, given an OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, the OBDM specification $rev(\Sigma) = \langle \widehat{S}, rev(\mathcal{M}), \mathcal{O} \rangle$ where $rev(\mathcal{M})$ is the mapping from $\widehat{S}$ to $\mathcal{O}$ obtained by simply reversing the arrows of $\mathcal{M}$ does not help in any way for finding source-to-ontology rewritings.

Chapter 4

Dealing with Inequalities in Lightweight Description Logics

As explained in the introduction, we will see that by allowing inequalities in a certain target query language considered in this thesis provide a better means to compute abstractions of data services compared to the same language without inequalities.

This chapter deals with the problem of adding inequalities to UCQs in the OBDM scenario. Notably, although UCQs constitute the most popular class of queries studied for both databases and ontologies, they have several limitations in expressive power. For instance, suppose we want to retrieve all the triangles in an undirected graph possibly with loops.[1] This property can not be expressed by means of a UCQ, whereas it can be expressed as the following CQ$^{\neq,b}$ $q = \{(x, y, z) \mid edge(x, y) \wedge edge(x, z) \wedge edge(y, z) \wedge x \neq y \wedge x \neq z \wedge y \neq z\}$, where the predicate *edge* represents the connections between vertices in the graph.

It is worth noting that in OBDM, while answering (U)CQs has been extensively studied, e.g., by establishing bound on the size of rewritings [Gottlob *et al.*, 2014], developing optimisation algorithms [Rosati and Almatelli, 2010], implementing engines for real-world applications [Calvanese *et al.*, 2011; Civili *et al.*, 2013; Calvanese *et al.*, 2017], etc., the problem of answering (U)CQ$^{\neq}$s has been rarely investigated. We start by analysing the case of knowledge bases.

To the best of our knowledge, the basic known facts about such problem can be summarised as follows, which hold regardless of whether the UNA is adopted or not:

- In stark contrast to the UCQ case, answering UCQ$^{\neq}$s even over $\textit{DL-Lite}_{core}$ knowledge bases is undecidable [Gutiérrez-Basulto *et al.*, 2012]. For $\textit{DL-Lite}_{\mathcal{R}}$ knowledge bases, undecidability holds already for CQ$^{\neq}$s [Gutiérrez-Basulto *et al.*, 2015]. Looking at these results, one can easily realise that the main of source of undecidability stems from both the ability of the ontology language to express incomplete information through existential quantifiers, and the possibility of imposing inequalities between existential variables in the query;

- In [Gutiérrez-Basulto *et al.*, 2015], it is also proved that for the subclasses of CQ$^{\neq}$s and UCQ$^{\neq}$s named *local CQ$^{\neq}$s* and *local UCQ$^{\neq}$s*, respectively, query answering over $\textit{DL-Lite}_{\mathcal{R}}$ knowledge bases is decidable, but with a coNExpTime

[1] In graph theory, a loop is an edge that connects a vertex with itself [Bondy and Murty, 2008].

upper bound in data complexity. Furthermore, it is provably intractable (coNP-hard in data complexity) already for local $\mathrm{CQ}^{\neq}$s. Local (U)CQs are special (U)CQs with inequalities, designed in such a way that each inequality atom in the query that contributes to a certain answer with respect to a DL knowledge base has at least one of its terms bound by an individual in the ABox.

The goal of this chapter is then to investigate under which conditions, stronger than local UCQs, tractability of answering queries with inequalities is recovered, or at least the complexity is lowered with respect to the one of local UCQs. The basic idea to achieve this goal is to explore ontology languages and query languages ensuring the following property: each inequality atom $\alpha \neq \beta$ that contributes to the certain answer to a query with respect to a DL knowledge base, it does so with both terms α and β bounded by individuals in the ABox. In order to follow this path, we explore two alternative strategies:

1. The first strategy is to consider $\mathrm{UCQ}^{\neq,b}$s, which restricts the application of the inequality atoms to either individuals or distinguished variables, i.e., we consider the problem of answering $(\mathrm{U})\mathrm{CQ}^{\neq,b}$s over *DL-Lite*$_\mathcal{R}$ knowledge bases. Notice that, although limited, the expressive power of $(\mathrm{U})\mathrm{CQ}^{\neq,b}$s allow for interesting queries to be expressed such as the one computing the triangles in an undirected graph with loops.

2. The second strategy is to consider *DL-Lite*$^{\neg}_{\mathsf{RDFS}}$ knowledge bases, which eliminates all the constructs introducing incomplete information resulting from existentially quantified assertions, i.e., we consider the problem of answering $\mathrm{UCQ}^{\neq}$s over *DL-Lite*$^{\neg}_{\mathsf{RDFS}}$ knowledge bases.

As a first consideration, observe that the above problems are straightforward generalisations of the UCQ case when the UNA is adopted. To see this, it is sufficient to consider the slight extension of the algorithm $\mathsf{PerfectRef}$ accepting $\mathrm{UCQ}^{\neq}$s as input queries rather than UCQs, where the predicate $\neq$ is treated in the reformulation algorithm as an additional atomic role of the input *DL-Lite*$_\mathcal{R}$ ontology. From now on, we implicitly refer to $\mathsf{PerfectRef}$ as the algorithm described in Subsection 2.5.3 with the above modification. With this discussion in mind, it is straightforward to verify that the computational complexity of the above defined problems remains the same as that of the problem of answering UCQs over *DL-Lite*$_\mathcal{R}$ knowledge bases.

Theorem 4.1. *Both the problem of answering $\mathrm{UCQ}^{\neq,b}$s over DL-Lite$_\mathcal{R}$ knowledge bases and the problem of answering $\mathrm{UCQ}^{\neq}$s over DL-Lite$^{\neg}_{\mathsf{RDFS}}$ knowledge bases are FOL-rewritable (and therefore in AC^0 in data complexity), and NP-complete in combined complexity.*

Proof. Let q be a $\mathrm{UCQ}^{\neq,b}$ (respectively, $\mathrm{UCQ}^{\neq}$) over a *DL-Lite*$_\mathcal{R}$ (respectively, *DL-Lite*$^{\neg}_{\mathsf{RDFS}}$) knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$. Analogously to [Calvanese *et al.*, 2007b, Theorem 29], since we are adopting the UNA, it is easy to see that $\vec{c} \in cert^{\mathcal{A}}_{q,\mathcal{O}}$ if and only if there is a disjunct $q' = \{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$ of q for which there is a homomorphism h from q' to $\mathcal{C}^{\mathcal{A}}_{\mathcal{O}}$ with $h(\vec{t}) = \vec{c}$. Note that (i) $h(\alpha) \neq h(\beta)$ for each inequality atom $\alpha \neq \beta$ occurring in q'; (ii) both $h(\alpha) \neq h(\beta)$ are individuals.

This is because, in the case that $\mathcal{O}$ is a *DL-Lite*$_\mathcal{R}$ ontology, $h(\alpha)$ and $h(\beta)$ are necessarily individuals occurring in $\mathcal{A}$ since q is a UCQ$^{\neq,b}$. In the case that $\mathcal{O}$ is a *DL-Lite*$_{\mathsf{RDFS}}$ ontology, again, both $h(\alpha)$ and $h(\beta)$ are necessarily individuals occurring in $\mathcal{A}$ since $\mathcal{C}_\mathcal{O}^\mathcal{A}$ does not introduce any variable in Var. Thus, with the same arguments of [Calvanese *et al.*, 2007b, Lemma 39], we can easily conclude that $cert_{q,\mathcal{O}}^{\mathcal{A}} = \mathsf{PerfectRef}(\mathcal{O}, q)^{\mathcal{I}_\mathcal{A}}$, from which the claim trivially follows. □

However, while answering UCQs over *DL-Lite*$_\mathcal{R}$ knowledge bases is insensitive to the adoption of the UNA [Artale *et al.*, 2009](cf. Subsection 2.5.3), the following example shows that this is not anymore the case even when answering CQ$^{\neq,b}$s over knowledge bases with empty TBoxes.

Example 4.1. Consider the knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$, where $\mathcal{O} = \emptyset$ and $\mathcal{A} = \{P(c_1, c_2)\}$. For the CQ$^{\neq,b}$ $q = \{(x_1, x_2) \mid P(x_1, x_2) \wedge x_1 \neq x_2\}$, it is easy to see that the tuple (c_1, c_2) is in the certain answers of q with respect to $\mathcal{O}$ and $\mathcal{A}$ when the UNA is adopted, while it is not if the UNA is not adopted. For this latter, consider the model $\mathcal{I} = \langle \Delta^\mathcal{I}, \cdot^\mathcal{I} \rangle$ of $\mathcal{K}$ with $\Delta^\mathcal{I} = c$, $c_1^\mathcal{I} = c_2^\mathcal{I} = c$, and $P^\mathcal{I} = \{(c, c)\}$, which does not satisfy the UNA. We have $q^\mathcal{I} = \emptyset$. □

All the rest of this chapter is devoted to analyse the computational complexity of the above problems when the UNA is not adopted, as in the `OWL 2` web ontology language profiles. We also make a connection on how these new results contribute, on the one hand, to new results on containment of UCQ$^{\neq}$s in the relational databases theory, and, on the other hand, to new results on the *Direct Semantics Entailment Regime (DSER)* [Glimm, 2011]. The complexity results are summarised in Figure 4.1.

In principle, since the UNA is not adopted, it may be reasonable to include in knowledge bases $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ also ABox assertions sanctioning the fact that two individuals have to be interpreted as different elements in all possible of models of $\mathcal{K}$, as can be effectively done in `OWL 2 QL` by means of the `DifferentIndividuals` assertions. Notice, however, that the expressive capabilities of the *DL-Lite*$_\mathcal{R}$ and *DL-Lite*$_{\mathsf{RDFS}}^{\neg}$ knowledge base languages would remain the same, since in fact each ABox assertion of the form $\alpha = \mathtt{DifferentIndividuals}(c_1, c_2)$ is equivalent to the ABox assertions $\mathtt{DIFF}_1(c_1, \alpha)$, $\mathtt{DIFF}_2(c_2, \alpha)$, where α is a fresh individual and $\mathtt{DIFF}_1$, $\mathtt{DIFF}_2$ are two atomic roles such that $\mathtt{DIFF}_1 \sqsubseteq \neg\mathtt{DIFF}_2$ is a disjointness assertion of every ontology $\mathcal{O}$. An analogous but different argument (since the ontology size of the equivalent knowledge base is linear on the size of the original ABox) holds also for *DL-Lite*$_{core}$ knowledge bases. Indeed, each ABox assertion of the form $\alpha = \mathtt{DifferentIndividuals}(c_1, c_2)$ is equivalent to the set of assertions $A_{\text{new}}^{\alpha_1}(c_1)$, $A_{\text{new}}^{\alpha_2}(c_2)$, and $A_{\text{new}}^{\alpha_1} \sqsubseteq \neg A_{\text{new}}^{\alpha_2}$, where $A_{\text{new}}^{\alpha_1}$ and $A_{\text{new}}^{\alpha_2}$ are fresh atomic concepts.

In what follows on this chapter, unless otherwise stated, we implicitly assume to deal with satisfiable knowledge bases.[2] Moreover, unless otherwise stated and without loss of generality, we consider only boolean UCQ$^{\neq}$s. Indeed, given an n-ary UCQ$^{\neq}$ $q_\mathcal{O}$, a *DL-Lite*$_\mathcal{R}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$, and an n-tuple $\vec{c}$ of constants, checking whether $\vec{c} \in cert_{q_\mathcal{O},\mathcal{O}}^{\mathcal{A}}$ is equivalent to checking whether $\mathcal{K} \models q_\mathcal{O}(\vec{c})$.

[2]As discussed in Subsection 2.5.3, for unsatisfiable knowledge bases query answering is trivial. Furthermore, the problem of checking whether a *DL-Lite*$_\mathcal{R}$ knowledge base is satisfiable is FOL-rewritable (and therefore in AC^0 in data complexity) and in PTime in the size of the ontology.

***DL-Lite*$_{\mathcal{R}}$:**

	Data	Combined
$\text{CQ}^{\neq,b}$s	in AC^0	NP-c
$\text{UCQ}^{\neq,b}$s	in AC^0	Π_2^p-c

***DL-Lite*$_{\mathsf{RDFS}}^{\neg}$:**

	Data	Combined
$\text{UCQ}^{1,\neq}$s	PTIME-c	NP-c
$\text{UCQ}^{\neq}$s	coNP-c	Π_2^p-c

Figure 4.1. Summary of data and combined complexity results of our considered problems when the UNA is not adopted ("-c" abbreviates "-complete"). The table on the left (respectively, right) consider *DL-Lite*$_{\mathcal{R}}$ (respectively, *DL-Lite*$_{\mathsf{RDFS}}^{\neg}$) knowledge bases.

4.1 The Chase for Answering Queries with Inequalities

The conceptual tool we use for addressing the problem of answering queries with inequalities over *DL-Lite*$_{\mathcal{R}}$ knowledge base is an extension of the chase illustrated in [Calvanese *et al.*, 2007b]. Specifically, given a *DL-Lite*$_{\mathcal{R}}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$, we build a (possibly infinite) set of atoms, starting from $Ch^0(\mathcal{K}) := \mathcal{A}$, and repeatedly computing $Ch^{j+1}(\mathcal{K})$ from $Ch^j(\mathcal{K})$ by applying suitable rules, where each rule can be applied only if certain conditions hold. In doing so, we make use of the alphabet Var of variables, and follow a deterministic strategy that is fair, i.e., if at some point a rule is applicable, then it will be eventually applied. Finally, we set $\mathcal{I}_{\mathcal{K}} := \bigcup_{i \geq 0} Ch^i(\mathcal{K})$. Notice that we make use of the additional binary predicate symbol *ineq*, which is used to record all inequalities logically implied by $\mathcal{K}$.

The rules we use include all the ones illustrated in [Calvanese *et al.*, 2007b, Definition 5]. For instance, if $A_1 \sqsubseteq \exists P \in \mathcal{O}$, $A_1(e_1) \in Ch^j(\mathcal{K})$, and no e_2 exists such that $P(e_1, e_2) \in Ch^j(\mathcal{K})$, then we set $Ch^{j+1}(\mathcal{K}) := Ch^j(\mathcal{K}) \cup \{P(e_1, e')\}$, where $e' \in \mathsf{Var}$ does not appear in $Ch^j(\mathcal{K})$. There are, however, two crucial additions related to the *ineq* predicate. In what follows, when we say $R(e_1, e_2)$ holds in $Ch^j(\mathcal{K})$, where R is a basic role, we mean (*i*) $P(e_1, e_2) \in Ch^j(\mathcal{K})$, if $R = P$, or (*ii*) $P(e_2, e_1) \in Ch^j(\mathcal{K})$, if $R = P^-$. Also, when we say that $B(e_1)$ holds in $Ch^j(\mathcal{K})$, where B is a basic concept, we mean (*i*) $A(e_1) \in Ch^j(\mathcal{K})$ if $B = A$, and (*ii*) $R(e_1, e_2)$ holds in $Ch^j(\mathcal{K})$ for some e_2, if $B = \exists R$. The additional two rules are as follows:

- if $B_1 \sqsubseteq \neg B_2 \in \mathcal{O}$, $B_1(e_1)$ and $B_2(e_2)$ hold in $Ch^j(\mathcal{K})$, and $ineq(e_1, e_2) \notin Ch^j(\mathcal{K})$, then $Ch^{j+1}(\mathcal{K}) := Ch^j(\mathcal{K}) \cup \{ineq(e_1, e_2), ineq(e_2, e_1)\}$;

- if $R_1 \sqsubseteq \neg R_2 \in \mathcal{O}$, $ineq(e_1, e_2) \notin Ch^j(\mathcal{K})$, and either $R_1(e_3, e_1)$ and $R_2(e_3, e_2)$ hold in $Ch^j(\mathcal{K})$ or $R_1(e_1, e_3)$ and $R_2(e_2, e_3)$ hold in $Ch^j(\mathcal{K})$, then $Ch^{j+1}(\mathcal{K}) := Ch^j(\mathcal{K}) \cup \{ineq(e_1, e_2), ineq(e_2, e_1)\}$.

In other words, for a *DL-Lite*$_{\mathcal{R}}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$, the set of atoms $\mathcal{I}_{\mathcal{K}}$ contains all atoms contained in $\mathcal{C}_{\mathcal{O}}^{\mathcal{A}}$ plus some possible atoms with *ineq* as predicate, used to record all and only the inequalities logically implied by $\mathcal{K}$.

Obviously, for a *DL-Lite*$_{\mathcal{R}}$ knowledge base $\mathcal{K}$, the set $\mathcal{I}_{\mathcal{K}}$ can be infinite, due to the presence of existential quantifiers in the right-hand side of inclusion assertions, which, by introducing fresh unknown variables, can trigger an infinite number of rule applications. It is easy to see that, on the contrary, for a *DL-Lite*$_{\mathsf{RDFS}}^{\neg}$ knowledge base $\mathcal{K}$, $\mathcal{I}_{\mathcal{K}}$ is finite, and can be computed in polynomial time in the size of $\mathcal{K}$.

We next show that $\mathcal{I}_{\mathcal{K}}$ enjoys some crucial properties.

Proposition 4.1. *If $\mathcal{M} = \langle \Delta^{\mathcal{M}}, \cdot^{\mathcal{M}} \rangle$ is a model of a DL-Lite$_{\mathcal{R}}$ knowledge base $\mathcal{K}$, then there exists a function Ψ from $\mathsf{dom}(\mathcal{I}_{\mathcal{K}})$ to $\Delta^{\mathcal{M}}$ such that:*

1. *for every $e \in \mathsf{dom}(\mathcal{I}_{\mathcal{K}})$, if $A(e) \in \mathcal{I}_{\mathcal{K}}$, then $\Psi(e) \in A^{\mathcal{M}}$;*

2. *for every pair $e_1, e_2 \in \mathsf{dom}(\mathcal{I}_{\mathcal{K}})$, if $P(e_1, e_2) \in \mathcal{I}_{\mathcal{K}}$, then $(\Psi(e_1), \Psi(e_2)) \in P^{\mathcal{M}}$;*

3. *for every pair $e_1, e_2 \in \mathsf{dom}(\mathcal{I}_{\mathcal{K}})$, if $ineq(e_1, e_2) \in \mathcal{I}_{\mathcal{K}}$, then $\Psi(e_1) \neq \Psi(e_2)$.*

Proof. Consider any model $\mathcal{M}$ of $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$. By [Calvanese *et al.*, 2007b, Lemma 28], there exists a function Ψ from $\mathsf{dom}(\mathcal{I}_{\mathcal{K}})$ to $\Delta^{\mathcal{M}}$ with $\Psi(c) = c^{\mathcal{M}}$ for each individual c occurring in $\mathcal{A}$ such that both conditions 1 and 2 hold.

As for condition 3, from the rules applied for building $\mathcal{I}_{\mathcal{K}}$, if two elements $e_1, e_2 \in \mathsf{dom}(\mathcal{I}_{\mathcal{K}})$ are such that $ineq(e_1, e_2) \in \mathcal{I}_{\mathcal{K}}$, then by definition one of the following two conditions holds: (i) $B_1 \sqsubseteq \neg B_2 \in \mathcal{O}$ and $B_1(e_1), B_2(e_2)$ hold in $\mathcal{I}_{\mathcal{K}}$ (thus, $B_1(e_1) \in \mathcal{I}_{\mathcal{K}}$ and $B_2(e_2) \in \mathcal{I}_{\mathcal{K}}$), (ii) $R_1 \sqsubseteq \neg R_2 \in \mathcal{O}$ and either $R_1(e_3, e_1)$ and $R_2(e_3, e_2)$ hold in $\mathcal{I}_{\mathcal{K}}$ (thus, $R_1(e_3, e_1) \in \mathcal{I}_{\mathcal{K}}$ and $R_2(e_3, e_2) \in \mathcal{I}_{\mathcal{K}}$), or $R_1(e_1, e_3)$ and $R_2(e_2, e_3)$ hold in $\mathcal{I}_{\mathcal{K}}$ (thus, $R_1(e_1, e_3) \in \mathcal{I}_{\mathcal{K}}$ and $R_2(e_2, e_3) \in \mathcal{I}_{\mathcal{K}}$).

Suppose by contradiction that Ψ does not satisfy condition 3, i.e., there are two elements $e_1, e_2 \in \mathsf{dom}(\mathcal{I}_{\mathcal{K}})$ such that $ineq(e_1, e_2) \in \mathcal{I}_{\mathcal{K}}$ and $\Psi(e_1) = \Psi(e_2)$.

If $ineq(e_1, e_2) \in \mathcal{I}_{\mathcal{K}}$ because of (i), then, by condition 1, we derive $\Psi(e_1) \in B_1^{\mathcal{M}}$ and $\Psi(e_2) \in B_2^{\mathcal{M}}$, which implies that $\mathcal{M}$ does not satisfy the disjointness assertion $B_1 \sqsubseteq \neg B_2$ of $\mathcal{O}$, thus contradicting the fact that $\mathcal{M}$ is a model of $\mathcal{K}$, as required.

If $ineq(e_1, e_2) \in \mathcal{I}_{\mathcal{K}}$ because of (ii), then, by condition 2, we derive that either $(\Psi(e_3), \Psi(e_1)) \in R_1^{\mathcal{M}}$ and $(\Psi(e_3), \Psi(e_2)) \in R_2^{\mathcal{M}}$, or $(\Psi(e_1), \Psi(e_3)) \in R_1^{\mathcal{M}}$ and $(\Psi(e_2), \Psi(e_3)) \in R_2^{\mathcal{M}}$. Notice, however, that both cases imply that $\mathcal{M}$ does not satisfy the disjointness assertion $R_1 \sqsubseteq \neg R_2$ of $\mathcal{O}$, thus contradicting the fact that $\mathcal{M}$ is a model of $\mathcal{K}$, as required.

Therefore, we conclude that Ψ must be such that for every pair $e_1, e_2 \in \mathsf{dom}(\mathcal{I}_{\mathcal{K}})$, if $ineq(e_1, e_2) \in \mathcal{I}_{\mathcal{K}}$, then $\Psi(e_1) \neq \Psi(e_2)$. □

A reasonable question to ask is whether $\mathcal{I}_{\mathcal{K}}$ is the right tool for query answering. The next theorem provides a positive answer to this question for CQ$^{\neq,b}$s. In what follows, given a CQ$^{\neq,b}$ q over an ontology $\mathcal{O}$, $\delta(q)$ denotes the query obtained by replacing each inequality atom $x_1 \neq x_2$ in q with the atom $ineq(x_1, x_2)$.

Theorem 4.2. *Let $\vec{c}$ be a tuple of individuals occurring in a satisfiable DL-Lite$_{\mathcal{R}}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$, and let q be a CQ$^{\neq,b}$ over $\mathcal{O}$. We have that $\vec{c} \in cert_{q,\mathcal{O}}^{\mathcal{A}}$ if and only if $\vec{c} \in \delta(q)^{\mathcal{I}_{\mathcal{K}}}$.*

Proof. Consider the boolean query $q(\vec{c})$. By definition, we have $\vec{c} \in cert_{q,\mathcal{O}}^{\mathcal{A}}$ if and only if $\mathcal{K} \models q(\vec{c})$, and $\vec{c} \in \delta(q)^{\mathcal{I}_{\mathcal{K}}}$ if and only if $\mathcal{I}_{\mathcal{K}} \models \delta(q(\vec{c}))$. We now prove that $\mathcal{K} \models q(\vec{c})$ if and only if $\mathcal{I}_{\mathcal{K}} \models \delta(q(\vec{c}))$, thus showing the claim.

"**If part:**" If $\mathcal{I}_{\mathcal{K}} \models \delta(q(\vec{c}))$, then, using Proposition 4.1, we easily derive that $\mathcal{M} \models q(\vec{c})$ for each model $\mathcal{M}$ of $\mathcal{K}$. Thus, $\mathcal{K} \models q(\vec{c})$, as required.

"**Only-if part:**" Suppose that $\mathcal{I}_{\mathcal{K}} \not\models \delta(q(\vec{c}))$. If in $\delta(q(\vec{c}))$ there is at least an atom, which is not an *ineq* atom, that is not satisfied by $\mathcal{I}_{\mathcal{K}}$, then, from $\mathcal{I}_{\mathcal{K}}$ itself, we easily obtain a model $\mathcal{M}_{\mathcal{K}}$ of $\mathcal{K}$ such that $\mathcal{M}_{\mathcal{K}} \not\models q(\vec{c})$, and therefore $\mathcal{K} \not\models q(\vec{c})$.

On the other hand, if $\mathcal{I}_\mathcal{K}$ satisfies all atoms of $\delta(q(\vec{c}))$ different from *ineq* atoms, then there is at least one atom of the form $ineq(c_1, c_2)$ in $\delta(q(\vec{c}))$ that is false in $\mathcal{I}_\mathcal{K}$. Observe that, since $q(\vec{c})$ is a boolean $\mathrm{CQ}^{\neq,b}$, both c_1 and c_2 must be individuals.

Consider the interpretation $\mathcal{M}' = \langle \Delta^\mathcal{M}, \cdot^\mathcal{M} \rangle$ for $\mathcal{K}$ obtained from $\mathcal{I}_\mathcal{K}$ as follows: (i) $\Delta^\mathcal{M} = \mathsf{Const}$; (ii) $c^{\mathcal{M}'} = c$, for each individual c occurring in $\mathcal{A}$ such that $c \neq c_1$ and $c \neq c_2$; (iii) $c_1^{\mathcal{M}'} = c_2^{\mathcal{M}'} = c'$, where $c' \in \mathsf{Const}$ is a fresh constant not occurring in $\mathcal{A}$; and (iv) the extensions of atomic concepts and atomic roles properly follows atoms of $\mathcal{I}_\mathcal{K}$, but where each variable $v \in \mathsf{Var}$ occurring in $\mathcal{I}_\mathcal{K}$ is replaced everywhere with a fresh constant $c_v \in \mathsf{Const}$. Since $ineq(c_1, c_2) \notin \mathcal{I}_\mathcal{K}$, we have that $\mathcal{M}'$ is a model of $\mathcal{K}$. In proof, if $\mathcal{M}'$ does not satisfy a disjointness assertion, then, by simple induction on the steps applied for building $\mathcal{I}_\mathcal{K}$, we would easily derive a contradiction on the fact that $ineq(c_1, c_2) \notin \mathcal{I}_\mathcal{K}$.

But then, $\mathcal{M}'$ is a model of $\mathcal{K}$ such that $c_1^{\mathcal{M}'} = c_2^{\mathcal{M}'}$. Since, however, $c_1 \neq c_2$ is an inequality atom occurring in the body of $q(\vec{c})$, we derive $\mathcal{M}' \not\models q(\vec{c})$. Therefore, $\mathcal{M}'$ is a model of $\mathcal{K}$ for which $\mathcal{M}' \not\models q(\vec{c})$. Thus, $\mathcal{K} \not\models q(\vec{c})$, as required. □

4.2 Answering $\mathrm{CQ}^{\neq,b}$s over *DL-Lite*$_\mathcal{R}$ knowledge bases

In this section, we first consider the problem of answering conjunctive queries with bounded inequalities ($\mathrm{CQ}^{\neq,b}$s) over satisfiable *DL-Lite*$_\mathcal{R}$ knowledge bases, and then we discuss how these results carry over on the one hand to the OBDM scenario, and, on the other hand, to the case of the DSER, which is the semantics usually adopted in the Semantic Web scenarios that slightly differs from the classical FOL semantics.

We start by introducing some notations. Given an inequality atom $x_1 \neq x_2$ and a disjointness assertion γ, we denote by $\rho(x_1 \neq x_2, \gamma)$ the following open formula:

- $\rho(x_1 \neq x_2, A_1 \sqsubseteq \neg A_2) = A_1(x_1) \land A_2(x_2)$,
- $\rho(x_1 \neq x_2, A \sqsubseteq \neg \exists R) = \rho(x_1 \neq x_2, \exists R \sqsubseteq \neg A) = \exists y . A(x_1) \land R(x_2, y)$, where y is a fresh existential variable,
- $\rho(x_1 \neq x_2, \exists R_1 \sqsubseteq \neg \exists R_2) = \exists y_1, y_2 . R_1(x_1, y_1) \land R_2(x_2, y_2)$, where y_1 and y_2 are fresh different existential variables, and
- $\rho(x_1 \neq x_2, R_1 \sqsubseteq \neg R_2) = \exists y . R_1(x_1, y) \land R_2(x_2, y) \lor \exists y . R_1(y, x_1) \land R_2(y, x_2)$, where y is a fresh existential variable,

where an atom of the form $R(x, y)$ stands for either $P(x, y)$ if R denotes an atomic role P, or $P(y, x)$ if R denotes the inverse of an atomic role, i.e., $R = P^-$.

Example 4.2. Consider the atom $x \neq c$ and the *DL-Lite*$_\mathcal{R}$ disjointness assertion $P_1 \sqsubseteq \neg P_2$. Then, we have that $\rho(x \neq c, P_1 \sqsubseteq \neg P_2) = \exists y . P_1(x, y) \land P_2(c, y) \lor \exists y . P_1(y, x) \land P_2(y, c)$. □

Given an inequality atom $x_1 \neq x_2$ and a *DL-Lite*$_\mathcal{R}$ ontology $\mathcal{O}$, we denote by $\sigma(x_1 \neq x_2, \mathcal{O})$ the following disjunction of atoms:

$$\bigvee_{i=1}^{m} (\rho(x_1 \neq x_2, \gamma_i) \lor \rho(x_2 \neq x_1, \gamma_i)),$$

where $\gamma_1, \ldots, \gamma_m$ are all the disjointness assertions of the *DL-Lite*$_\mathcal{R}$ ontology $\mathcal{O}$.

Example 4.3. Consider the atom $x \neq c$ and the *DL-Lite*$_{\mathcal{R}}$ ontology $\mathcal{O} = \{A \sqsubseteq \exists P, P_1 \sqsubseteq \neg P_2, P_3 \sqsubseteq P_2\}$. We have that $\sigma(x \neq c, \mathcal{O}) = \exists y.P_1(x,y) \wedge P_2(c,y) \vee \exists y.P_1(y,x) \wedge P_2(y,c) \vee \exists y.P_1(c,y) \wedge P_2(x,y) \vee \exists y.P_1(y,c) \wedge P_2(y,x)$. □

Finally, for a CQ$^{\neq}$ q over a *DL-Lite*$_{\mathcal{R}}$ ontology $\mathcal{O}$, we denote by $\lambda(q, \mathcal{O})$ the query obtained from q by replacing every inequality atom $x_1 \neq x_2$ with $\sigma(x_1 \neq x_2, \mathcal{O})$, and then turning the resulting formula into an equivalent UCQ.

Example 4.4. Consider the CQ$^{\neq}$ $q = \{(x) \mid \exists y_1.P(x,y_1) \wedge x \neq c\}$ over the ontology $\mathcal{O}$ of Example 4.3. We have that $\lambda(q, \mathcal{O}) = \{q_1, q_2, q_3, q_4\}$, where

- $q_1 = \{(x) \mid \exists y_1, y_2.P(x,y_1) \wedge P_1(x,y_2) \wedge P_2(c,y_2)\}$.
- $q_2 = \{(x) \mid \exists y_1, y_2.P(x,y_1) \wedge P_1(y_2,x) \wedge P_2(y_2,c)\}$.
- $q_3 = \{(x) \mid \exists y_1, y_2.P(x,y_1) \wedge P_1(c,y_2) \wedge P_2(x,y_2)\}$.
- $q_4 = \{(x) \mid \exists y_1, y_2.P(x,y_1) \wedge P_1(y_2,c) \wedge P_2(y_2,x)\}$. □

The next important proposition, whose proof relies on [Calvanese *et al.*, 2007b, Theorem 29] and on Theorem 4.2, states that computing $cert^{\mathcal{A}}_{q,\mathcal{O}}$, for a given *DL-Lite*$_{\mathcal{R}}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ and CQ$^{\neq,b}$ q over $\mathcal{O}$, can be reduced to computing the certain answers of the UCQ $\lambda(q, \mathcal{O})$ with respect to $\mathcal{O}$ and $\mathcal{A}$.

Proposition 4.2. *Let $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ be a DL-Lite$_{\mathcal{R}}$ knowledge base, and let q be a CQ$^{\neq,b}$ over $\mathcal{O}$. Then, we have that $cert^{\mathcal{A}}_{q,\mathcal{O}} = cert^{\mathcal{A}}_{\lambda(q,\mathcal{O}),\mathcal{O}}$.*

Proof. Let $\vec{c}$ be any tuple of individuals occurring in $\mathcal{A}$. By definition, we have $\vec{c} \in cert^{\mathcal{A}}_{q,\mathcal{O}}$ if and only if $\mathcal{K} \models q(\vec{c})$. Analogously, $\vec{c} \in cert^{\mathcal{A}}_{\lambda(q,\mathcal{O}),\mathcal{O}}$ if and only if $\mathcal{K} \models \lambda(q, \mathcal{O})(\vec{c})$. We now prove that $\mathcal{K} \models q(\vec{c})$ if and only if $\mathcal{K} \models \lambda(q, \mathcal{O})(\vec{c})$.

"**If part:**" If $\mathcal{K} \models \lambda(q, \mathcal{O})(\vec{c})$, then, due to [Calvanese *et al.*, 2007b, Theorem 29], we derive that there is a boolean CQ q' in the boolean UCQ $\lambda(q, \mathcal{O})(\vec{c})$ for which $\mathcal{I}_{\mathcal{K}} \models q'$. Observe that, by construction, each boolean CQ q' in the boolean UCQ $\lambda(q, \mathcal{O})(\vec{c})$ is such that $q' = \{() \mid \alpha_1 \wedge \ldots \wedge \alpha_n \wedge \beta_1 \wedge \beta'_1 \wedge \ldots \wedge \beta_m \wedge \beta'_m\}$, where α_is are atoms that were not inequality atoms in the boolean CQ $q(\vec{c})$, whereas the pair of atoms (β_i, β'_i)s are obtained after replacing an inequality atom of the form $c_1 \neq c_2$ with $\sigma(c_1 \neq c_2, \mathcal{O})$. Since $q(\vec{c})$ is a boolean CQ, both c_1 and c_2 must be individuals.

Notice, however, that since $\mathcal{I}_{\mathcal{K}} \models \beta_i \wedge \beta'_i$, by construction of $\sigma(c_1 \neq c_2, \mathcal{O})$ and the additional rules applied for building $\mathcal{I}_{\mathcal{K}}$ (the ones regarding the *ineq* predicate), we easily derive that $ineq(c_1, c_2) \in \mathcal{I}_{\mathcal{K}}$. Moreover, since this is true for each pair of atoms (β_i, β'_i) for $i \in [1, m]$, we further derive that $\delta(q(\vec{c}))$ is such that $\mathcal{I}_{\mathcal{K}} \models \delta(q(\vec{c}))$, thus implying that $\vec{c} \in \delta(q)^{\mathcal{I}_{\mathcal{K}}}$. By virtue of Theorem 4.2, we derive that $\vec{c} \in cert^{\mathcal{A}}_{q,\mathcal{O}}$, and therefore $\mathcal{K} \models q(\vec{c})$, as required.

"**Only-if part:**" If $\mathcal{K} \models q(\vec{c})$, then, by virtue of Theorem 4.2, we derive that $\mathcal{I}_{\mathcal{K}} \models \delta(q(\vec{c}))$. Consider each atom of the form $ineq(c_1, c_2)$ occurring in $\delta(q(\vec{c}))$. Observe that, since $\delta(q(\vec{c}))$ is a boolean CQ, both c_1 and c_2 must be individuals. From the premises of the additional rules for building $\mathcal{I}_{\mathcal{K}}$ (the ones regarding the *ineq* predicate), we easily derive that the formula $\sigma(c_1 \neq c_2, \mathcal{O})$ is true when evaluated over $\mathcal{I}_{\mathcal{K}}$, i.e., $\mathcal{I}_{\mathcal{K}} \models \sigma(c_1 \neq c_2, \mathcal{O})$. Therefore, it is easy to see that $\mathcal{I}_{\mathcal{K}} \models q'$ for at least a boolean CQ q' in $\lambda(q, \mathcal{O})(\vec{c})$. Thus, due to [Calvanese *et al.*, 2007b, Theorem 29], we derive $\mathcal{K} \models \lambda(q, \mathcal{O})(\vec{c})$, as required. □

Let $\mathsf{NoUNAPerfectRef}$ denotes the algorithm that, given in input a *DL-Lite*$_{\mathcal{R}}$ ontology $\mathcal{O}$ and a CQ$^{\neq,b}$ q over $\mathcal{O}$, it returns the UCQ $\mathsf{PerfectRef}(\mathcal{O}, \lambda(q, \mathcal{O}))$. By combining the above proposition with [Calvanese *et al.*, 2007b, Lemma 39], we immediately derive the following result.

Corollary 4.1. *Let $\mathcal{O}$ be a DL-Lite$_{\mathcal{R}}$ ontology, and let q be a CQ$^{\neq,b}$ over $\mathcal{O}$. For every ABox $\mathcal{A}$ such that the DL-Lite$_{\mathcal{R}}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ is satisfiable, we have $cert_{q,\mathcal{O}}^{\mathcal{A}} = \mathsf{NoUNAPerfectRef}(\mathcal{O}, q)^{\mathcal{I}_{\mathcal{A}}}$.*

Example 4.5. Consider the *DL-Lite*$_{\mathcal{R}}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$, where $\mathcal{O}$ is the *DL-Lite*$_{\mathcal{R}}$ ontology described in Example 4.3 and $\mathcal{A} = \{A(c_1), P_1(c_2, c), P_3(c_2, c)\}$. Let, moreover, q be the CQ$^{\neq}$ described in Example 4.4. $\mathsf{NoUNAPerfectRef}(\mathcal{O}, q)$ contains, among others, the following CQ:

$$q_5 = \{(x) \mid \exists y_2 . A(x) \land P_1(y_2, c) \land P_3(y_2, x)\},$$

which is obtained from the CQ q_4 of $\lambda(q, \mathcal{O})$ described in Example 4.4 by first applying τ to q_4, thus obtaining $\{(x) \mid \exists y_2 . P(x, \_) \land P_1(y_2, c) \land P_2(y_2, x)\}$, and then by applying the inclusion assertions $A \sqsubseteq \exists P$ and $P_3 \sqsubseteq P_2$ to atoms $P(x, \_)$ and $P_2(y_2, x)$, respectively. Obviously, $q_5^{\mathcal{I}_{\mathcal{A}}} = \{(c_1)\}$. One can easily verify that $(c_1) \in cert_{q,\mathcal{O}}^{\mathcal{A}}$ (more specifically, $cert_{q,\mathcal{O}}^{\mathcal{A}} = \{(c_1)\}$), as expected. $\square$

The next theorem shows that, even when the UNA is not adopted, answering CQ$^{\neq,b}$s over *DL-Lite*$_{\mathcal{R}}$ knowledge bases has exactly the same computational complexity of answering UCQs over *DL-Lite*$_{\mathcal{R}}$ knowledge bases.

Theorem 4.3. *Answering CQ$^{\neq,b}$s over DL-Lite$_{\mathcal{R}}$ knowledge bases is FOL-rewritable (and therefore in AC^0 in data complexity) and NP-complete in combined complexity.*

Proof. FOL-rewritability is a direct consequence of Corollary 4.1.

The membership in NP in combined complexity can be easily proven with a version of the $\mathsf{NoUNAPerfectRef}$ algorithm that nondeterministically guess a CQ of the final UCQ $\mathsf{NoUNAPerfectRef}(\mathcal{O}, q)$, which can be generated after a polynomial number of transformations of the initial CQ$^{\neq,b}$ q. Finally, NP-hardness already follows from CQ evaluation over relational databases [Abiteboul *et al.*, 1995]. $\square$

4.2.1 Queries with Inequalities in Ontology-based Data Management

We now briefly discuss how the above results behave in the context of the OBDM scenario when the UNA is not adopted. Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification where $\mathcal{O}$ is a *DL-Lite*$_{\mathcal{R}}$ ontology and $\mathcal{M}$ is a GLAV mapping. Clearly, since *DL-Lite*$_{\mathcal{R}}$ is insensitive to the adoption of the UNA for UCQ answering, given a UCQ $q_{\mathcal{O}}$ over $\mathcal{O}$ of arity n, it is not hard to verify that $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \cup \mathsf{PerfRef}_{\mathcal{V}_{\mathcal{O}}^n,\Sigma}$ is the perfect $\mathcal{O}$-to-$\mathcal{S}$ Σ-rewriting of $q_{\mathcal{O}}$ even when the UNA is not adopted, where $\mathcal{V}_{\mathcal{O}}$ is the violation query for $\mathcal{O}$ and $\mathsf{PerfRef}_{q,\Sigma} := \mathsf{MapRef}(\mathsf{PerfectRef}(\mathcal{O}, q), \mathcal{M})$ for a UCQ q over $\mathcal{O}$ (cf. Section 2.6).

Let us now turn to the case of CQ$^{\neq,b}$s. Given a CQ$^{\neq,b}$ $q_{\mathcal{O}}$ over $\mathcal{O}$ of arity n, we denote by $\mathsf{NoUNAPerfRef}_{q_{\mathcal{O}},\Sigma}$ the UCQ over $\mathcal{S}$ obtained by first executing the algorithm $\mathsf{NoUNAPerfectRef}$ on $\mathcal{O}$ and $q_{\mathcal{O}}$, and then by rewriting the obtained UCQ with respect to mapping $\mathcal{M}$, i.e., $\mathsf{NoUNAPerfRef}_{q_{\mathcal{O}},\Sigma} := \mathsf{MapRef}(\mathsf{NoUNAPerfectRef}(\mathcal{O}, q_{\mathcal{O}}), \mathcal{M})$.

Theorem 4.4. *Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification where $\mathcal{O}$ is a DL-Lite$_\mathcal{R}$ ontology and $\mathcal{M}$ is a GLAV mapping. When the UNA is not adopted, given a $CQ^{\neq,b}$ over $\mathcal{O}$, we have that $\mathsf{NoUNAPerfRef}_{q_\mathcal{O},\Sigma} \cup \mathsf{PerfRef}_{\mathcal{V}^n_\mathcal{O},\Sigma}$ is the perfect $\mathcal{O}$-to-$\mathcal{S}$ Σ-rewriting of $q_\mathcal{O}$, i.e., $(\mathsf{NoUNAPerfRef}_{q_\mathcal{O},\Sigma} \cup \mathsf{PerfRef}_{\mathcal{V}^n_\mathcal{O},\Sigma})^D = cert^D_{q_\mathcal{O},\Sigma}$, for every $\mathcal{S}$-database D.*

Proof. Let D be an $\mathcal{S}$-database. If D is inconsistent with Σ, i.e., $Mod_D(\Sigma) = \emptyset$, then $\mathsf{PerfRef}^D_{\mathcal{V}^n_\mathcal{O},\Sigma} = cert^D_{q_\mathcal{O},\Sigma}$, which corresponds to the set of all possible tuples of constants occurring in D whose arity is the one of the query.

On the other hand, if D is consistent with Σ (and therefore $\mathsf{PerfRef}^D_{\mathcal{V}^n_\mathcal{O},\Sigma} = \emptyset$), then, due to Corollary 4.1, we have $cert^D_{q_\mathcal{O},\Sigma} = cert^D_{q',\Sigma'}$, where $q' = \mathsf{NoUNAPerfectRef}(\mathcal{O}, q_\mathcal{O})$ and $\Sigma' = \langle \emptyset, \mathcal{S}, \mathcal{M} \rangle$. But then, since q' is a UCQ and the ontology of $\Sigma' = \langle \emptyset, \mathcal{S}, \mathcal{M} \rangle$ contains no assertions, $cert^D_{q',\Sigma'} = \mathsf{MapRef}(q', \mathcal{M})^D$ (cf. Section 2.6). It follows that $cert^D_{q_\mathcal{O},\Sigma} = \mathsf{MapRef}(\mathsf{NoUNAPerfectRef}(\mathcal{O}, q_\mathcal{O}), \mathcal{M})^D$, i.e., $cert^D_{q_\mathcal{O},\Sigma} = \mathsf{NoUNAPerfRef}^D_{q_\mathcal{O},\mathcal{M}}$, as required. □

4.2.2 The case of the Direct Semantics Entailment Regime

The de facto standard query language for the Semantic Web is `SPARQL`[3]. As defined in a W3C standard specification [Glimm and Ogbuji, 2013], the query language `SPARQL 1.1` [Harris and Seaborne, 2013] (i.e., the current version of `SPARQL`) relies on the notion of ***SPARQL** entailment regime*, which defines:

1. the syntax and the semantics of assertions of the queried knowledge base;

2. the syntax of conjunctive queries considered legal for the regime;

3. the semantics of such queries, i.e., what are the answers to a query.

As for 1, we assume to deal with (satisfiable) `OWL 2 QL` (equivalently, *DL-Lite*$_\mathcal{R}$) knowledge bases without the UNA. As for 2, we now consider general $UCQ^{\neq}$s. Finally, as for 3, we follow the `OWL 2 QL` DSER, which is the most typical `SPARQL` Entailment Regime for `OWL 2 QL` (equivalently, *DL-Lite*$_\mathcal{R}$) knowledge bases.

Specifically, given a *DL-Lite*$_\mathcal{R}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ and a $CQ^{\neq}$ $q = \{(t_1, \ldots, t_n) \mid \exists y_1, \ldots, y_m . \phi(\vec{x}, \vec{y})\}$ over $\mathcal{O}$, DSER defines the answers to q with respect to $\mathcal{O}$ and $\mathcal{A}$, denoted by $DScert^\mathcal{A}_{q,\mathcal{O}}$, as follows: an n-tuple of individuals $\vec{c} = (c_1, \ldots, c_n)$ (with $c_j = t_j$ for each $j \in [1, n]$ in which t_j is a constant) is in $DScert^\mathcal{A}_{q,\mathcal{O}}$ if and only if there exists an m-tuple of individuals $\vec{c'} = (c'_1, \ldots, c'_n)$ for which $\mathcal{K} \models q(\vec{c}, \vec{c'})$, where $q(\vec{c}, \vec{c'}) = \{() \mid \phi(\vec{x}/\vec{c}, \vec{y}/\vec{c'})\}$ denotes the boolean $CQ^{\neq}$ in which $\phi(\vec{x}/\vec{c}, \vec{y}/\vec{c'})$ is the body obtained from $\phi(\vec{x}, \vec{y})$ by replacing all the occurrences of the term t_i (respectively, y_i) with the individual c_i (respectively, c'_i), for each $i \in [1, n]$ (respectively, $i \in [1, m]$). In other words, for an n-tuple of individuals $\vec{c}$, in contrast to classical logic, existential variables $\vec{y}$, although projected out from the final answers, are required to be bound to the same m-tuple of constants $\vec{c'}$, in every model of $\mathcal{K}$.

Example 4.6. Let us recall Example 4.5. We have that $(c_1) \notin DScert^\mathcal{A}_{q,\mathcal{O}}$, since there is no y_1 for which $P(c, y_1)$ is known to hold. □

[3] `SPARQL Protocol and RDF Query Language`: https://www.w3.org/2001/sw/wiki/SPARQL

Furthermore, given a *DL-Lite*$_{\mathcal{R}}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ and a UCQ$^{\neq}$ $q = q_1 \cup \ldots \cup q_l$ over $\mathcal{O}$, again differently from the classical logic, we have $DScert^{\mathcal{A}}_{q,\mathcal{O}} = DScert^{\mathcal{A}}_{q_1,\mathcal{O}} \cup \ldots \cup DScert^{\mathcal{A}}_{q_l,\mathcal{O}}$.

Example 4.7. Consider the knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$, where $\mathcal{O} = \emptyset$ and $\mathcal{A} = \{P(a,b)\}$. For the boolean UCQ$^{\neq,b}$ $Q = q_1 \cup q_2$ over $\mathcal{O}$, where $q_1 = \{() \mid P(a,a)\}$ and $q_2 = \{() \mid a \neq b\}$, it is easy to see that both $DScert^{\mathcal{A}}_{q_1,\mathcal{O}} = \emptyset$ and $DScert^{\mathcal{A}}_{q_2,\mathcal{O}} = \emptyset$ hold. For the former, consider a model $\mathcal{M}$ of $\mathcal{K}$ in which $a^{\mathcal{M}} \neq b^{\mathcal{M}}$. For the latter, consider a model $\mathcal{M}$ of $\mathcal{K}$ in which $a^{\mathcal{M}} = b^{\mathcal{M}}$. Therefore, $DScert^{\mathcal{A}}_{Q,\mathcal{O}} = \emptyset$ by definition.

Notice, however, that $\mathcal{K} \models Q$. Indeed, for any model $\mathcal{M}$ of $\mathcal{K}$, $a^{\mathcal{M}} \neq b^{\mathcal{M}}$ implies $\mathcal{M} \models q_2$, whereas $a^{\mathcal{M}} = b^{\mathcal{M}}$ implies $\mathcal{M} \models q_1$. Thus, $cert^{\mathcal{A}}_{Q,\mathcal{O}} = \{()\}$. $\square$

We now present the algorithm DSERPerfectRef for answering UCQ$^{\neq}$s over *DL-Lite*$_{\mathcal{R}}$ knowledge bases under the DSER semantics. Given in input a *DL-Lite*$_{\mathcal{R}}$ ontology $\mathcal{O}$ and a UCQ$^{\neq}$ q over $\mathcal{O}$, DSERPerfectRef returns a reformulated UCQ.

Algorithm 4.1 DSERPerfectRef

Input:
DL-Lite$_{\mathcal{R}}$ ontology $\mathcal{O}$;
UCQ$^{\neq}$ $q = q_1 \cup \ldots \cup q_n$ over $\mathcal{O}$

Output:
UCQ q_r over $\mathcal{O}$

$q_r := \emptyset$
for $i \leftarrow 1$ to n **do**
  Let $q_i = \{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$
  Compute $\lambda(q'_i, \mathcal{O})$, where $q'_i = \{(\vec{t}, \vec{y}) \mid \phi(\vec{x}, \vec{y})\}$
  for each CQ $q' \in$ PerfectRef$(\mathcal{O}, \lambda(q'_i, \mathcal{O}))$ **do**
    Let $q' = \{(\vec{t}, \vec{y}) \mid \phi(\vec{x}, \vec{y})\}$
    $q_r := q_r \cup q$, where $q = \{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$
  end for
end for
return q_r

Roughly speaking, according to the meaning of the union operator under DSER, the algorithm treats each CQ of q separately. In particular, for each CQ $q_i = \{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$ of q, according to the meaning of the existential quantifiers under DSER, the algorithm consider the CQ $q'_i = \{(\vec{t}, \vec{y}) \mid \phi(\vec{x}, \vec{y})\}$, i.e., the query obtained from q in which existential variables $\vec{y}$ become distinguished (and therefore bound). After that, it rewrites inequality atoms through the function λ to capture only those inequalities logically implied. Finally, for each CQ in PerfectRef$(\mathcal{O}, \lambda(q'_i, \mathcal{O}))$, the algorithm projects those variables that were originally existential variables.

Example 4.8. Consider the *DL-Lite*$_{\mathcal{R}}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$, where

- $\mathcal{O} = \{$ Male $\sqsubseteq \neg$Female, Pregnant $\sqsubseteq$ Female, Employee $\sqsubseteq \exists$WorksFor $\}$

- $\mathcal{A} = \begin{array}{l} \{ \quad \mathsf{WorksFor}(\text{John}, \text{HOPE}), \mathsf{WorksFor}(\text{Cleo}, \text{HOPE}), \mathsf{Employee}(\text{Alice}), \\ \quad\; \mathsf{Parent}(\text{John}, \text{Bill}), \mathsf{Parent}(\text{John}, \text{Wendy}), \mathsf{Parent}(\text{Alice}, \text{Bill}), \\ \quad\; \mathsf{Parent}(\text{Alice}, \text{Wendy}), \mathsf{Parent}(\text{Cleo}, \text{Alex}), \mathsf{Parent}(\text{Cleo}, \text{Alexander}), \\ \quad\; \mathsf{Male}(\text{Bill}), \mathsf{Pregnant}(\text{Wendy}) \; \}. \end{array}$

Consider also the following $\text{CQ}^{\neq}$ q over $\mathcal{O}$ asking for those people working for some company and having at least two children:

$$q = \{(x) \mid \exists y_1, y_2, y_3.\mathsf{WorksFor}(x, y_1) \wedge \mathsf{Parent}(x, y_2) \wedge \mathsf{Parent}(x, y_3) \wedge y_2 \neq y_3\}.$$

As a first step, $\mathsf{DSERPerfectRef}(\mathcal{O}, q_{\mathcal{O}})$ computes the UCQ $\lambda(q', \mathcal{O})$, where $q' = \{(x, y_1, y_2, y_3) \mid \mathsf{WorksFor}(x, y_1) \wedge \mathsf{Parent}(x, y_2) \wedge \mathsf{Parent}(x, y_3) \wedge y_2 \neq y_3\}$. Note that $\lambda(q', \mathcal{O})$ contains, among others, the CQ $\{(x, y_1, y_2, y_3) \mid \mathsf{WorksFor}(x, y_1) \wedge \mathsf{Parent}(x, y_2) \wedge \mathsf{Parent}(x, y_3) \wedge \mathsf{Male}(y_2) \wedge \mathsf{Female}(y_3)\}$.

Then, $\mathsf{PerfectRef}(\mathcal{O}, \lambda(q', \mathcal{O}))$ computes, among others, the CQ $\{(x, y_1, y_2, y_3) \mid \mathsf{WorksFor}(x, y_1) \wedge \mathsf{Parent}(x, y_2) \wedge \mathsf{Parent}(x, y_3) \wedge \mathsf{Male}(y_2) \wedge \mathsf{Pregnant}(y_3)\}$. Note that, since y_1 is a distinguished variable in q', the inclusion assertion $\mathsf{Employee} \sqsubseteq \exists\mathsf{WorksFor}$ is not used to rewrite q' with respect to $\mathcal{O}$.

As a last step, the algorithm returns the UCQ which contains, among others, the CQ $q_1 = \{(x) \mid \exists y_1, y_2, y_3.\mathsf{WorksFor}(x, y_1) \wedge \mathsf{Parent}(x, y_2) \wedge \mathsf{Parent}(x, y_3) \wedge \mathsf{Male}(y_2) \wedge \mathsf{Pregnant}(y_3)\}$. Observe that $q_1^{\mathcal{I}_{\mathcal{A}}} = \{(\text{John})\}$. One can indeed verify that $\mathit{DScert}_{q,\mathcal{O}}^{\mathcal{A}} = \{(\text{John})\}$, as expected. Finally, note that $(\text{Cleo}) \notin \mathit{DScert}_{q,\mathcal{O}}^{\mathcal{A}}$ and $(\text{Alice}) \notin \mathit{DScert}_{q,\mathcal{O}}^{\mathcal{A}}$ since Alex and Alexander are not known to be different individuals and it is not known which is the project Alice works for, respectively. $\square$

From the results given previously for $\text{CQ}^{\neq,b}$s, and from the considerations of how the algorithm capture the DSER, we immediately derive the following result.

Proposition 4.3. *Let $\mathcal{O}$ be a DL-Lite$_{\mathcal{R}}$ ontology, and let q be a $\text{UCQ}^{\neq}$ over $\mathcal{O}$. For every ABox $\mathcal{A}$ such that the DL-Lite$_{\mathcal{R}}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ is satisfiable, we have $\mathit{DScert}_{q,\mathcal{O}}^{\mathcal{A}} = \mathsf{DSERPerfectRef}(\mathcal{O}, q)^{\mathcal{I}_{\mathcal{A}}}$.*

We also get that answering $\text{UCQ}^{\neq}$s over *DL-Lite*$_{\mathcal{R}}$ knowledge bases under DSER has exactly the same computational complexity of answering UCQs over *DL-Lite*$_{\mathcal{R}}$ knowledge bases.

Theorem 4.5. *Answering $\text{UCQ}^{\neq}$s over DL-Lite$_{\mathcal{R}}$ knowledge bases under DSER is FOL-rewritable (and therefore in AC^0 in data complexity) and NP-complete in combined complexity.*

Related to this problem is the result in [Gutiérrez-Basulto *et al.*, 2015], which shows that answering even $\text{CQ}^{\neq}$s over *DL-Lite*$_{\mathcal{R}}$ knowledge bases is in general undecidable. We point out, however, that such a negative result is due to the fact that existential variables are assigned the standard logical meaning, which is not the case in DSER. To the best of our knowledge, only few works [Kontchakov *et al.*, 2014; Gottlob and Pieris, 2015; Poggi, 2016; Arenas *et al.*, 2018] have investigated the problem of answering queries over *DL-Lite*$_{\mathcal{R}}$ knowledge bases under DSER. With the only exception of [Poggi, 2016] (which shows that the problem of answering $\text{UCQ}^{\neq}$s over *DL-Lite*$_{\mathcal{R}}$ knowledge bases under DSER can be reduced to the evaluation of a

Datalog program, and therefore is in PTime in data complexity and in ExpTime in combined complexity), none of the other works consider queries possibly containing inequalities. In fact, in [Gottlob and Pieris, 2015] it is even claimed that inequalities cannot occur within legal `OWL 2 QL` *basic graph patterns*, which is a strong assumption, because inequalities can be expressed by means of `DifferentIndividuals` atoms.

4.3 Answering UCQ$^{\neq,b}$s over *DL-Lite*$_{\mathcal{R}}$ knowledge bases

In this section, we consider the problem of answering UCQ$^{\neq,b}$s over satisfiable *DL-Lite*$_{\mathcal{R}}$ knowledge bases. The next example shows that, differently from the UCQ case, the certain answers of UCQ$^{\neq,b}$s over *DL-Lite*$_{\mathcal{R}}$ knowledge bases is not necessarily equivalent to the union of the certain answers of the singles CQs.

Example 4.9. Recall Example 4.7, where $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ with $\mathcal{O} = \emptyset$ and $\mathcal{A} = \{P(a, b)\}$. For the boolean UCQ$^{\neq,b}$ $Q = q_1 \cup q_2$ over $\mathcal{O}$, where $q_1 = \{() \mid P(a, a)\}$ and $q_2 = \{() \mid a \neq b\}$, it is easy to see that both $\mathcal{K} \not\models q_1$ and $\mathcal{K} \not\models q_2$. For the former, consider a model $\mathcal{M}$ of $\mathcal{K}$ in which $a^{\mathcal{M}} \neq b^{\mathcal{M}}$. For the latter, consider a model $\mathcal{M}$ of $\mathcal{K}$ in which $a^{\mathcal{M}} = b^{\mathcal{M}}$. Notice, however, that $\mathcal{K} \models Q$. Indeed, for any model $\mathcal{M}$ of $\mathcal{K}$, $a^{\mathcal{M}} \neq b^{\mathcal{M}}$ implies $\mathcal{M} \models q_2$, and $a^{\mathcal{M}} = b^{\mathcal{M}}$ implies $\mathcal{M} \models q_1$. □

The above example also tells us that answering UCQ$^{\neq,b}$s over *DL-Lite*$_{\mathcal{R}}$ knowledge bases $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ can not be done simply using the interpretation $\mathcal{I}_{\mathcal{K}}$. Indeed, one can verify that $\delta(Q) = \delta(q_1) \cup \delta(q_2)$ is such that $\mathcal{I}_{\mathcal{K}} \not\models \delta(Q)$, whereas $\mathcal{K} \models Q$.

The conceptual tool we use for addressing the problem of answering queries with inequalities over *DL-Lite*$_{\mathcal{R}}$ knowledge base is the notion of *equivalence relation*.[4] Specifically, we now introduce the notions of e-satisfiability and e-entailment for an equivalence relation e. In what follows, for an equivalence relation e, we write $c_1 \sim_e c_2$ to actually denote $(c_1, c_2) \in e$.

Definition 4.1. Let $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ be a knowledge base, e be an equivalence relation on a set I of individuals occurring in $\mathcal{A}$, and $\mathcal{M}$ be a model of $\mathcal{K}$. Then, we say that $\mathcal{M}$ is an *e-model* of $\mathcal{K}$, denoted by $\mathcal{M} \models_e \mathcal{K}$, if for any pair of individuals c_1, c_2 occurring in I, we have that $c_1^{\mathcal{M}} = c_2^{\mathcal{M}}$ if and only if $c_1 \sim_e c_2$. Furthermore, we say that $\mathcal{K}$ is *e-satisfiable* if it has an e-model.

Definition 4.2. Let $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ be a knowledge base, q be a boolean query over $\mathcal{O}$, and e be an equivalence relation on a set I of individuals occurring in $\mathcal{A}$. Then, we say that $\mathcal{K}$ *e-entails* q, denoted by $\mathcal{K} \models_e q$, if $\mathcal{M} \models q$ for each e-model $\mathcal{M}$ of $\mathcal{K}$.

Observe that, for a knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$, if e is the equivalence relation on the set of all individuals occurring in $\mathcal{A}$ such that $e = \{(c, c) \mid c \text{ occurs in } \mathcal{A}\}$ (respectively, e is the equivalence relation on the empty set of individuals, i.e., $e = \{\}$), then the notions of e-satisfiability and e-entailment coincide with the usual notion of satisfiability and entailment, respectively, when the UNA is adopted (respectively, when the UNA is not adopted).

[4] An equivalence relation e on a set of individuals I is a binary relation over I that is reflexive, symmetric, and transitive.

In what follows, for a knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ and an equivalence relation e, we denote by $\mathcal{K}_e = \langle \mathcal{O}_e, \mathcal{A}_e \rangle$ the knowledge base obtained from $\mathcal{K}$ by adding e in the alphabet of $\mathcal{O}$ as a new atomic role (obtaining the ontology $\mathcal{O}_e$), and $\mathcal{A}_e$ is the ABox for $\mathcal{O}_e$ such that $\mathcal{A}_e = \mathcal{A} \cup e$, i.e., $\mathcal{A}_e = \mathcal{A} \cup \{e(c_1, c_2) \mid (c_1, c_2) \in e\}$. Moreover, for a query q over an ontology $\mathcal{O}$, we denote by I_q the set of all individuals occurring in q.

We now prove that, in principle, the notion of e-entailment can be used for the problem of answering UCQ$^{\neq,b}$s over *DL-Lite*$_{\mathcal{R}}$ knowledge bases.

Proposition 4.4. *Let $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ be a DL-Lite$_{\mathcal{R}}$ knowledge base, and q be a boolean UCQ$^{\neq,b}$. We have that $\mathcal{K} \not\models q$ if and only if there exists an equivalence relation e on I_q such that $\mathcal{K} \not\models_e q$.*

Proof. "**If part:**" Suppose that there exists an equivalence relation e on I_q such that $\mathcal{K} \models_e q$. This implies that there is an e-model $\mathcal{M}$ of $\mathcal{K}$ such that $\mathcal{M} \not\models q$. Since $\mathcal{M}$ is an e-model of $\mathcal{K}$, then $\mathcal{M}$ is also a model of $\mathcal{K}$. Thus, $\mathcal{K} \not\models q$, as required.

"**Only-if part:**" Suppose that $\mathcal{K} \not\models q$. It follows that there is a model $\mathcal{M}$ of $\mathcal{K}$ for which $\mathcal{M} \not\models q$. Consider now the equivalence relation e on I_q given by the model $\mathcal{M}$, that is, $(c_1, c_2) \in e$ if and only if $c_1^{\mathcal{M}} = c_2^{\mathcal{M}}$ and $c_1, c_2 \in I_q$.

Let $\mathcal{I}$ be the interpretation for $\mathcal{K}$ similar to $\mathcal{M}$ but, for each individual $c \notin I_q$, we set $c^{\mathcal{I}} = c$ and replace each occurrence of the object $c^{\mathcal{M}}$ with $c^{\mathcal{I}} = c$. Since $\mathcal{M} \not\models q$, it can be easily verified that $\mathcal{I} \not\models q$ as well. In particular, since $c^{\mathcal{I}} = c^{\mathcal{M}}$ for each $c \in I_q$ (i.e., for each individual appearing in the query), the evaluation of each inequality atom of q (since q is a boolean UCQ$^{\neq,b}$, the inequality atoms are only between individuals) over $\mathcal{I}$ and $\mathcal{M}$ coincide. Notice, moreover, that since $\mathcal{M}$ is a model of $\mathcal{K}$, we have that $\mathcal{I}$ is a model of $\mathcal{K}$ as well, and therefore an e-model of $\mathcal{K}$ by construction. So, $\mathcal{I}$ is an e-model of $\mathcal{K}$ such that $\mathcal{I} \not\models q$, and therefore $\mathcal{K} \not\models_e q$.

To conclude the proof, observe that e is an equivalence relation on the set of individuals I_q such that $\mathcal{K} \not\models_e q$, as required. □

The above proposition suggests a nondeterministic algorithm for the problem of answering UCQ$^{\neq,b}$s over *DL-Lite*$_{\mathcal{R}}$ knowledge bases, which consists in guessing an equivalence relation e on I_q, and then checking whether $\mathcal{K} \not\models_e q$, where $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ and q are the input *DL-Lite*$_{\mathcal{R}}$ knowledge base and UCQ$^{\neq,b}$ over $\mathcal{O}$, respectively. So, for the case of *DL-Lite*$_{\mathcal{R}}$ knowledge bases $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$, we now study the problems of (i) checking whether $\mathcal{K}$ is e-satisfiable for an arbitrary equivalence relation e on a set I of individuals of $\mathcal{A}$; (ii) checking whether $\mathcal{K} \models_e q$ for a boolean UCQ$^{\neq,b}$ q over $\mathcal{O}$ and an arbitrary equivalence relation e on I_q such that $\mathcal{K}$ is e-satisfiable. Specifically, we next show that the computational complexity of both the above problems remains the same of the case when e simulates the adoption of the UNA.

We start with the e-satisfiability check. We assume to deal with satisfiable knowledge bases (notice that e-satisfiability always implies satisfiability, and therefore, in case of unsatisfiable knowledge bases, e-satisfiability becomes trivial).

Proposition 4.5. *Let $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ be a DL-Lite$_{\mathcal{R}}$ knowledge base and e be an equivalence relation on a set I of individuals occurring in $\mathcal{A}$. The problem of checking whether $\mathcal{K}$ is e-satisfiable is in AC^0 in the size of $\mathcal{A}$ and e, and in* PTIME *in the size of $\mathcal{O}$.*

Proof. Let $\mathcal{V}^e_{\mathcal{O}}$ be the e-violation query for $\mathcal{O}$ obtained from $\mathcal{V}_{\mathcal{O}}$ by adding the following disjuncts over the signature of $\mathcal{O}_e$:

- $\{() \mid \exists y_1, y_2 . A_1(y_1) \wedge A_2(y_2) \wedge e(y_1, y_2)\}$ for each disjointness assertion of the form $A_1 \sqsubseteq \neg A_2$ belonging to $\mathcal{O}$,
- $\{() \mid \exists y_1, y_2, y_3 . A(y_1) \wedge R(y_2, y_3) \wedge e(y_1, y_2)\}$ for each disjointness assertion of the form $A \sqsubseteq \neg\exists R$ or of the form $\exists R \sqsubseteq \neg A$ belonging to $\mathcal{O}$,
- $\{() \mid \exists y_1, y_2, y_3, y_4 . R_1(y_1, y_3) \wedge R_2(y_2, y_4) \wedge e(y_1, y_2)\}$ for each disjointness assertion of the form $\exists R_1 \sqsubseteq \neg\exists R_2$ belonging to $\mathcal{O}$,
- $\{() \mid \exists y_1, y_2, y_3, y_4 . R_1(y_1, y_3) \wedge R_2(y_2, y_4) \wedge e(y_1, y_2) \wedge e(y_3, y_4)\}$ for each disjointness assertion of the form $R_1 \sqsubseteq \neg R_2$ belonging to $\mathcal{O}$,

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., $R = P^-$.

Intuitively, for checking e-satisfiability it is sufficient to check whether the equivalence relation e contradicts a disjointness assertion of $\mathcal{O}$. We now prove that $\mathcal{K}$ is e-satisfiable if and only if $\mathcal{K}_e \not\models \mathcal{V}^e_{\mathcal{O}}$.

If $\mathcal{K}_e \models q$ for some disjunct q of $\mathcal{V}^e_{\mathcal{O}}$, then it is easy to see that there are at least two individuals $c_1 \sim_e c_2$ of I such that $\mathcal{I} \not\models \mathcal{K}$ in each interpretation $\mathcal{I}$ with $c_1^{\mathcal{I}} = c_2^{\mathcal{I}}$, and therefore $\mathcal{K}$ is not e-satisfiable, as required. Conversely, if $\mathcal{K}_e \not\models \mathcal{V}^e_{\mathcal{O}}$, then from $\mathcal{I}_{\mathcal{K}}$ it is possible to obtain a model $\mathcal{M}$ where $c_1^{\mathcal{M}} = c_2^{\mathcal{M}}$ for each $(c_1, c_2) \in e$. It is easy to see that $\mathcal{M} \models_e \mathcal{K}$, and therefore $\mathcal{K}$ is e-satisfiable.

Finally, observe that $\mathcal{V}^e_{\mathcal{O}}$ can be constructed in PTime in the size of $\mathcal{O}$, and, as usual, checking whether $\mathcal{K}_e \not\models \mathcal{V}^e_{\mathcal{O}}$ is FOL-rewritable (and therefore in AC^0 in the size of $\mathcal{A}_e$) and in PTime in the size of $\mathcal{O}_e$, thus showing the claim. □

We now focus on the problem of checking whether $\mathcal{K} \models_e q$ for a *DL-Lite*$_{\mathcal{R}}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$, boolean UCQ$^{\neq,b}$ over $\mathcal{O}$, and equivalence relation e on I_q. To this aim, we provide algorithm EquivalenceRelationRef that, given in input a boolean UCQ$^{\neq,b}$ q over $\mathcal{O}$ and an equivalence relation e on I_q, it reformulates q according to e and returns a UCQ over $\mathcal{O}_e$. In the algorithm, we assume that boolean queries may contain `false` and `true` atoms in their body, with their obvious semantics. Furthermore, $\mathfrak{P}(\cdot)$ is a function returning the *power set* of a given set as input. Finally, $\mathsf{ej}(\cdot)$ is the function that takes a CQ q as input, and returns the set of all existential variables of q that are join existential variables, i.e., the existential variables y such that $m_y \geq 2$, where m_y denotes the number of occurrences of the variable y in the body of q.

Intuitively, the algorithm first reformulates the given UCQ$^{\neq,b}$ q into a UCQ by directly evaluating each inequality atom $c_1 \neq c_2$ based on the equivalence relation e (notice that, since q is a boolean UCQ$^{\neq,b}$, each inequality atom $c_1 \neq c_2$ is such that both c_1 and c_2 are individuals).

Afterwards, for each possible set $\mathcal{Y} \in \mathfrak{P}(\mathsf{ej}(q_i))$ of join existential variables occurring in the CQ q_i, the algorithm reformulates q_i by allowing that different occurrences $y^1, y^2, \ldots, y^{m_y}$ of the same existential variable $y \in \mathcal{Y}$ may be possibly mapped to also distinct individuals, with the proviso that such distinct individuals belong to the same equivalence class in e (which is captured by $e(z, z')$ atoms).

Algorithm 4.2 EquivalenceRelationRef

Input:
boolean UCQ$^{\neq,b}$ $q = q_1 \cup \ldots \cup q_n$ over an ontology $\mathcal{O}$;
equivalence relation e on I_q

Output:
UCQ q_e over $\mathcal{O}_e$

$q_e := \emptyset$
$\mathsf{PR} := \emptyset$
for $i \leftarrow 1$ to n **do**
  for each inequality atom $\alpha : c_1 \neq c_2$ occurring in the body of the CQ q_i **do**
    if $c_1 \sim_e c_2$ **then**
      Replace atom α with `false`
    else
      Replace atom α with `true`
    end if
  end for
  $\mathsf{PR} := \mathsf{PR} \cup \{q_i\}$
  for each $\mathcal{Y} \in \mathfrak{P}(\mathsf{ej}(q_i))$ **do**
    $q_{\mathcal{Y}} := q_i$
    for each $y \in \mathcal{Y}$ **do**
      Let $y^1, y^2, \ldots, y^{m_y}$ denote all the occurrences of y in the body of $q_{\mathcal{Y}}$
      for $j \leftarrow 1$ to m_y **do**
        Replace the occurrence y^j of y with a fresh existential variable z_y^j
      end for
      for each pair (k, l) with $0 \leq k < l \leq m_y$ **do**
        Add the atom $e(z_y^k, z_y^l)$ in conjunction to the body of $q_{\mathcal{Y}}$
      end for
    end for
    $\mathsf{PR} := \mathsf{PR} \cup \{q_{\mathcal{Y}}\}$
  end for
end for
for each $q \in \mathsf{PR}$ **do**
  for each individual c occurring in q **do**
    Replace each occurrence of c in q with a fresh existential variable y_c
    Add the atom $e(y_c, c)$ in conjunction to the body of q
  end for
  $q_e := q_e \cup q$
end for
return q_e

An analogous consideration holds for the various individuals occurring in queries, where in the last steps of the algorithm each CQ q is further reformulated to allow that an existential variable y_c may match an individual c' that is not necessarily the original individual c of q, but it is such that $c \sim_e c'$, i.e., an individual that is in the same equivalence class of c in e.

Example 4.10. Consider the knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$, where $\mathcal{O} = \emptyset$, and

$$\mathcal{A} = \{\mathsf{Author}(\text{Nicky}), \mathsf{Author}(\text{Nicolas}), \mathsf{Cited}(\text{Nicolas}, \text{Nicky}), \mathsf{Cited}(\text{Nicky}, \text{Nicolas})\}.$$

Consider also the following UCQ$^{\neq,b}$ over $\mathcal{O}$ asking for the pair of authors that either are known to be different, or have been cited by the same author:

$$q' = \begin{array}{l} \{(x_1, x_2) \mid \mathsf{Author}(x_1) \wedge \mathsf{Author}(x_2) \wedge x_1 \neq x_2\} \cup \\ \{(x_1, x_2) \mid \exists y.\mathsf{Cited}(y, x_1) \wedge \mathsf{Cited}(y, x_2)\}. \end{array}$$

Suppose we want to check whether $(\text{Nicky}, \text{Nicolas}) \in cert^{\mathcal{A}}_{q',\mathcal{O}}$, i.e., whether $\mathcal{K} \models q$, where $q = \{() \mid \mathsf{Author}(\text{Nicky}), \mathsf{Author}(\text{Nicolas}) \wedge \text{Nicky} \neq \text{Nicolas}\} \cup \{() \mid \exists y.\mathsf{Cited}(y, \text{Nicky}) \wedge \mathsf{Cited}(y, \text{Nicolas})\}$. By exploiting Proposition 4.4, we consider the two possible equivalence relation on I_q:

- $e_1 = \{(\text{Nicky}, \text{Nicky}), (\text{Nicolas}, \text{Nicolas})\}$;
- $e_2 = \{(\text{Nicky}, \text{Nicky}), (\text{Nicolas}, \text{Nicolas}), (\text{Nicky}, \text{Nicolas}), (\text{Nicolas}, \text{Nicky})\}$.

As for e_1, we easily derive that $\mathcal{K} \models_{e_1} q$ because $\text{Nicky}^{\mathcal{I}} \neq \text{Nicolas}^{\mathcal{I}}$ in each e_1-model $\mathcal{I}$ of $\mathcal{K}$. Moreover, one can easily verify that $\mathcal{K}_{e_1} \models q_{e_1}$ for $q_{e_1} = \mathsf{EquivalenceRelationRef}(q, e_1)$, since the inequality atom Nicky $\neq$ Nicolas of the first disjunct of q is replaced with the atom `true` in q_{e_1}.

As for e_2, we have $\mathcal{K} \models_{e_2} q$ as well, since in each model $\mathcal{I}$ of $\mathcal{K}$ for which $\text{Nicky}^{\mathcal{I}} = \text{Nicolas}^{\mathcal{I}} = o \in \Delta^{\mathcal{I}}$ we have that object o has been cited by itself. Moreover, observe that the second disjunct of $q_{e_2} = \mathsf{EquivalenceRelationRef}(q, e_2)$ is

$$\begin{array}{ll} \{() \mid & \exists z_y^1, z_y^2, y_{\text{Nicky}}, y_{\text{Nicolas}}.\mathsf{Cited}(z_y^1, y_{\text{Nicky}}) \wedge \mathsf{Cited}(z_y^2, y_{\text{Nicolas}}) \wedge \\ & e(z_y^1, z_y^2) \wedge e(y_{\text{Nicky}}, \text{Nicky}) \wedge e(y_{\text{Nicolas}}, \text{Nicolas})\}. \end{array}$$

One can easily verify that, as expected, $\mathcal{K}_{e_2} \models q_{e_2}$ with the bindings for the above disjunct $z_y^1 = y_{\text{Nicolas}} \to \text{Nicolas}$, and $z_y^2 = y_{\text{Nicky}} \to \text{Nicky}$.

We conclude that $\mathcal{K} \models q$, and thus $(\text{Nicky}, \text{Nicolas}) \in cert^{\mathcal{A}}_{q',\mathcal{O}}$. □

The next crucial proposition states that checking whether $\mathcal{K} \models_e q$, for a given *DL-Lite*$_{\mathcal{R}}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$, boolean UCQ$^{\neq,b}$ q over $\mathcal{O}$, and equivalence relation e on I_q such that $\mathcal{K}$ is e-satisfiable, can be reduced to checking whether $\mathcal{K}_e \models q_e$, where q_e is the UCQ returned by $\mathsf{EquivalenceRelationRef}(q, e)$.

Proposition 4.6. *Let $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ be a DL-Lite$_{\mathcal{R}}$ knowledge base, q be a boolean UCQ$^{\neq,b}$ over $\mathcal{O}$, and e be an equivalence relation e on I_q such that $\mathcal{K}$ is e-satisfiable. We have that $\mathcal{K} \models_e q$ if and only if $\mathcal{K}_e \models q_e$, where q_e is the UCQ $\mathsf{EquivalenceRelationRef}(q, e)$.*

Proof. "**If part:**" Suppose that $\mathcal{K}_e \models q_e$. Since $\mathcal{K}$ is e-satisfiable by assumption (and therefore $\mathcal{K}_e$ is a satisfiable *DL-Lite*$_{\mathcal{R}}$ knowledge base) and q_e is a UCQ, due to [Calvanese *et al.*, 2007b, Theorem 29], there is a disjunct $q_{\mathcal{Y}}$ of q_e for which there is a homomorphism h from $q_{\mathcal{Y}}$ to $\mathcal{C}^{\mathcal{A}_e}_{\mathcal{O}_e}$, where $\mathcal{C}^{\mathcal{A}_e}_{\mathcal{O}_e}$ is the canonical structure of $\mathcal{O}_e$ with respect to $\mathcal{A}_e$, and $q_{\mathcal{Y}}$ is a CQ obtained from some disjunct q_i of q by reformulating it according to e (thus removing inequality atoms) and the set $\mathcal{Y} \in \mathfrak{P}(\mathsf{ej}(q_i))$ of join existential variables of q_i. We now prove that $\mathcal{M} \models q_i$ for each e-model $\mathcal{M}$ of $\mathcal{K}$, thus implying that $\mathcal{K} \models_e q$.

Consider any e-model $\mathcal{M}$ of $\mathcal{K}$, and let ψ be the function satisfying conditions 1 and 2 of Proposition 4.1. As a first consideration, note that each possible inequality atom of q (which is only between individuals) is evaluated exactly as in the algorithm over the model $\mathcal{M}$. Furthermore, even if two variables z_y^k and z_y^l of $q_{\mathcal{Y}}$ that replace an existential variable $y \in \mathcal{Y}$ of q_i may match two distinct individuals in $\mathcal{C}_{\mathcal{O}_e}^{\mathcal{A}_e}$ under h (i.e., $h(z_y^k) \neq h(z_y^l)$), the facts that $e(h(z_y^k), h(z_y^l)) \in \mathcal{C}_{\mathcal{O}_e}^{\mathcal{A}_e}$ and that $\mathcal{M}$ is an e-model of $\mathcal{K}$ implies that ${z_y^k}^{\mathcal{M}} = {z_y^l}^{\mathcal{M}}$. An analogous consideration holds also for variables y_c of $q_{\mathcal{Y}}$ replacing all occurrences of individuals $c \in I_q$ of q_i.

But then, consider the function $f_{\mathcal{M}}$ with (i) $f_{\mathcal{M}}(y) = \psi(h(z_y^i))$ for an arbitrary $i \in [1, m_y]$, for each $y \in \mathcal{Y}$, (ii) $f_{\mathcal{M}}(y) = \psi(h(y))$, for each $y \notin \mathcal{Y}$, and (iii) $f(c) = \psi(h(y_c))$, for each individual $c \in I_{q_i}$. It is not hard to see that $f_{\mathcal{M}}$ consists in a homomorphism from q_i to $\mathcal{M}$, and therefore $\mathcal{M} \models q_i$ as required.

"**Only-if part:**" Suppose that $\mathcal{K} \models_e q$. Consider the *DL-Lite*$_{\mathcal{R}}$ knowledge base $\mathcal{K}' = \langle \mathcal{O}, \mathcal{A}' \rangle$ and the UCQ$^{\neq,b}$ q', where $\mathcal{A}'$ and q' are the ABox and the query, respectively, obtained from $\mathcal{A}$ and q by replacing each individual $c \in I_q$ with a fresh individual denoting its equivalence class in e. Obviously, since $\mathcal{K} \models_e q$, we have that $\mathcal{K}' \models q'$. In fact, there is a homomorphism h from some disjunct q_i' of q' to $\mathcal{C}_{\mathcal{O}}^{\mathcal{A}'}$ with $h(c) = c$ for each individual $c \in I_{q'}$, and therefore $h(c_1) \neq h(c_2)$ for each inequality atom $c_1 \neq c_2$ of q_i'.

Let now $\mathcal{Y} \in \mathfrak{P}(\mathsf{ej}(q_i))$ be the set of join existential variables of q_i' for which $y \in \mathcal{Y}$ if and only if $h(y)$ is one of the fresh individual introduced in $\mathcal{A}'$. It is not hard to ascertain that, if we reformulate the disjunct q_i of q with respect to $\mathcal{Y}$ as described in the algorithm, then we obtain a CQ $q_{\mathcal{Y}}$ for which there is a homomorphism from $q_{\mathcal{Y}}$ to $\mathcal{C}_{\mathcal{O}_e}^{\mathcal{A}_e}$, thus implying that $\mathcal{K}_e \models q_{\mathcal{Y}}$. It follows that $\mathcal{K}_e \models q_e$, as required. □

Based on this results, we are now ready to characterise the computational complexity of checking whether $\mathcal{K} \models_e q$ for a *DL-Lite*$_{\mathcal{R}}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$, boolean UCQ$^{\neq,b}$ q, and equivalence relation e on I_q.

Theorem 4.6. *Let $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ be a DL-Lite$_{\mathcal{R}}$ knowledge base, q be a boolean UCQ$^{\neq,b}$ over $\mathcal{O}$, and e be an equivalence relation e on I_q. The problem of checking whether $\mathcal{K} \models_e q$ is NP-complete, and in AC^0 in the size of $\mathcal{A}$ and e.*

Proof. Checking whether $\mathcal{K} \models_e q$ can be done by first checking whether $\mathcal{K}$ is e-satisfiable, and then, by exploiting Proposition 4.6, checking whether $\mathcal{K}_e \models q_e$, where q_e is the UCQ returned by $\mathsf{EquivalenceRelationRef}(q, e)$. Notice that, due to Proposition 4.5, the e-satisfiability check can be done in AC^0 in the size of $\mathcal{A}$ and e, and in polynomial time in the size of $\mathcal{O}$. Moreover, as usual, the problem of checking whether $\mathcal{K}_e \models q_e$ for a *DL-Lite*$_{\mathcal{R}}$ knowledge base $\mathcal{K}_e = \langle \mathcal{O}_e, \mathcal{A}_e \rangle$ and a UCQ q_e is in AC^0 in the size of $\mathcal{A}_e$ and in NP in combined complexity.

Thus, as for the claim of the theorem, the membership in AC^0 in the size of $\mathcal{A}$ and e is a direct consequence of Propositions 4.5 and 4.6, whereas the membership in NP in the size of the input can be easily proven by considering a version of the $\mathsf{EquivalenceRelationRef}$ algorithm that nondeterministically guess a disjunct q_i of q and a subset $\mathcal{Y} \in \mathfrak{P}(\mathsf{ej}(q_i))$, and then with a further NP-step checks whether the reformulated q_i according to $\mathcal{Y}$ is logically implied by $\mathcal{K}_e$.

Finally, NP-hardness follows from CQ evaluation over relational databases. $\square$

With the above result at hand, and by recalling Proposition 4.4, we are ready to show an upper bound for our problem. Notably, answering UCQ$^{\neq,b}$s over *DL-Lite*$_{\mathcal{R}}$ knowledge bases has the same data complexity of the UCQs and CQ$^{\neq,b}$s cases.

Theorem 4.7. *Answering UCQ$^{\neq,b}$s over DL-Lite$_{\mathcal{R}}$ knowledge bases is in AC^0 in data complexity and in Π_2^p in combined complexity.*

Proof. For a *DL-Lite*$_{\mathcal{R}}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ and a UCQ$^{\neq,b}$ q over $\mathcal{O}$, we now show how to decide $\mathcal{K} \not\models q$ in AC^0 in data complexity and in Σ_2^p in combined complexity, thus proving the claim. Due to Proposition 4.4, $\mathcal{K} \not\models q$ can be decided as follows:

1. Guess an equivalence relation e on I_q;
2. If $\mathcal{K} \not\models_e q$, return true (i.e., $\mathcal{K} \not\models q$), otherwise return false (i.e., $\mathcal{K} \models q$),

where, due to Theorem 4.6, this last step can be done in AC^0 in the size of $\mathcal{A}$ and e, and in coNP in the size of $\mathcal{A}$, e, and q.

So, we easily derive a nondeterministic algorithm deciding $\mathcal{K} \not\models q$ that first requires an NP step in the size of q in order to guess an equivalence relation e on I_q (observe that the equivalence relation is only on the set I_q of individuals occurring in q, and thus such step is constant time in the size of $\mathcal{A}$). Finally, with a single call to an coNP-oracle, the algorithm checks whether $\mathcal{K} \not\models_e q$, where, as already said, this check can be done in AC^0 in the size of $\mathcal{A}$ and e. $\square$

We conclude the analysis of the considered problem of this section by establishing that the Π_2^p combined complexity upper bound is tight. Thus, unless the polynomial hierarchy collapses to the first level, answering UCQ$^{\neq,b}$s over *DL-Lite*$_{\mathcal{R}}$ knowledge bases does not have the same combined complexity of the UCQs and CQ$^{\neq,b}$s cases.

Theorem 4.8. *Answering UCQ$^{\neq,b}$s over DL-Lite$_{\mathcal{R}}$ knowledge bases is Π_2^p-hard in combined complexity.*

Proof. The proof is by a LogSpace reduction from the $\forall\exists$-CNF problem, which is Π_2^p-complete [Stockmeyer, 1976]. $\forall\exists$-CNF is the problem of deciding, given a 3-CNF formula $F = c_1 \wedge \ldots \wedge c_p$ on a set of variables $X = \{x_1, \ldots, x_m\} \cup Y = \{y_1, \ldots, y_n\}$ such that the variables in X (respectively, Y) are universally (respectively, existentially) quantified, whether F is true, i.e., whether for each possible truth assignment to the variables in X, there exists a truth assignment to the variables in Y that satisfies F. Each clause c_i is a disjunction of three literals, where each literal is either a variable $z \in X \cup Y$ or its negated. For $i = 1, \ldots, p$, we denote by $z_{i,1}$, $z_{i,2}$, $z_{i,3}$ the first, the second, and the third, respectively, variable appearing (either positive or negated) in clause c_i.

Let F be an instance of the $\forall\exists$-CNF problem. We now construct a *DL-Lite*$_{\mathcal{R}}$ knowledge base $\mathcal{K}_F = \langle \mathcal{O}, \mathcal{A}_F \rangle$ and a UCQ$^{\neq,b}$ Q_F.

For each clause c_i of F, observe that there are exactly seven satisfying truth assignment for the clause c_i. As individuals of $\mathcal{A}_F$ we have 0, 1, an individual x_i for each $i \in [1, m]$, and an individual $A_{i,k}$ for each $i \in [1, p]$ and for each $k \in [1, 7]$.

Intuitively, $A_{i,k}$ denotes the k-th satisfying truth assignment $\{v_{k,1}, v_{k,2}, v_{k,3}\}$ for the clause c_i, where, for $j \in [1,3]$, value $v_{k,j}$ corresponds to the truth assignment (i.e., either 0 or 1) given to the variable $z_{i,j}$. Then, the ABox $\mathcal{A}_F$ is defined as follows:

$$\mathcal{A}_F = \begin{array}{l}\{P_{i,1}(A_{i,k}, v_{k,1}), P_{i,2}(A_{i,k}, v_{k,2}), P_{i,3}(A_{i,k}, v_{k,3}) \mid i \in [1,p] \text{ and } k \in [1,7]\} \cup \\ \{H_i(x_i) \mid i \in [1,m]\},\end{array}$$

where (i) for each $i \in [1,p]$ and for each $j \in [1,3]$, $P_{i,j}$ is an atomic role in the alphabet of $\mathcal{O}$, and (ii) for each universally quantified variable $x_i \in X$, H_i is an atomic concept in the alphabet of $\mathcal{O}$. Finally, the knowledge base $\mathcal{K}_F$ is defined as the pair $\mathcal{K}_F = \langle \mathcal{O}, \mathcal{A}_F \rangle$, where the ontology $\mathcal{O}$ contains no axioms, i.e., $\mathcal{O} = \emptyset$.

As for the UCQ$^{\neq,b}$, let $Q_F = q_1 \cup \ldots \cup q_m \cup q_F$, where $q_i = \{(x_1, \ldots, x_m) \mid H_1(x_1) \wedge \ldots \wedge H_m(x_m) \wedge x_i \neq 0 \wedge x_i \neq 1\}$ for each $i \in [1,m]$, and q_F is

$$\{(x_1, \ldots, x_m) \mid \exists a_1, \ldots, a_p, y_1, \ldots, y_n. \\ \bigwedge_{i \in [1,p]} (P_{i,1}(a_i, z_{i,1}) \wedge P_{i,2}(a_i, z_{i,2}) \wedge P_{i,3}(a_i, z_{i,3})) \wedge \bigwedge_{i \in [1,m]} (H_i(x_i))\}.$$

Observe that $\mathcal{O} = \emptyset$ is fixed, whereas both $\mathcal{A}_F$ and Q_F can be constructed in LogSpace from F. To illustrate the reduction, consider the formula $F = \forall x_1, x_2. \exists y_1, y_2. ((y_1 \vee y_2 \vee x_1) \bigwedge (\neg y_1 \vee \neg y_2 \vee \neg x_2))$. In this case, the reduction produces the knowledge base $\mathcal{K}_F = \langle \mathcal{O}, \mathcal{A}_F \rangle$, where $\mathcal{O} = \emptyset$ and $\mathcal{A}_F$ contains the assertions $\{H_1(x_1), H_2(x_2)\}$ in union to all the the assertions involving atomic roles $P_{i,j}$ for $i \in [1,2]$ and $j \in [1,3]$, which, for ease of exposition, are visualised as the extension of each atomic role $P_{i,j}$ in Figure 4.2.

$P_{1,1}$	$P_{1,2}$	$P_{1,3}$	$P_{2,1}$	$P_{2,2}$	$P_{2,3}$
$A_{1,1}, 0$	$A_{1,1}, 0$	$A_{1,1}, 1$	$A_{2,1}, 0$	$A_{2,1}, 0$	$A_{2,1}, 0$
$A_{1,2}, 0$	$A_{1,2}, 1$	$A_{1,2}, 0$	$A_{2,2}, 0$	$A_{2,2}, 0$	$A_{2,2}, 1$
$A_{1,3}, 0$	$A_{1,3}, 1$	$A_{1,3}, 1$	$A_{2,3}, 0$	$A_{2,3}, 1$	$A_{2,3}, 0$
$A_{1,4}, 1$	$A_{1,4}, 0$	$A_{1,4}, 0$	$A_{2,4}, 0$	$A_{2,4}, 1$	$A_{2,4}, 1$
$A_{1,5}, 1$	$A_{1,5}, 0$	$A_{1,5}, 1$	$A_{2,5}, 1$	$A_{2,5}, 0$	$A_{2,5}, 0$
$A_{1,6}, 1$	$A_{1,6}, 1$	$A_{1,6}, 0$	$A_{2,6}, 1$	$A_{2,6}, 0$	$A_{2,6}, 1$
$A_{1,7}, 1$	$A_{1,7}, 1$	$A_{1,7}, 1$	$A_{2,7}, 1$	$A_{2,7}, 1$	$A_{2,7}, 0$

Figure 4.2. Extension of atomic roles $P_{i,j}$. Each row $A_{i,k}, v_{k,j}$ (with $v_{k,j}$ being either 0 or 1) of an atomic role $P_{i,j}$ has to be read as the ABox assertion $P_{i,j}(A_{i,k}, v_{k,j})$

Finally, the query Q_F produced by the reduction contains the following disjuncts:

$$q_1 = \{(x_1, x_2) \mid H_1(x_1) \wedge H_2(x_2) \wedge x_1 \neq 0 \wedge x_1 \neq 1\} \cup$$
$$q_2 = \{(x_1, x_2) \mid H_1(x_1) \wedge H_2(x_2) \wedge x_2 \neq 0 \wedge x_2 \neq 1\} \cup$$
$$q_F = \{(x_1, x_2) \mid \exists a_1, a_2, y_1, y_2. \\ P_{1,1}(a_1, y_1) \wedge P_{1,2}(a_1, y_2) \wedge P_{1,3}(a_1, x_1) \wedge P_{2,1}(a_2, y_1) \wedge P_{2,2}(a_2, y_2) \wedge \\ P_{2,3}(a_2, x_2) \wedge H_1(x_1) \wedge H_2(x_2)\}.$$

Informally, any model $\mathcal{I}$ of $\mathcal{K}_F$ for which both $x_i^\mathcal{I} \neq 0^\mathcal{I}$ and $x_i^\mathcal{I} \neq 1^\mathcal{I}$ hold for some $i \in [1,m]$ is such that the tuple $(x_1^\mathcal{I}, \ldots, x_m^\mathcal{I}) \in q_i^\mathcal{I}$, and therefore $(x_1^\mathcal{I}, \ldots, x_m^\mathcal{I}) \in Q_F^\mathcal{I}$.

Conversely, if this is not the case, then we can see the model $\mathcal{I}$ as a truth assignment to the universally quantified variables of F. Consider now the query q_F. For each $i \in [1, m]$, because of the presence of atom $H_i(x_i)$ and the ABox assertion $H(x_i)$, the distinguished variable x_i of q_F is either $1^{\mathcal{I}}$ or $0^{\mathcal{I}}$. Then, given a truth assignment to the universally quantified variables of F, the query q_F asks whether there exists a truth assignment to the existentially quantified variables that satisfies formula F.

More formally, let F be any instance of the of the $\forall\exists$-CNF problem. We now prove that $(x_1, \ldots, x_m) \in cert^{\mathcal{A}_F}_{Q_F,\mathcal{O}}$ if and only if F is true.

"**If part:**" Suppose that F is true, that is, for every possible truth assignment to the variables in X, there exists a truth assignment to the variables in Y that satisfies F. Consider any possible model $\mathcal{I}$ for $\mathcal{K}_F$. If both $x_i^{\mathcal{I}} \neq 0^{\mathcal{I}}$ and $x_i^{\mathcal{I}} \neq 1^{\mathcal{I}}$ for some $i \in [1, m]$, then it is straightforward to verify that $(x_1^{\mathcal{I}}, \ldots, x_m^{\mathcal{I}}) \in q_i^{\mathcal{I}}$. Then, let individual x_i be such that either $x_i^{\mathcal{I}} = 0^{\mathcal{I}}$ or $x_i^{\mathcal{I}} = 1^{\mathcal{I}}$, for each $i \in [1, m]$. We can see such interpretation $\mathcal{I}$ as a truth assignment to the variables in X. Specifically, let $V_X = \{v_{x_1}, \ldots, v_{x_m}\}$ be the truth values given by $\mathcal{I}$ to the variables in X, i.e., $v_{x_i} = 1$ if $x_i^{\mathcal{I}} = 1^{\mathcal{I}}$, otherwise (i.e., $x_i^{\mathcal{I}} = 0^{\mathcal{I}}$) $v_{x_i} = 0$.

Consider the partial function h from variables of q_F to $\mathcal{I}$ such that $h(x_i) = x_i^{\mathcal{I}}$. Since for every truth assignment to the variables in X there exists a truth assignment to the variables in Y that satisfies F, let $V_Y = \{v_{y_1}, \ldots, v_{y_n}\}$ be such truth assignment for the variables in Y. For each $i \in [1, p]$, consider the following extension of h: $h(y_i) = v_{y_i}^{\mathcal{I}}$ and $h(a_i) = A_{i,k}^{\mathcal{I}}$ for an arbitrary $k \in [1, 7]$ satisfying (i) $(A_{i,k}^{\mathcal{I}}, h(z_{i,1})) \in P_{i,1}^{\mathcal{I}}$, (ii) $(A_{i,k}^{\mathcal{I}}, h(z_{i,2})) \in P_{i,2}^{\mathcal{I}}$, and (iii) $(A_{i,k}^{\mathcal{I}}, h(z_{i,3})) \in P_{i,3}^{\mathcal{I}}$. Observe that, since by assumption clause c_i is satisfied under the truth assignment $V = V_X \cup V_Y$, at least one individual $A_{i,k}$ for some $k \in [1, 7]$ must exists by construction.

But then, it can be easily verified that h is a homomorphism from q_F to $\mathcal{I}$ such that $h(\vec{x}) = (x_1^{\mathcal{I}}, \ldots, x_m^{\mathcal{I}})$, thus implying $(x_1^{\mathcal{I}}, \ldots, x_m^{\mathcal{I}}) \in q_F^{\mathcal{I}}$. It follows that, for each possible model $\mathcal{I}$ for $\mathcal{K}_F$, either $(x_1^{\mathcal{I}}, \ldots, x_m^{\mathcal{I}}) \in q_i^{\mathcal{I}}$ for some $i \in [1, m]$, or $(x_1^{\mathcal{I}}, \ldots, x_m^{\mathcal{I}}) \in q_F^{\mathcal{I}}$. Thus, $(x_1, \ldots, x_m) \in cert^{\mathcal{A}_F}_{Q_F,\mathcal{O}}$, as required.

"**Only-if part:**" Suppose, for the sake of contradiction, that F is not true, that is, there exists a truth assignment to the variables in X such that every possible truth assignment to the variables in Y does not satisfy F. Let $V_X = \{v_1, \ldots, v_m\}$ be the assignment to the variables in X (v_i is either 1 or 0, for each $i \in [1, m]$) that makes F not satisfiable. Consider the model $\mathcal{I}$ of $\mathcal{K}_F$ such that (i) $\Delta^{\mathcal{I}} = \{0, 1\} \cup \{A_{i,k} \mid i \in [1, p] \text{ and } k \in [1, 7]\}$, (ii) $x_i^{\mathcal{I}} = v_i$ for each $i \in [1, m]$, $0^{\mathcal{I}} = 0$, $1^{\mathcal{I}} = 1$, and $A_{i,k}^{\mathcal{I}} = A_{i,k}$ for each $i \in [1, p]$ and for each $k \in [1, 7]$, and (iii) $P_{i,j}^{\mathcal{I}} = P_{i,j}^{\mathcal{I}_A}$ for each $i \in [1, p]$ and for each $j \in [1, 3]$, and $H_i^{\mathcal{I}} = \{v_i\}$ for each $i \in [1, m]$.

For each $i \in [1, m]$, since individual x_i is such that either $x_i^{\mathcal{I}} = 1^{\mathcal{I}}$ or $x_i^{\mathcal{I}} = 0^{\mathcal{I}}$, we have $q_i^{\mathcal{I}} = \emptyset$. Moreover, due to the fact that, for each $i \in [1, m]$, we have $H_i^{\mathcal{I}} = \{v_i\}$ and $H_i(x_i)$ is an atom of q_F, every possible homomorphism h from q_F to $\mathcal{I}$ must be such that $h(x_i) = v_i$ for each $i \in [1, m]$. Since, however, by assumption F is not satisfiable when replacing variables x_is with truth values v_is, by construction of q_F and $\mathcal{I}$ we easily conclude that $q_F^{\mathcal{I}} = \emptyset$. It follows that, for the model $\mathcal{I}$ of $\mathcal{K}_F$, we have $(x_1^{\mathcal{I}}, \ldots, x_m^{\mathcal{I}}) \notin Q_F^{\mathcal{I}}$. Thus, $(x_1, \ldots, x_m) \notin cert^{\mathcal{A}_F}_{Q_F,\mathcal{O}}$, as required.

Finally, observe that the same proof works even for boolean UCQ$^{\neq,b}$s. To see this, it is sufficient to consider the boolean UCQ$^{\neq,b}$ Q'_F similar to Q_F, but where each disjunct has an empty target list and the distinguished variable x_i is replaced with the corresponding individual x_i of the ABox $\mathcal{A}_F$, for each $i \in [1, m]$. $\square$

Corollary 4.2. *Answering UCQ$^{\neq,b}$s over DL-Lite$_{\mathcal{R}}$ knowledge bases is in AC^0 in data complexity and Π^p_2-complete in combined complexity.*

By looking at the proof of the above theorem, one can see that Π^p_2-hardness holds already for UCQ$^{\neq,b}$s having at most two inequalities per disjunct and for knowledge bases having an empty ontology (i.e., an ontology without assertions).

4.4 Answering UCQ$^{\neq}$s over *DL-Lite*$^{\neg}_{\mathsf{RDFS}}$ knowledge bases

In this section, we study the problem of answering UCQ$^{\neq}$s over satisfiable *DL-Lite*$^{\neg}_{\mathsf{RDFS}}$ knowledge bases. Theorem 4.2 tells us that the certain answers to a CQ$^{\neq,b}$ q over a *DL-Lite*$^{\neg}_{\mathsf{RDFS}}$ knowledge base $\mathcal{K}$ are the tuples of individuals $\vec{c}$ such that $\vec{c} \in \delta(q)^{\mathcal{I}_{\mathcal{K}}}$. The next example shows that the problem drastically changes as soon as we consider general CQ$^{\neq}$s.

Example 4.11. Consider the *DL-Lite*$^{\neg}_{\mathsf{RDFS}}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$, where $\mathcal{O} = \{A_1 \sqsubseteq \neg A_2\}$ and $\mathcal{A} = \{A_1(a_1), A_2(a_2), P(b, c_1), P(b, c_2), P(c_1, a_1), P(c_2, a_2)\}$. For the boolean CQ$^{\neq}$ $q = \{() \mid \exists y_1, y_2, y_3 . P(y_1, y_2) \wedge P(y_1, y_3) \wedge y_2 \neq y_3\}$, we have that $\mathcal{I}_{\mathcal{K}} \not\models \delta(q)$ because $ineq(c_1, c_2) \notin \mathcal{I}_{\mathcal{K}}$ (regarding the *ineq* predicate, only $ineq(a_1, a_2)$ and $ineq(a_2, a_1)$ are in $\mathcal{I}_{\mathcal{K}}$). Notice, however, that $\mathcal{K} \models q$. Indeed, in each model $\mathcal{M}$ with $c_1^{\mathcal{M}} \neq c_2^{\mathcal{M}}$, we have $\mathcal{M} \models q$ with the bindings $y_1, y_2, y_3 \to b, c_1, c_2$, whereas, in each model $\mathcal{M}$ with $c_1^{\mathcal{M}} = c_2^{\mathcal{M}}$, we have $\mathcal{M} \models q$ with the bindings $y_1, y_2, y_3 \to c_1, a_1, a_2$. $\square$

The above example provides a hint on how to design an algorithm for our problem. Intuitively, given a *DL-Lite*$^{\neg}_{\mathsf{RDFS}}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ and a boolean UCQ$^{\neq}$ q over $\mathcal{O}$, we can check whether $\mathcal{K} \not\models q$ by simply guessing an equivalence relation e on the set I of individuals occurring in $\mathcal{A}$ for which $\mathcal{K} \not\models_e q$.

Proposition 4.7. *Let $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ be a DL-Lite$^{\neg}_{\mathsf{RDFS}}$ knowledge base and let q be a boolean UCQ$^{\neq}$ over $\mathcal{O}$. We have that $\mathcal{K} \not\models q$ if and only if there exists an equivalence relation e on a set I of individuals occurring in $\mathcal{A}$ such that $\mathcal{K} \not\models_e q$.*

Proof. "**If part:**" Same as in the proof of the "If part" of Proposition 4.4.

"**Only-if part:**" Suppose that $\mathcal{K} \not\models q$. It follows that there exists a model $\mathcal{M}$ of $\mathcal{K}$ such that $\mathcal{M} \not\models q$. Consider the equivalence relation e on the set I of all individuals of $\mathcal{A}$ such that, for any pair c_1, c_2 of individuals of $\mathcal{A}$, $(c_1, c_2) \in e$ if and only if $c_1^{\mathcal{M}} = c_2^{\mathcal{M}}$. By definition, we have that $\mathcal{M}$ is an e-model of $\mathcal{K}$ for which $\mathcal{M} \not\models q$. Thus, e is the equivalence relation such that $\mathcal{K} \not\models_e q$, as required. $\square$

From the above result, we can derive upper bounds in both data and combined complexity for the problem of answering UCQ$^{\neq}$s over *DL-Lite*$^{\neg}_{\mathsf{RDFS}}$ knowledge bases.

Theorem 4.9. *Answering UCQ$^{\neq}$s over DL-Lite$^{\neg}_{\mathsf{RDFS}}$ ontologies is in coNP in data complexity and in Π^p_2 in combined complexity.*

Proof. For a $DL\text{-}Lite^{\neg}_{\mathsf{RDFS}}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ and a UCQ$^{\neq}$ q over $\mathcal{O}$, we now show how to decide $\mathcal{K} \not\models q$ in NP in data complexity and in Σ_2^p in combined complexity, thus proving the claim. Due to Proposition 4.7, $\mathcal{K} \not\models q$ can be decided as follows:

1. Guess an equivalence relation e on the set I of all individuals of $\mathcal{A}$;
2. If $\mathcal{K} \not\models_e q$, return true (i.e., $\mathcal{K} \not\models q$), otherwise return false (i.e., $\mathcal{K} \models q$),

where, due to the fact that we are considering $DL\text{-}Lite^{\neg}_{\mathsf{RDFS}}$ knowledge base (and thus, $\mathcal{I}_{\mathcal{K}}$ is finite and does not introduce any variable), this last step can be done by simply computing $\mathcal{I}_{\mathcal{K}}$ and, after replacing each occurrence of every individual c in $\mathcal{I}_{\mathcal{K}}$ and in q with a new object denoting its equivalence class in e, we can check whether the resulting $\mathcal{I}_{\mathcal{K}}$ and q are such that (*i*) $\mathcal{I}_{\mathcal{K}} \not\models \delta(q)$ and (*ii*) there is no object o for which $ineq(o, o) \in \mathcal{I}_{\mathcal{K}}$ (i.e., $\mathcal{K}$ is e-satisfiable).

So, we easily derive a nondeterministic algorithm deciding $\mathcal{K} \not\models q$ that first requires an NP step in the size of $\mathcal{A}$ in order to guess an equivalence relation e. Then, it requires a polynomial time step for computing $\mathcal{I}_{\mathcal{K}}$, replacing all individuals c in $\mathcal{I}_{\mathcal{K}}$ and q with a new object denoting its equivalence class in e, and check whether there is no object o for which $ineq(o, o) \in \mathcal{I}_{\mathcal{K}}$. Finally, with a single call to an coNP-oracle, the algorithm checks whether the resulting $\mathcal{I}_{\mathcal{K}}$ and q are such that $\mathcal{I}_{\mathcal{K}} \not\models \delta(q)$. Observe that this last check can be done in AC^0 in the size of $\mathcal{A}$. □

We now provide matching lower bounds for both data and combined complexity, showing that they hold already for the case of CQ$^{\neq}$s. We start with data complexity.

Theorem 4.10. *Answering CQ$^{\neq}$s over $DL\text{-}Lite^{\neg}_{\mathsf{RDFS}}$ knowledge bases is coNP-hard in data complexity.*

Proof. The proof is by a LogSpace reduction from the data complexity version of the problem of answering CQ$^{\neq}$s over a *data exchange setting* [Fagin *et al.*, 2005a], known to be coNP-hard (in fact, coNP-complete) already for boolean CQ$^{2,\neq}$s [Madry, 2005]. Since the target schema T of the problem considered in [Madry, 2005] uses only binary predicates and has no *target constraints* (i.e., T has no assertions), and since it uses a set of LAV source-to-target dependencies, such problem can be reformulated in the OBDM scenario as follows: there exists an OBDM specification $\Sigma = \langle T, \mathcal{S}, \mathcal{M} \rangle$ with $T = \emptyset$ and $\mathcal{M}$ being a LAV mapping, and a boolean CQ$^{2,\neq}$ q over T such that, given an $\mathcal{S}$-database D, checking whether $cert^{D}_{q,\Sigma}$ is true (i.e., $cert^{D}_{q,\Sigma} = \{()\}$) when the UNA is adopted is in general a coNP-hard problem.

Let D be an $\mathcal{S}$-database. We now construct a $DL\text{-}Lite^{\neg}_{\mathsf{RDFS}}$ knowledge base $\mathcal{K}_D = \langle \mathcal{O}, \mathcal{A}_D \rangle$, where $\mathcal{O} = \{P_1 \sqsubseteq \neg P_2\}$ with the alphabet being composed of all the binary predicates (equivalently, atomic roles) of the target schema T plus the two fresh atomic roles P_1 and P_2. Note that the above CQ$^{2,\neq}$ q is also a query over $\mathcal{O}$.

In order to construct $\mathcal{A}_D$, we consider $\mathcal{M}(D)$, i.e., the chase of D with respect to $\mathcal{M}$. Specifically, from $\mathcal{M}(D)$ we obtain a set of ABox assertions by replacing each variable $v \in \mathsf{Var}$ occurring in $\mathcal{M}(D)$ with a different fresh individual denoted by c_v. Furthermore, for each pair of constants c_1, c_2 occurring in D with $c_1 \neq c_2$, $\mathcal{A}_D$ contains the ABox assertions $P_1(\alpha_{c_1,c_2}, c_1)$ and $P_2(\alpha_{c_1,c_2}, c_2)$, where α_{c_1,c_2} is a fresh individual. Intuitively, such assertions simulates the UNA but only on the individuals that were in D.

Observe that both $\mathcal{O}$ and q do not depend on D, which is the only input of the (data complexity version of the) problem we are reducing from. Finally, since $\mathcal{M}$ is fixed, the ABox $\mathcal{A}_D$ can be constructed in LogSpace from D [Arenas *et al.*, 2010].

Let D be any $\mathcal{S}$-database. We now prove that $\mathcal{K}_D \models q$ (when the UNA is not adopted) if and only if $cert^D_{q,\Sigma} = \{()\}$ (when the UNA is adopted).

"**If part:**" Suppose that $\mathcal{K}_D \not\models q$, i.e., there is a model $\mathcal{I}$ of $\mathcal{K}_D$ for which $\mathcal{I} \not\models q$. By construction, however, since $\mathcal{I}$ is a model of $\mathcal{K}_D$, we have $c_1^{\mathcal{I}} \neq c_2^{\mathcal{I}}$ for each pair of individuals c_1, c_2 occurring in D with $c_1 \neq c_2$. But then, from $\mathcal{I}$ it is straightforward to obtain a model $\mathcal{I}' \in Mod_D(\Sigma)$ satisfying the UNA for which $\mathcal{I}' \not\models q$. Thus, $cert^D_{q,\Sigma} = \emptyset$, as required.

"**Only-if part:**" Suppose that $cert^D_{q,\Sigma} = \emptyset$ when the UNA is adopted. It follows that there is a model $\mathcal{I}' = \langle \Delta^{\mathcal{I}'}, \cdot^{\mathcal{I}'} \rangle$ of Σ relative to D that satisfies the UNA for which $\mathcal{I}' \not\models q$. By [Fagin *et al.*, 2005a, Theorem 3.3], there exists a function ψ from $\mathsf{dom}(\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)})$ to $\Delta^{\mathcal{I}'}$ satisfying conditions 1 and 2 of Proposition 4.1 such that $\psi(c) = c^{\mathcal{I}'}$ for each constant $c \in \mathsf{dom}(D)$.

Consider the interpretation $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ for $\mathcal{K}_D$, where $\Delta^{\mathcal{I}} = \Delta^{\mathcal{I}'} \cup \{\alpha_{c_1,c_2} \mid c_1, c_2 \in \mathsf{dom}(D) \wedge c_1 \neq c_2\}$, and $\cdot^{\mathcal{I}}$ extends $\cdot^{\mathcal{I}'}$ by (i) assigning to each individual c_v of $\mathcal{A}_D$ that replace a variable v in $\mathcal{M}(D)$ the image of v under ψ, i.e., $c_v^{\mathcal{I}} = \psi(v)$, (ii) assigning $\alpha^{\mathcal{I}}_{c_1,c_2} = \alpha_{c_1,c_2}$ to each introduced individual α_{c_1,c_2} in $\mathcal{A}_D$, and (iii) for both $i = 1$ and $i = 2$, assigning to atomic role P_i the set $P_i^{\mathcal{I}} = \{(\alpha, c) \mid P_i(\alpha, c) \in \mathcal{A}_D\}$.

Since $\mathcal{I}' \in Mod_D(\Sigma)$ and satisfies the UNA (i.e., $c_1^{\mathcal{I}} \neq c_2^{\mathcal{I}}$ for each pair of individuals c_1, c_2 occurring in D with $c_1 \neq c_2$) by assumption, we derive that $\mathcal{I}$ is a model of $\mathcal{K}_D$. Furthermore, since $\mathcal{I}' \not\models q$, we have that $\mathcal{I} \not\models q$ as well. It follows that $\mathcal{K}_D \not\models q$, as required.

Finally, we point out that the same result holds even for the problem of answering UCQ$^{\neq}$s over *DL-Lite*$_{\mathsf{RDFS}}$ knowledge bases (in particular, even when ontologies have no assertions). In particular, observe that it is sufficient to apply the following changes to the above reduction: (i) ontology $\mathcal{O}'$ has the same alphabet of $\mathcal{O}$ but has no assertions, i.e., $\mathcal{O}' = \emptyset$, and (ii) the query issued over the ontology $\mathcal{O}'$ is the fixed UCQ $q' = q \cup q_P$, where q is the CQ$^{2,\neq}$ as in the above reduction and q_P is the CQ $\{() \mid \exists y_1, y_2 . P_1(y_1, y_2) \wedge P_2(y_1, y_2)\}$, which intuitively asks whether two constants c_1, c_2 occurring in D with $c_1 \neq c_2$ are interpreted as the same domain object.

Let D be any $\mathcal{S}$-database. Using similar arguments as the ones given above, it is easy to see that $\mathcal{K}'_D \models q'$ (when the UNA is not adopted) if and only if $cert^D_{q,\Sigma} = \{()\}$ (when the UNA is adopted), where $\mathcal{K}'_D$ is the *DL-Lite*$_{\mathsf{RDFS}}$ knowledge base $\mathcal{K}'_D = \langle \mathcal{O}', \mathcal{A}_D \rangle$. □

The proof of the above theorem has two interesting implications: (i) coNP-hardness in data complexity holds even for CQ$^{2,\neq}$s; (ii) Answering UCQ$^{2,\neq}$s (in particular, the union of a CQ$^{2,\neq}$ and of a CQ) over knowledge bases with empty ontologies is coNP-hard in data complexity, too.

Interestingly, implication (ii) corrects an erroneous statement in [Rosati, 2007, Theorem 11], where it is claimed that answering UCQ$^{\neq}$s over *DL-Lite*$_{\mathsf{RDFS}}$ knowledge bases is in LogSpace in data complexity regardless of whether the UNA is adopted or not. It turns out that, unless LogSpace = NP, this latter statement is true only when the UNA is adopted (cf. beginning of this chapter).

We now provide the matching lower bound for combined complexity.

Theorem 4.11. *Answering $CQ^{\neq}$s over DL-Lite$^{\neg}_{\mathsf{RDFS}}$ knowledge bases is Π^p_2-hard in combined complexity.*

Proof. The proof is by a LogSpace reduction from the containment problem for conjunctive queries with inequalities, known to be Π^p_2-hard (in fact, Π^p_2-complete) even when restricted to boolean $CQ^{\neq}$s over a schema with all predicates having arity at most two [Kolaitis *et al.*, 1998]. The containment problem for conjunctive queries with inequalities is the problem of deciding, given two $CQ^{\neq}$s q_1, q_2 over the same schema $\mathcal{S}$, whether $q_1 \sqsubseteq q_2$, i.e., whether $q_1^D \subseteq q_2^D$ for each $\mathcal{S}$-database D.

Let q_1, q_2 be two boolean $CQ^{\neq}$s over the same database schema $\mathcal{S}$ with all the predicates of $\mathcal{S}$ having arity at most 2. We define a *DL-Lite*$^{\neg}_{\mathsf{RDFS}}$ knowledge base $\mathcal{K}_{q_1} = \langle \mathcal{O}, \mathcal{A}_{q_1} \rangle$ where $\mathcal{O} = \{P_1 \sqsubseteq \neg P_2\}$ and the alphabet of $\mathcal{O}$ is composed of all the binary predicates (equivalently, atomic roles) and unary predicates (equivalently, atomic concepts) of the schema $\mathcal{S}$, plus the two fresh atomic roles P_1 and P_2.

To construct $\mathcal{A}_{q_1}$, we consider the freezing of q_1, i.e., the set of facts obtained from the body of q_1 by replacing each variable in v with a fresh individual c_v.

Specifically, $\mathcal{A}_{q_1}$ is such that (i) every non-inequality atom occurring in the body of q_1 becomes an ABox assertion of $\mathcal{A}$ (after replacing variables v with its corresponding individual c_v); (ii) for each inequality atom $z_1 \neq z_2$ occurring in q_1, we have the ABox assertions $P_1(\alpha_{z_1,z_2}, z'_1)$ and $P_2(\alpha_{z_1,z_2}, z'_2)$, where α_{z_1,z_2} is a fresh individual, and for both $i = 1$ and $i = 2$, if z_i is a variable v, then $z'_i = c_v$, otherwise (i.e., z_i is a constant), $z'_i = z_i$; and finally, (iii) for each pair c_1, c_2 of constants occurring in q_1 with $c_1 \neq c_2$, we have the ABox assertions $P_1(\alpha_{c_1,c_2}, c_1)$ and $P_2(\alpha_{c_1,c_2}, c_2)$, where α_{z_1,z_2} is a fresh individual. Intuitively, such assertions simulates the UNA on the individuals that were in q_1 and on the terms $t_1 \neq t_2$ that were inequality atoms in (the freezing of) q_1.

We now prove that $q_1 \sqsubseteq q_2$ if and only if $\mathcal{K}_{q_1} \models q_2$.

"**If part:**" Suppose that $q_1 \not\sqsubseteq q_2$, i.e., there exists an $\mathcal{S}$-database D for which $D \models q_1$ and $D \not\models q_2$. Consider the homomorphism h from q_1 to D (at least one exists because $D \models q_1$), and let $\mathcal{I}$ be an interpretation for $\mathcal{K}_{q_1}$ such that (i) $\Delta^{\mathcal{I}} = \mathsf{dom}(D) \cup \{\alpha \mid \text{for each introduced individual } \alpha\}$; (ii) $c^{\mathcal{I}} = c$ for each constant c occurring in q_1, and $\alpha^{\mathcal{I}} = \alpha$ for each introduced individual α; (iii) $c_v^{\mathcal{I}} = h(v)$ for each variable v occurring in q_1; (iv) for both $i = 1$ and $i = 2$, $P_i^{\mathcal{I}} = \{(\alpha, c) \mid P_i(\alpha, c) \in \mathcal{A}_{q_1}\}$; and (v) the extension in $\mathcal{I}$ of each atomic role and atomic concept of $\mathcal{O}$ corresponding to a relation in $\mathcal{S}$ is the same as the corresponding relation in D.

Since $h(z_1) \neq h(z_2)$ for each inequality atom $z_1 \neq z_2$ of q_1, we have that $\mathcal{I}$ is a model of $\mathcal{K}_{q_1}$ by construction. Moreover, since $D \not\models q_2$, we derive $\mathcal{I} \not\models q_2$ as well. But then, $\mathcal{I}$ is a model of $\mathcal{K}_{q_1}$ for which $\mathcal{I} \not\models q_2$. It follows that $\mathcal{K}_{q_1} \not\models q_2$, as required.

"**Only-if part:**" Suppose that $\mathcal{K}_{q_1} \not\models q_2$, i.e., there exists a model $\mathcal{I}$ of $\mathcal{K}_{q_1}$ for which $\mathcal{I} \not\models q_2$. Since $\mathcal{I}$ is a model of $\mathcal{K}_{q_1}$, by construction we have that $z_1'^{\mathcal{I}} \neq z_2'^{\mathcal{I}}$ for each inequality atom $z_1 \neq z_2$ occurring in q_1, where for both $i = 1$ and $i = 2$ if z_i is a variable v, then $z'_i = c_v$, otherwise (i.e., z_i is a constant), $z'_i = z_i$.

Consider now the $\mathcal{S}$-database D similar to $\mathcal{I}$ where each unary (resp., binary) predicate in D has the same extension of the corresponding atomic concept (resp., role) in $\mathcal{I}$, but, for every constant c of q_1, every occurrence of the object $c^{\mathcal{I}}$ is replaced with the constant c in D (observe that, since $\mathcal{I}$ is a model of $\mathcal{K}_{q_1}$, $c_1^{\mathcal{I}} \neq c_2^{\mathcal{I}}$ for each pair c_1, c_2 of constants occurring in q_1 with $c_1 \neq c_2$). Obviously, since $\mathcal{I} \not\models q_2$, we have that $D \not\models q_2$ as well.

Furthermore, let h be a function from variables and constants of q_1 to constants of D such that (i) $h(c) = c$ for each constant c of q_1; and (ii) $h(v) = c_v^{\mathcal{I}}$ for each variable v of q_1. It is straightforward to verify that h is a homomorphism from q_1 to D. But then, the $\mathcal{S}$-database D is such that $D \models q_1$ and $D \not\models q_2$. It follows that $q_1 \not\sqsubseteq q_2$, as required. □

Corollary 4.3. *Answering UCQ$^{\neq}$s over DL-Lite$_{\mathsf{RDFS}}^{\neg}$ knowledge bases is coNP-complete in data complexity and Π_2^p-complete in combined complexity.*

By looking at the proof of the above theorem, one can see that the output of the reduction produces a CQ$^{\neq}$ whose number of inequalities is equal to the number of inequalities of the input query q_2, and therefore is not fixed a priori.

To the best of our knowledge, it is not known whether checking $q_1 \sqsubseteq q_2$ is Π_2^p-hard even if q_2 uses a fixed number of inequalities. More generally, it is thus natural to ask which is the minimum number of inequalities in CQ$^{\neq}$s that makes the problem Π_2^p-hard in combined complexity. Similarly to the case of the coNP-hardness result in data complexity, we conjecture that such number is two.

Conjecture 4.1. *Answering CQ$^{2,\neq}$s over DL-Lite$_{\mathsf{RDFS}}^{\neg}$ knowledge bases is Π_2^p-hard in combined complexity.*

Even though we have not been able to prove this conjecture, interestingly, the proof of Theorem 4.8 shows that Π_2^p-hardness holds for UCQ$^{2,\neq,b}$s (i.e., UCQ$^{2,\neq}$s with bounded inequalities) over knowledge bases with empty ontologies. Notice, however, that the illustrated UCQ$^{2,\neq,b}$ does not have a fixed number of disjuncts.

Actually, with a slight adaptation of such proof, we now prove that the same result holds even for UCQ$^{2,\neq}$s that are the union of a fixed (but not bounded) CQ$^{2,\neq}$ and a CQ without inequalities.

Theorem 4.12. *Answering UCQ$^{2,\neq}$s over knowledge bases with empty ontologies is Π_2^p-hard in data complexity.*

Proof. Consider the following changes to the reduction illustrated in the proof of Theorem 4.8:

- The ontology $\mathcal{O}$ contains the additional atomic concept B in its alphabet;
- For each $i \in [1, m]$, the ABox $\mathcal{A}_F$ additionally contains the ABox assertion $B(x_i)$ (recall that x_i is the individual corresponding to the universally quantified variable x_i);
- The boolean UCQ$^{2,\neq}$ Q_F is the union of a fixed CQ$^{2,\neq}$ q and of q_F, where:
  - $q = \{() \mid \exists y.B(y) \wedge y \neq 0 \wedge y \neq 1\}$;

– q_F is the boolean version of the CQ defined in the reduction of the proof of Theorem 4.8:

$$\{() \mid \exists a_1, \dots, a_p, y_1, \dots, y_n. \bigwedge_{i\in[1,p]} (P_{i,1}(a_i, z_{i,1}) \wedge P_{i,2}(a_i, z_{i,2}) \wedge P_{i,3}(a_i, z_{i,3})) \wedge \bigwedge_{i\in[1,m]} (H_i(x_i))\}.$$

Intuitively, through the possibility of involving existential variables in inequality atoms, the $\mathrm{CQ}^{2,\neq}$ q encases the behaviour of the $\mathrm{UCQ}^{\neq,b}$ $q_1 \cup \ldots \cup q_m$ that asks whether some individual x_i (corresponding to the universally quantified variable x_i of formula F) in a possible interpretation $\mathcal{I}$ is such that $x_i^{\mathcal{I}} \neq 0^{\mathcal{I}}$ and $x_i^{\mathcal{I}} \neq 1^{\mathcal{I}}$.

Let F be any instance of the of the $\forall\exists$-CNF problem. Using analogous arguments to the ones provided in the proof of Theorem 4.8, it is easy to see that F is true if and only if $\mathcal{K}_F \models Q_F$, where $\mathcal{K}_F = \langle \mathcal{O}, \mathcal{A}_F \rangle$ and Q_F are the knowledge base and the query, respectively, obtained by modifying the reduction illustrated in the proof of Theorem 4.8 as explained above. □

Observe that the hardness results of Theorems 4.10 and 4.11 do not hold if we replace *DL-Lite*$^{\neg}_{\mathsf{RDFS}}$ with *DL-Lite*$_{\mathsf{RDFS}}$. This is due to the inability of the *DL-Lite*$_{\mathsf{RDFS}}$ knowledge base language to express inconsistencies. Specifically, given a $\mathrm{CQ}^{\neq}$ q over a *DL-Lite*$_{\mathsf{RDFS}}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$, either q contains an inequality atom, and therefore $\mathcal{K} \not\models q$ trivially holds, or q is a CQ. This clearly proves that answering $\mathrm{CQ}^{\neq}$s over *DL-Lite*$_{\mathsf{RDFS}}$ knowledge bases has the same computational complexity of answering CQs over the same language (cf. beginning of this chapter).

Theorem 4.13. *Answering $\mathrm{CQ}^{\neq}$s over DL-Lite$_{\mathsf{RDFS}}$ knowledge bases is in AC^0 in data complexity and NP-complete in combined complexity.*

As already noticed, if we move to consider $\mathrm{UCQ}^{\neq}$s over *DL-Lite*$_{\mathsf{RDFS}}$ knowledge bases, the adaptation illustrated in the reduction of the proof of Theorem 4.10 and Theorem 4.12 show a jump from AC^0 to coNP for data complexity and from NP to Π_2^p in combined complexity, respectively.

Theorem 4.14. *Answering $\mathrm{UCQ}^{\neq}$s over DL-Lite$_{\mathsf{RDFS}}$ knowledge bases is coNP-complete in data complexity and Π_2^p-complete in combined complexity. Both the hardness results already hold for $\mathrm{UCQ}^{2,\neq}$s over knowledge bases with empty ontologies.*

To fully complete the picture of the problem of answering $\mathrm{UCQ}^{\neq}$s over *DL-Lite*$^{\neg}_{\mathsf{RDFS}}$ knowledge bases, it remains to study the case of $\mathrm{UCQ}^{1,\neq}$s, i.e., UCQs having at most one inequality per disjunct. We do so in the next section.

4.5 Answering $\mathrm{UCQ}^{1,\neq}$s over *DL-Lite*$^{\neg}_{\mathsf{RDFS}}$ knowledge bases

In what follows, without loss of generality, we assume that each $\mathrm{UCQ}^{1,\neq}$ q is written as $q = q_1 \cup q_2$, where q_2 is a UCQ with no inequalities and q_1 is a $\mathrm{UCQ}^{1,\neq}$ having exactly one inequality per disjunct.

In principle, for answering $\mathrm{UCQ}^{1,\neq}$s over *DL-Lite*$^{\neg}_{\mathsf{RDFS}}$ knowledge bases, it is possible to use the algorithm provided in [Fagin *et al.*, 2005a, Theorem 5.12] in the context of data exchange. Notice, however, that the running time of this algorithm would be polynomial in the size of the ABox but exponential in the size of the query. On the contrary, by elaborating on the idea of [Fagin *et al.*, 2005a, Theorem 5.12], we are able to prove that the problem is in PTIME in data complexity and in NP in combined complexity. We start with the following definition and algorithm.

Definition 4.3. Let $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ be a $DL\text{-}Lite^{\neg}_{\mathsf{RDFS}}$ knowledge base, q be a boolean UCQ$^{1,\neq}$ over $\mathcal{O}$, and $F = [f_1, \ldots, f_m]$ be a list of functions from the set of variables and individuals occurring in q to the set of individuals occurring in $\mathcal{A}$. We say that F is *a good sequence with respect to* $\mathcal{K}$ *and* q if $\mathsf{CheckGood}(\mathcal{K}, q, F)$ returns `true`.

Algorithm 4.3 CheckGood

Input:

$DL\text{-}Lite^{\neg}_{\mathsf{RDFS}}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$;
boolean UCQ$^{1,\neq}$ $q = q_1 \cup q_2$ over $\mathcal{O}$, where q_1 is a UCQ$^{1,\neq}$ having exactly one inequality per disjunct and q_2 is a UCQ;
list of functions $F = [f_1, \ldots, f_m]$

Output:

`true` or `false`

Compute $D := \mathcal{I}_{\mathcal{K}}$
for $i \leftarrow 1$ to $m - 1$ **do**
    if f_i is a homomorphism from some disjunct of q_1 to D **then**
        Let $z_1 \neq z_2$ be the inequality atom of such disjunct
        if $ineq(f_i(z_1), f_i(z_2)) \in D$ **then**
            return `true`
        else
            replace each occurrence of individual $f_i(z_1)$ in D and in q with $f_i(z_2)$
        end if
    else
        return `false`
    end if
end for
if f_m is a homomorphism from some disjunct of q_2 to D **then**
    return `true`
else
    return `false`
end if

Roughly speaking, starting from a set of facts $D := \mathcal{I}_{\mathcal{K}}$ over $\mathcal{O}$ (notice that $\mathcal{I}_{\mathcal{K}}$ does not introduce variables), in each step i from 1 to $m-1$ such that f_i is a homomorphism from a disjunct q' of q_1 to D, the algorithm $\mathsf{CheckGood}$ replaces everywhere the individual $f_i(z_1)$ with the individual $f_i(z_2)$, to consider the models in which $f_i(z_1) = f_i(z_2)$, where $z_1 \neq z_2$ is the only inequality atom occurring in the body of q'. Indeed, since f_i is a homomorphism from q' to D, q_1 is true in those models where $f_i(z_1) \neq f_i(z_2)$.

Afterwards, the algorithm sanctions that F is a good sequence if and only if either it is not possible to equate two individuals without contradicting an *ineq* fact of D, or the resulting D and q_2 are such that $D \models q_2$.

Using the above notion of good sequence, it is possible to derive the following characterisation ($n_{\mathcal{A}}$ denotes the number of individuals occurring in the ABox $\mathcal{A}$).

Proposition 4.8. *Let $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ be a DL-Lite$^{\neg}_{\mathsf{RDFS}}$ knowledge base, and let q be a boolean UCQ$^{1,\neq}$ over $\mathcal{O}$. We have that $\mathcal{K} \models q$ if and only if there exists a list $F = [f_1, \ldots, f_m]$ of functions, with $m \leq n_{\mathcal{A}}$, such that F is a good sequence with respect to $\mathcal{K}$ and q.*

Proof. "**If part:**" Suppose $F = [f_1, \ldots, f_m]$ (with $m \leq n_{\mathcal{A}}$) is a good sequence with respect to $\mathcal{K}$ and q, i.e., $\mathsf{CheckGood}(\mathcal{K}, q, F)$ returns true. It follows that, after possibly applying $l \leq m - 1$ equalities between individuals on D and q (where D starts from $\mathcal{I}_{\mathcal{K}}$), either $l \leq m - 2$ and f_{l+1} is a homomorphism from a disjunct q' of q_1 to the resulting set of facts D such that $ineq(f_{l+1}(z_1), f_{l+1}(z_2)) \in D$ (where $z_1 \neq z_2$ is the only inequality atom in the body of q'), or $l = m - 1$ and f_m is a homomorphism from a disjunct of q_2 to D.

In both cases, consider each homomorphism f_i from some of the disjuncts of q_1 to D, for $i = [1, l]$. It is easy to see that all models $\mathcal{I}$ of $\mathcal{K}$ in which $f_i(z_1)^{\mathcal{I}} \neq f_i(z_2)^{\mathcal{I}}$ (where $z_1 \neq z_2$ is the inequality atom of the disjunct q' of q_1 for which f_i is a homomorphism to D) is such that $\mathcal{I} \models q_1$, and therefore $\mathcal{I} \models q$. So, at each iteration the algorithm equates $f_i(z_1)$ and $f_i(z_2)$ in D to consider all the other possible models in which $f_i(z_1)^{\mathcal{I}} = f_i(z_2)^{\mathcal{I}}$.

If f_{l+1} is a homomorphism from a disjunct q' of q_1 to the resulting set of facts D such that $ineq(f_{l+1}(z_1), f_{l+1}(z_2)) \in D$ (where $z_1 \neq z_2$ is the only inequality atom in the body of q'), due to Proposition 4.1, we derive that there is no models $\mathcal{I}$ in which $f_i(z_1)^{\mathcal{I}} = f_i(z_2)^{\mathcal{I}}$. It trivially follows that $\mathcal{K} \models q_1$, and therefore $\mathcal{O} \models q$.

Finally, in the case that f_m is a homomorphism from a disjunct of q_2 to the resulting set of facts D, due to the above considerations, it can be easily proven that each model $\mathcal{I}$ of $\mathcal{K}$ is such that either $\mathcal{I} \models q_1$, or $\mathcal{I} \models q_2$. Thus, $\mathcal{K} \models q$, as required.

"**Only-if part:**" Suppose there is no such good sequence F with respect to $\mathcal{K}$ and q, and consider the set of facts obtained in the following way starting from $D := \mathcal{I}_{\mathcal{K}}$: for each possible homomorphism h from a disjunct of q_1 to D, replace each occurrence of $h(z_1)$ in D and in q with $h(z_2)$, where $z_1 \neq z_2$ is the inequality atom of the disjunct of q_1 for which h is a homomorphism to D.

Since there are $n_{\mathcal{A}}$ individuals in the ABox $\mathcal{A}$, and so also in D, there can be at most $n_{\mathcal{A}} - 1$ of such homomorphisms. Indeed, after applying $n_{\mathcal{A}} - 1$ replacing as described above, the resulting set of facts D would contain only one individual.

For the resulting set of facts D and query q observe that (i) it is never the case that $ineq(h(z_1), h(z_2)) \in D$ for some homomorphism h; and (ii) there is no disjunct of q_2 for which there is a homomorphism to D. In proof, if either point (i) or point (ii) is not true, then we would easily derive a contradiction on the fact that there exists no good sequence F with respect to $\mathcal{K}$ and q.

But then, by construction D is such that $D \not\models q_1$ and $D \not\models q_2$, and therefore $D \not\models q$. From the resulting D, moreover, it is immediate to construct a model $\mathcal{I}_D$ of $\mathcal{K}$ such that $\mathcal{I}_D \not\models q$. It follows that $\mathcal{K} \not\models q$, as required. $\square$

We are now ready to establish upper bound results for the problem of answering UCQ$^{1,\neq}$s over *DL-Lite*$^{\neg}_{\mathsf{RDFS}}$ knowledge bases.

Theorem 4.15. *Answering UCQ$^{1,\neq}$s over DL-Lite$^{\neg}_{\mathsf{RDFS}}$ knowledge bases is in* PTIME *in data complexity and in NP (and therefore NP-complete) in combined complexity.*

Proof. Due to Proposition 4.8, checking whether $\mathcal{K} \models q$ for a *DL-Lite*$^{\neg}_{\mathsf{RDFS}}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ and UCQ$^{1,\neq}$ q can be done as follows:

- Guess a list $F = [f_1, \ldots, f_m]$ of $m \leq n_{\mathcal{A}}$ functions from the set of variables and individuals occurring in q to the set of individuals occurring in $\mathcal{A}$;

- Check whether F is a good sequence with respect to $\mathcal{K}$ and q,

where checking whether F is a good sequence with respect to $\mathcal{K}$ and q can be done by means of the above described $\mathsf{CheckGood}$ algorithm.

So, we easily derive a nondeterministic algorithm deciding $\mathcal{K} \models q$ that first requires an NP step in order to guess the list $F = [f_1, \ldots, f_m]$ with $m \leq n_{\mathcal{A}}$ of functions. This can be done in polynomial time in the size of $\mathcal{A}$. Then, by exploiting the $\mathsf{CheckGood}$ algorithm, we check whether F is a good sequence with respect to $\mathcal{K}$ and q using: (i) a polynomial time step in the size of $\mathcal{K}$ for computing $D := \mathcal{I}_{\mathcal{K}}$; (ii) for each $i \in [1, m-1]$, a polynomial time step for checking whether f_i is a homomorphism from a disjunct of q_1 to D, $ineq(f_i(z_1), f_i(z_2)) \notin D$ (where $z_1 \neq z_2$ is the inequality atom of such a disjunct), and for replacing each occurrence of $f_i(z_1)$ with $f_i(z_2)$; finally, (iii) another polynomial time step for checking whether f_m is a homomorphism from some disjunct of the UCQ q_2 to the resulting D. $\square$

While NP-hardness in combined complexity trivially follows from CQ evaluation over relational databases, we now provide a matching lower bound for data complexity, showing that it holds already for the case of CQ$^{1,\neq}$s.

Theorem 4.16. *Answering CQ$^{1,\neq}$s over DL-Lite$^{\neg}_{\mathsf{RDFS}}$ knowledge bases is* PTIME*-hard in data complexity.*

Proof. The proof is by a LOGSPACE reduction from the entailment problem for HORN-3CNF, known to be PTIME-complete [Börger *et al.*, 1997]. Given a set of formulas $F = \{f_1, \ldots, f_m\}$ on a set of propositional variables $A = \{a_1, \ldots, a_n\}$ where f_i is either of the form $f_i = (a_j \wedge a_k \rightarrow a_h)$ or of the form $f_i = (\top \rightarrow a_h)$, for each $i = [1, n]$, and given a propositional variable $a_w \in A$, the entailment problem for HORN-3CNF is the problem of deciding whether $F \models a_w$.

Let H, P_1, and P_2 be three atomic roles, and let B_1 and B_2 be two atomic concepts. We define the following fixed *DL-Lite*$^{\neg}_{\mathsf{RDFS}}$ ontology $\mathcal{O}$ and boolean CQ$^{1,\neq}$ q over $\mathcal{O}$ with $\mathcal{O} = \{B_1 \sqsubseteq \neg B_2\}$ and $q = \{() \mid \exists y_1, y_2 . H(y_1, y_2) \wedge P_1(y_2, t) \wedge P_2(y_2, t) \wedge y_1 \neq t\}$, where t is an individual.

Given a HORN-3CNF formula F, we construct an ABox $\mathcal{A}_F$ as follows: (i) for each formula $f_i \in F$ of the form $f_i = (a_j \wedge a_k \rightarrow a_h)$, we include the ABox assertions $H(a_h, f_i)$, $P_1(f_i, a_j)$, and $P_2(f_i, a_k)$, where a_h, f_i, a_j, and a_k are individuals of $\mathcal{A}$; (ii) for each formula $f_i \in F$ of the form $f_i = (\top \rightarrow a_h)$, we include the ABox assertions $H(a_h, f_i)$, $P_1(f_i, t)$, and $P_2(f_i, t)$, where a_h, f_i and t are individuals of $\mathcal{A}$; and, (iii)

we include the ABox assertions $B_1(a_w)$ and $B_2(t)$, stating that individuals a_w and t have to be interpreted as different elements in each possible model of $\mathcal{K}_F = \langle \mathcal{O}, \mathcal{A}_F \rangle$.

Observe that $\mathcal{O}$ and q do not depend on the inputs of the entailment problem for HORN-3CNF, while $\mathcal{A}_F$ can be constructed in LogSpace from them.

We now prove that $\mathcal{K}_F \models q$ if and only if $F \models a_w$, where $\mathcal{K}_F$ is the $DL\text{-}Lite^{\neg}_{\mathsf{RDFS}}$ knowledge base $\mathcal{K}_F = \langle \mathcal{O}, \mathcal{A}_F \rangle$.

"**If part:**" Suppose that $F \models a_w$. Then, it is possible to derive a_w using the following inference rules:

- $F \models a_h$ for each formula of the form $(\top \to a_h)$ occurring in F;
- if $F \models a_j$, $F \models a_k$, and $(a_j \wedge a_k \to a_h)$ is a formula in F, then $F \models a_h$.

For each propositional variable $a_h \in A$ such that $F \models a_h$ we now show, by induction on the length of the derivation of a_h from F, that $a_h^{\mathcal{I}} = t^{\mathcal{I}}$ in each interpretation $\mathcal{I}$ satisfying the ABox assertions in $\mathcal{A}_F$ and such that $\mathcal{I} \not\models q$.

Base case (l=0): Let $f_i \in F$ be a formula of the form $f_i = (\top \to a_h)$. Then, in the ABox $\mathcal{A}_F$ there are the assertions $H(a_h, f_i)$, $P_1(f_i, t)$, and $P_2(f_i, t)$. It is easy to see that every interpretation $\mathcal{I}$ satisfying such assertions is such that either $\mathcal{I} \models q$, or $a_h^{\mathcal{I}} = t^{\mathcal{I}}$. It follows that, for each propositional variable a_h such that $(\top \to a_h) \in F$ and for each interpretation $\mathcal{I}$ satisfying the ABox assertions in $\mathcal{A}_F$ and such that $\mathcal{I} \not\models q$, we have $a_h^{\mathcal{I}} = t^{\mathcal{I}}$, as required.

Inductive step: Let $f_i \in F$ be a formula of the form $f_i = (a_j \wedge a_k \to a_h)$, where both propositional variables a_j and a_k are derived from F at length $\lambda \leq l - 1$. By the inductive hypothesis, we have that $a_j^{\mathcal{I}} = a_k^{\mathcal{I}} = t^{\mathcal{I}}$ in each interpretation $\mathcal{I}$ satisfying the ABox assertions in $\mathcal{A}_F$ and such that $\mathcal{I} \not\models q$. Since, however, $H(a_h, f_i)$, $P_1(f_i, a_j)$, and $P_2(f_i, a_k)$ are assertions occurring in $\mathcal{A}_F$, for such interpretations $\mathcal{I}$ we have $(a_h^{\mathcal{I}}, f_i^{\mathcal{I}}) \in H^{\mathcal{I}}$, $(f_i^{\mathcal{I}}, a_j^{\mathcal{I}} = t^{\mathcal{I}}) \in P_1^{\mathcal{I}}$, and $(f_i^{\mathcal{I}}, a_k^{\mathcal{I}} = t^{\mathcal{I}}) \in P_2^{\mathcal{I}}$. Thus, for each propositional variable a_h derived at length l and for each interpretation $\mathcal{I}$ satisfying the ABox assertions in $\mathcal{A}_F$ and such that $\mathcal{I} \not\models q$, we derive that $a_h^{\mathcal{I}} = t^{\mathcal{I}}$, as required.

Finally, due to the fact that $F \models a_w$ by assumption, we have that $a_w^{\mathcal{I}} = t^{\mathcal{I}}$ in each interpretation $\mathcal{I}$ satisfying the ABox assertions in $\mathcal{A}_F$ and such that $\mathcal{I} \not\models q$. But then, since both $B_1(a_w)$ and $B_2(t)$ are assertions in $\mathcal{A}_F$, those interpretations $\mathcal{I}$ do not satisfy the ontology assertion $B_1 \sqsubseteq \neg B_2$, and therefore they are not models of $\mathcal{K}_F$. Therefore, every model $\mathcal{I}$ of $\mathcal{K}_F$ is such that $\mathcal{I} \models q$. Thus, $\mathcal{K}_F \models q$, as required.

"**Only-if part:**" Suppose that $F \not\models a_w$. Let $V \subset A$ be the set of propositional variables $a_h \in A$ such that $F \models a_h$. Consider the interpretation $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ with $\Delta^{\mathcal{I}} = \{a_h \mid a_h \in A \setminus V\} \cup \{f_i \mid f_i \in F\} \cup \{t\}$, where (i) $t^{\mathcal{I}} = t$; (ii) $f_i^{\mathcal{I}} = f$, for each formula $f_i \in F$; (iii) $a_h^{\mathcal{I}} = t^{\mathcal{I}} = t$, for each $a_h \in V$; (iv) $a_h^{\mathcal{I}} = a_h$, for each $a_h \in A \setminus V$; (v) $H^{\mathcal{I}} = \{(a_h^{\mathcal{I}}, f_i) \mid f_i = (a_j \wedge a_k \to a_h) \in F\} \cup \{(a_h^{\mathcal{I}}, f_i) \mid f_i = (\top \to a_h) \in F\}$, $P_1^{\mathcal{I}} = \{(f_i, a_j^{\mathcal{I}}) \mid f_i = (a_j \wedge a_k \to a_h) \in F\} \cup \{(f_i, t) \mid f_i = (\top \to a_h) \in F\}$, and $P_2^{\mathcal{I}} = \{(f_i, a_k^{\mathcal{I}}) \mid f_i = (a_j \wedge a_k \to a_h) \in F\} \cup \{(f_i, t) \mid f_i = (\top \to a_h) \in F\}$; and (vi) $B_1^{\mathcal{I}} = \{a_w^{\mathcal{I}}\}$ and $B_2^{\mathcal{I}} = \{t^{\mathcal{I}} = t\}$.

It is straightforward to verify that $\mathcal{I} \not\models q$. Moreover, since $F \not\models a_w$, we have $a_w^{\mathcal{I}} \neq t^{\mathcal{I}} = t$, and therefore $\mathcal{I}$ is a model of $\mathcal{K}$. It follows that $\mathcal{K}_F \not\models q$, as required.

Finally, observe that the same proof works even by removing the ontology assertion $B_1 \sqsubseteq \neg B_2$ (i.e., with an empty ontology $\mathcal{O}$), but adding the fixed boolean disjunct $\{() \mid \exists y.B_1(y) \land B_2(y)\}$ to the query q, and therefore answering UCQ$^{1,\neq}$s over *DL-Lite*$_{\mathsf{RDFS}}$ ontologies is PTIME-hard in data complexity, too. $\square$

Corollary 4.4. *Answering UCQ$^{1,\neq}$s over DL-Lite$^{\neg}_{\mathsf{RDFS}}$ knowledge bases is* PTIME-*complete in data complexity and NP-complete in combined complexity.*

Again, from the proof of the above theorem we can derive interesting observations: (i) since the result holds even for ontologies having disjointness assertions only between concepts, the theorem strengthens the PTIME-hardness result of [Gutiérrez-Basulto *et al.*, 2015, Theorem 15] for the case of *DL-Lite*$_{core}$ knowledge bases when the UNA is not adopted; (ii) answering UCQ$^{1,\neq}$s over *DL-Lite*$_{\mathsf{RDFS}}$ knowledge bases is PTIME-hard in data complexity, too.

We conclude this section with an observation for *DL-Lite*$_{\mathsf{RDFS}}$ knowledge bases. If we move to consider UCQ$^{1,\neq}$s rather than CQ$^{\neq}$s, we only have a jump from AC^0 to PTIME for data complexity, while the combined complexity remains the same.

Theorem 4.17. *Answering UCQ$^{1,\neq}$s over DL-Lite$_{\mathsf{RDFS}}$ knowledge bases is* PTIME-*complete in data complexity and NP-complete in combined complexity. Both the hardness results already hold for knowledge bases with empty ontologies.*

4.6 Containment of UCQ$^{\neq}$s in Relational Databases

The results presented in Sections 4.4 and 4.5 have some interesting implications in the context of containment of UCQs with inequalities in relational databases [Klug, 1988; van der Meyden, 1997; Kolaitis *et al.*, 1998; Koutris *et al.*, 2017].

The containment problem for UCQ$^{\neq}$s has been shown to be in Π_2^p and conjectured to be Π_2^p-complete in [Klug, 1988]. Such conjecture has been then confirmed to be true in [van der Meyden, 1997]. Later on, [Kolaitis *et al.*, 1998] has studied the impact on the computational complexity of some syntactic and structural conditions for such problem. Specifically, the problem of checking whether $q' \sqsubseteq q$ remains Π_2^p-hard even when restricted to queries q' and q such that all database predicates have arity at most two and every database predicate occurs at most three times in the body of q'. Finally, note that the problem of checking whether $q' \sqsubseteq q$ is coNP-complete in the size of q', i.e., when the query q is assumed to be fixed.

To the best of our knowledge, however, it has never been investigated how the number of inequality atoms affects the computational complexity of this problem. We do so in the remaining of this section.

4.6.1 Lower Bounds

In order to derive some interesting lower bound results, we start by proving that the problem of checking whether $\mathcal{K} \models q$, for a *DL-Lite*$^{\neg}_{\mathsf{RDFS}}$ knowledge base and boolean UCQ$^{\neq}$ q over $\mathcal{O}$, can be polynomially reduced to the problem of checking whether $q'_{\mathcal{K}} \sqsubseteq q$, where $q'_{\mathcal{K}}$ is a CQ$^{\neq}$.

Proposition 4.9. *Let $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ be a DL-Lite$^{\neg}_{\mathsf{RDFS}}$ knowledge base, and let q be a boolean UCQ$^{\neq}$ over $\mathcal{O}$. The problem of checking whether $\mathcal{K} \models q$ is polynomially reducible to the problem of checking whether $q'_{\mathcal{K}} \sqsubseteq q$, where $q'_{\mathcal{K}}$ is a boolean CQ$^{\neq}$.*

Proof. Without loss of generality, we can assume that q contains no constants in the bodies of its disjuncts. If this is not the case, then for each constant c occurring in a disjunct q' of q, we add the assertion $A_c(c)$ to the ABox $\mathcal{A}$, the atom $A_c(y_c)$ in conjunction to the body of q', and replace each occurrence of c in the body of q' with y_c, where A_c and y_c are a fresh atomic concept and a fresh existential variable, respectively. It is straightforward to verify that query entailment is preserved with this polynomial time transformation.

Given a *DL-Lite*$^{\neg}_{\mathsf{RDFS}}$ knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$, we construct a CQ$^{\neq}$ $q'_{\mathcal{K}}$ by means of the following polynomial time steps:

1. We compute $\mathcal{I}_{\mathcal{K}}$;

2. We consider the set of atoms S obtained from $\mathcal{I}_{\mathcal{K}}$ in the following way: for every individual c occurring in $\mathcal{I}_{\mathcal{K}}$, we replace each occurrence of c in $\mathcal{I}_{\mathcal{K}}$ with a fresh existential variable v_c;

3. We replace each atom of the form $\mathit{ineq}(v_{c_1}, v_{c_2}) \in S$ with the inequality atom $v_{c_1} \neq v_{c_2}$;

4. We set $q'_{\mathcal{K}} := \{() \mid \exists \vec{y}.S_q(\vec{y})\}$, where $S_q(\vec{y})$ denotes the conjunction of all atoms occurring in the resulting set of atoms S.

We now prove that $q'_{\mathcal{K}} \sqsubseteq q$ if and only if $\mathcal{K} \models q$, thus showing the claim.

"**If part:**" Suppose that $q'_{\mathcal{K}} \not\sqsubseteq q$. Since both $q'_{\mathcal{K}}$ and q are queries over the schema $\mathcal{O}$, it follows that there is a set of facts D over $\mathcal{O}$ for which $D \not\models q$ and $D \models q'_{\mathcal{K}}$. Since $D \models q'_{\mathcal{K}}$, there is a homomorphism h from $q'_{\mathcal{K}}$ to D (and therefore $h(v_{c_1}) \neq h(v_{c_2})$ for each inequality atom $v_{c_1} \neq v_{c_2}$ occurring in the body of $q'_{\mathcal{K}}$). Consider the interpretation $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ for $\mathcal{K}$ with (i) $\Delta^{\mathcal{I}} = \mathsf{dom}(D)$, (ii) $c^{\mathcal{I}} = h(v_c)$, for each constant c occurring in $\mathcal{A}$, and finally (iii) the extension of each atomic atomic concept and atomic role is the same as in D.

Clearly, since $D \not\models q$ and $D \models q'_{\mathcal{K}}$, we have $\mathcal{I} \not\models q$ and $\mathcal{I} \models q'_{\mathcal{K}}$ as well. Moreover, since $h(v_{c_1}) \neq h(v_{c_2})$ for each inequality atom $v_{c_1} \neq v_{c_2}$ occurring in the body of $q'_{\mathcal{K}}$, we have $c_1^{\mathcal{I}} \neq c_2^{\mathcal{I}}$ for each atom of the form $\mathit{ineq}(c_1, c_2) \in \mathcal{I}_{\mathcal{K}}$. This, together with the fact that $\mathcal{I} \models q'_{\mathcal{K}}$, implies that $\mathcal{I}$ is a model of $\mathcal{K}$. Thus, $\mathcal{I}$ is a model of $\mathcal{K}$ such that $\mathcal{I} \not\models q$. It follows that $\mathcal{K} \not\models q$, as required.

"**Only-if part:**" Suppose that $\mathcal{K} \not\models q$, i.e., there is a model $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ of $\mathcal{K}$ for which $\mathcal{I} \not\models q$. Consider the function h from the variables occurring in $q'_{\mathcal{K}}$ to $\Delta^{\mathcal{I}}$ such that $h(v_c) = c^{\mathcal{I}}$, for each variable v_c of $q'_{\mathcal{K}}$ (note that v_c is the variable that has replaced the constant c of $\mathcal{A}$ in the step 2 of the reduction).

Since $\mathcal{I}$ is a model of $\mathcal{K}$, it is straightforward to verify that h consists in a homomorphism from $q'_{\mathcal{K}}$ to $\mathcal{I}$ (and therefore $h(v_{c_1}) \neq h(v_{c_2})$ for each inequality atom $v_{c_1} \neq v_{c_2}$ occurring in the body of $q'_{\mathcal{K}}$), and so $\mathcal{I} \models q'_{\mathcal{K}}$. From the facts that $\mathcal{I} \not\models q$ and $\mathcal{I} \models q'_{\mathcal{K}}$, we can easily obtain a set of facts $D_{\mathcal{I}}$ over $\mathcal{O}$ for which $D_{\mathcal{I}} \not\models q$ and $D_{\mathcal{I}} \models q'_{\mathcal{K}}$. It follows that $q'_{\mathcal{K}} \not\sqsubseteq q$, as required. □

Observe that (i) if $\mathcal{K}$ is a *DL-Lite*$_{\mathsf{RDFS}}$ knowledge base rather than a *DL-Lite*$^{\neg}_{\mathsf{RDFS}}$ knowledge base, then the query $q'_{\mathcal{K}}$ produced by the above illustrated reduction contains no inequality atoms because $\mathcal{I}_{\mathcal{K}}$ contains no atoms with *ineq* as predicate, and therefore $q'_{\mathcal{K}}$ is a boolean CQ rather than a boolean CQ$^{\neq}$; and (ii) if $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$

is a *DL-Lite*$^{\neg}_{\mathsf{RDFS}}$ knowledge base such that $\mathcal{O}$ contains no inclusion assertions, then the above illustrated reduction is in fact a LogSpace reduction. From these observations, we immediately derive the following two corollaries.

Corollary 4.5. *Let $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ be a DL-Lite$_{\mathsf{RDFS}}$ knowledge base, and let q be a boolean UCQ$^{\neq}$ over $\mathcal{O}$. The problem of checking whether $\mathcal{K} \models q$ is polynomially reducible to the problem of checking whether $q'_{\mathcal{K}} \sqsubseteq q$, where $q'_{\mathcal{K}}$ is a boolean CQ.*

Corollary 4.6. *Let $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ be a DL-Lite$^{\neg}_{\mathsf{RDFS}}$ knowledge base such that $\mathcal{O}$ does not have inclusion assertions, and let q be a boolean UCQ$^{\neq}$ over $\mathcal{O}$. The problem of checking whether $\mathcal{K} \models q$ is* LogSpace *reducible to the problem of checking whether $q'_{\mathcal{K}} \sqsubseteq q$, where $q'_{\mathcal{K}}$ is a boolean CQ$^{\neq}$.*

Using results of Section 4.4, we are now ready to derive interesting lower bounds. Let the containment problem for UCQ$^{\neq}$s be the following decision problem: given two UCQ$^{\neq}$s q', q over the same schema $\mathcal{S}$, check whether $q' \sqsubseteq q$.

Theorem 4.18. *The containment problem for UCQ$^{\neq}$s is Π^p_2-hard (and therefore Π^p_2-complete) already when (*i*) both the input queries q' and q are boolean and every database predicate have arity at most two, (*ii*) q' is a CQ$^{\neq}$ (respectively, CQ), and (*iii*) q is a CQ$^{\neq}$ (respectively, UCQ$^{2,\neq}$ which is the union of a fixed CQ$^{2,\neq}$ and of a CQ without inequalities).*

Proof. By looking at the proof of Theorem 4.11 (respectively, Theorem 4.12), one realises that checking whether $\mathcal{K} \models q$ for a given *DL-Lite*$^{\neg}_{\mathsf{RDFS}}$ (respectively, *DL-Lite*$_{\mathsf{RDFS}}$) knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ and CQ$^{\neq}$ (respectively, UCQ$^{2,\neq}$ which is the union of a fixed CQ$^{2,\neq}$ and of a given CQ without inequalities) q is in general Π^p_2-hard. Since due to Proposition 4.9 (respectively, Corollary 4.5) this problem is polynomially reducible to checking whether $q'_{\mathcal{K}} \sqsubseteq q$, where $q'_{\mathcal{K}}$ is a CQ$^{\neq}$ (respectively, CQ), then the claim trivially follows. $\square$

We also conjecture a stronger version of the above result, which turns out to be valid as soon as Conjecture 4.1 is verified.

Conjecture 4.2. *The containment problem for UCQ$^{\neq}$s is Π^p_2-hard (and therefore Π^p_2-complete) already when (*i*) both the input queries q' and q are boolean and every databases predicate have arity at most two, (*ii*) q' is a CQ$^{\neq}$, and (*iii*) q is a CQ$^{2,\neq}$.*

As for the complexity of the containment problem for UCQ$^{\neq}$s when the containing query q is assumed to be fixed, we derive the following lower bounds.

Theorem 4.19. *When the containing query q is assumed to be fixed, the containment problem for UCQ$^{\neq}$s is coNP-hard (and therefore coNP-complete) already when (*i*) both the input query q' and the fixed query q are boolean and every database predicate have arity at most two, (*ii*) q' is a CQ$^{\neq}$ (respectively, CQ), and (*iii*) q is a CQ$^{2,\neq}$ (respectively, UCQ$^{2,\neq}$ which is the union of a CQ$^{2,\neq}$ and of a CQ without inequalities).*

Proof. By looking at the proof of Theorem 4.10, one realises that there exists a CQ$^{2,\neq}$ (respectively, UCQ$^{2,\neq}$ which is the union of a CQ$^{2,\neq}$ and of a CQ without inequalities) q such that checking whether $\mathcal{K} \models q$ for a given *DL-Lite*$^{\neg}_{\mathsf{RDFS}}$ (respectively, *DL-Lite*$_{\mathsf{RDFS}}$) knowledge base $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ is in general coNP-hard. Since due to Proposition 4.9 (respectively, Corollary 4.5) this problem is polynomially reducible to checking whether $q'_{\mathcal{K}} \sqsubseteq q$, where $q'_{\mathcal{K}}$ is a CQ$^{\neq}$ (respectively, CQ), then the claim trivially follows. □

As for the case of the containment for queries having only one inequality per disjunct, we have the following lower bound.

Theorem 4.20. *When the containing query q is assumed to be fixed, the containment problem for UCQ$^{1,\neq}$s is* PTime*-hard already when the input query q' is a CQ$^{1,\neq}$ (respectively, CQ) and the fixed query q is a CQ$^{1,\neq}$ (respectively, UCQ$^{1,\neq}$ which is the union of a CQ$^{1,\neq}$ and of a CQ without inequalities).*

Proof. By looking at the proof of Theorem 4.16, one realises that there exists a CQ$^{1,\neq}$ (respectively, UCQ$^{1,\neq}$ which is the union of a CQ$^{1,\neq}$ and of a CQ without inequalities) q such that checking whether $\mathcal{K} \models q$ for a given *DL-Lite*$^{\neg}_{\mathsf{RDFS}}$ (respectively, *DL-Lite*$_{\mathsf{RDFS}}$) $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$ with $\mathcal{O}$ having no inclusion assertions is in general PTime-hard. Since due to Corollary 4.6 (respectively, the combination of Corollary 4.6 and Corollary 4.5) this problem is LogSpace reducible to checking whether $q'_{\mathcal{K}} \sqsubseteq q$, where $q'_{\mathcal{K}}$ is a CQ$^{1,\neq}$ (respectively, CQ), then the claim trivially follows. □

4.6.2 Upper Bounds

By exploiting again the close connection between answering UCQ$^{\neq}$s over *DL-Lite*$^{\neg}_{\mathsf{RDFS}}$ knowledge bases and the containment problem for UCQ$^{\neq}$s, we can prove new upper bound complexity results for the containment problem for UCQ$^{1,\neq}$s. We follow a presentation path very similar to the one of Section 4.5, and start with the following definition and CheckFContains algorithm.

Definition 4.4. Let q' be a CQ$^{\neq}$, q be a UCQ$^{1,\neq}$, and $F = [f_1, \ldots, f_m]$ be a list of functions from the set of variables and constants occurring in q to the set of variables and constants occurring in q'. We say that F is *a good sequence with respect to q' and q* if CheckFContains(q', q, F) returns `true`.

In the algorithm, for a CQ$^{\neq}$ q' and two terms $f_i(z_1)$ and $f_i(z_2)$ of q', the query ${q'}^{f_i(z_2)}_{f_i(z_1)}$ denotes the CQ$^{\neq}$ obtained from q' in the following way: if one among $f_i(z_1)$ and $f_i(z_2)$ is a constant, then the variable is replaced everywhere in q' by the constant; if both are variables, then one is replaced everywhere in q' by the other.

Roughly speaking, consider each step i from 1 to $m-1$ such that f_i is a homomorphism from a disjunct q'' of q. Clearly, if conditions of step 3 are satisfied, then we trivially have that $q' \sqsubseteq q''$ and the algorithm CheckFContains returns `true`. If not, the algorithm replaces everywhere in q' one of the two terms among $f_i(z_1)$ and $f_i(z_2)$ with the other, to consider a "representative database" of q' in which $f_i(z_1) = f_i(z_2)$, where $z_1 \neq z_2$ is the inequality atom occurring in the body of q''.

Thus, the algorithm returns `true` if and only if either it is not possible to equate two terms without contradicting an inequality atom of q' (or because they are both constants), or the resulting q' is such that $q' \sqsubseteq q_2$.

Using the above notion of good sequence, it is possible to derive the following characterisation ($n_{q'}$ denotes the number of terms occurring in the query q').

Algorithm 4.4 CheckFContains

Input:

CQ$^{\neq}$ $q' = \{\vec{t'} \mid \exists \vec{y'}.\phi'(\vec{x'}, \vec{y'})\}$

UCQ$^{1,\neq}$ $q = q_1 \cup q_2$, where q_1 is a UCQ$^{1,\neq}$ having exactly one inequality per disjunct and q_2 is a UCQ;

list of functions $F = [f_1, \dots, f_m]$

Output:

`true` or `false`

1: **for** $i \leftarrow 1$ to $m - 1$ **do**
2:     **if** f_i is a homomorphism from some disjunct of q_1 to q' **then**
3:         **if** $f_i(z_1) \neq f_i(z_2) \in \phi'(\vec{x'}, \vec{y'})$ **or** $f_i(z_1)$ and $f_i(z_2)$ are both constants **then**
4:             **return** `true`
5:         **else**
6:             $q' := {q'}^{f_i(z_2)}_{f_i(z_1)}$
7:         **end if**
8:     **else**
9:         **return** `false`
10:     **end if**
11: **end for**
12: **if** f_m is a homomorphism from some disjunct of q_2 to the resulting q' **then**
13:     **return** `true`
14: **else**
15:     **return** `false`
16: **end if**

Proposition 4.10. *Let q' and q be a CQ$^{\neq}$ and a UCQ$^{1,\neq}$, respectively. We have that $q' \sqsubseteq q$ if and only if there exists a list $F = [f_1, \dots, f_m]$ of functions, with $m \leq n_{q'}$, such that F is* a good sequence with respect to q' and q.

Proof. "**If part:**" Suppose $F = [f_1, \dots, f_m]$ (with $m \leq n_{q'}$) is a good sequence with respect to q' and q, i.e., CheckFContains(q', q, F) returns true. It follows that, after possibly applying $l \leq m - 1$ equalities between terms on q', either $l \leq m - 2$ and f_{l+1} is a homomorphism satisfying conditions of steps 2 and 3 of the algorithm, or $l = m - 1$ and f_m is a homomorphism from some disjunct of q_2 to q'.

In both cases, consider each homomorphism f_i from some of the disjuncts of q_1 to q', for $i = [1, l]$. It is easy to see that all databases D and tuples of constants $\vec{c}$ for which there is a homomorphism h from q' to D with $h(\vec{t'}) = \vec{c}$ (thus $\vec{c} \in {q'}^D$) and in addition with $h(f_i(z_1)) \neq h(f_i(z_2))$ is such that $\vec{c} \in {q''}^D$ as well, where $z_1 \neq z_2$ is the only inequality atom of the disjunct q'' of q_1 for which f_i is a homomorphism to q'. So, at each iteration the algorithm equates $f_i(z_1)$ and $f_i(z_2)$ in q' to consider all the other possible representative databases in which $f_i(z_1) = f_i(z_2)$.

If f_{l+1} is a homomorphism from a disjunct q'' of q_1 to the resulting q' such that either $f_{l+1}(z_1) \neq f_{l+1}(z_2)$ is an inequality atom of q' (where $z_1 \neq z_2$ is the only inequality atom of q'') or $f_{l+1}(z_1)$ and $f_{l+1}(z_2)$ are both constants, then we clearly have $q' \sqsubseteq q_1$, and therefore $q' \sqsubseteq q$.

Finally, in the case that f_m is a homomorphism from a disjunct of q_2 to the resulting q', due to the above considerations, it can be easily proven that each database D is such that either ${q'}^D \subseteq {q_1}^D$, or ${q'}^D \subseteq {q_2}^D$. Thus, $q' \sqsubseteq q$, as required.

"**Only-if part:**" Suppose there is no such good sequence F with respect to q' and q, and consider the following changes to the CQ$^{\neq}$ q': for each possible homomorphism h from a disjunct of q_1 to q', we replace everywhere in q' one of the two terms $h(z_1)$ and $h(z_2)$ with the other, as described in step 6 of the algorithm, where $z_1 \neq z_2$ is the inequality atom of the disjunct of q_1 for which h is a homomorphism to q'.

Since there are $n_{q'}$ terms in the CQ$^{\neq}$ q', there can be at most $n_{q'} - 1$ of such homomorphisms. Indeed, after applying $n_{q'} - 1$ replacing as described above, the resulting q' would contain only one term.

For the CQ$^{\neq}$ q' and UCQ$^{1,\neq}$ $q = q_1 \cup q_2$ we have that (i) there is no homomorphism h from a disjunct of q_1 to q' with either $h(z_1) \neq h(z_2)$, or $h(z_1)$ and $h(z_2)$ being both constants, where $z_1 \neq z_2$ is the inequality atom of the disjunct for which h is a homomorphism to q'; and (ii) there is no disjunct of q_2 for which there is a homomorphism to q'. In proof, if either point (i) or point (ii) is not true, then we would easily derive a contradiction on the fact that there exists no good sequence F with respect to q' and q.

Consider now the freezing of the resulting CQ$^{\neq}$ q', i.e., the set of facts $D_{q'}$, here denoted by D for the sake of readability, corresponding to the set of all atoms occurring in the body of q' that are not inequality atoms, but where each variable v is replaced with a different fresh constant c_v. Let, moreover, $\vec{c'}$ be the tuple of constants obtained from the target list $\vec{t'}$ of the resulting q' after replacing each distinguished variable v with the constant c_v.

Clearly, D and $\vec{c'}$ are such that $\vec{c'} \in {q'}^D$ by construction. Furthermore, due to the facts that q' and q are such that both the above points (i) and (ii) hold, we derive that $\vec{c'} \notin q^D$. Thus, ${q'}^D \not\subseteq q^D$, and therefore $q' \not\sqsubseteq q$, as required. □

For UCQs with no inequalities, it is known that query containment is in NP (in particular, NP-complete), and is in PTime when the containing query q is assumed to be fixed. We are now ready to show that the same holds even when the contained query q' is a UCQ$^{\neq}$ and the containing query q is a UCQ$^{1,\neq}$.

Theorem 4.21. *The containment problem for UCQ$^{\neq}$s when the containing query q is a UCQ$^{1,\neq}$ is in NP (and therefore NP-complete). Moreover, it is in* PTime *when the containing UCQ$^{1,\neq}$ q is assumed to be fixed.*

Proof. Given a UCQ$^{\neq}$ q' and UCQ$^{1,\neq}$ q, observe that $q' \sqsubseteq q$ if and only if $q'' \sqsubseteq q$ for each disjunct q'' of q', where q'' is a CQ$^{\neq}$.

Due to Proposition 4.10, checking whether $q'' \sqsubseteq q$ for a CQ$^{\neq}$ q'' and UCQ$^{1,\neq}$ q can be done as follows:

- Guess a list $F = [f_1, \ldots, f_m]$ of $m \leq n_{q''}$ functions from the set of variables and constants occurring in q to the set of variables and constants occurring in q'';

- Check whether F is a good sequence with respect to q'' and q, where checking whether F is a good sequence with respect to q'' and q can be done by means of the above described CheckFContains algorithm.

So, we easily derive a nondeterministic algorithm deciding $q' \sqsubseteq q$. With an NP step, for each disjunct q'' of q', we guess a list $F_{q''} = [f_1, \ldots, f_m]$ with $m \leq n_{q''}$ of functions from the set of variables and constants occurring in q to the set of variables and constants occurring in q''. Notice that this can be done in polynomial time in the size of q'. Then, for each disjunct q'' of q, by exploiting the CheckFContains algorithm we check whether $F_{q''}$ is a good sequence with respect to q'' and q using: (i) for each $i \in [1, m-1]$, a polynomial time step for checking whether f_i is a homomorphism from a disjunct of q_1 to q'', and for replacing everywhere in q'' one of the two terms among $f_i(z_1)$ and $f_i(z_2)$ with the other; finally, (ii) another polynomial time step for checking whether f_m is a homomorphism from some disjunct of the UCQ q_2 to the resulting CQ$^{\neq}$ q''. □

Notice, however, that when the containing query is assumed to be fixed, the containment problem for UCQs with no inequalities can be solved even in AC^0 (it is indeed sufficient to evaluate the containing query q over the database associated to the contained query q' [Chandra and Merlin, 1977]). On the other hand, as shown in Theorem 4.20, the containment problem for UCQ$^{1,\neq}$s is PTime-hard when the containing query is assumed to be fixed. Since AC^0 $\subset$ PTime, this makes the two problems significantly different. We conclude the chapter by characterising the computational complexity of the containment problem for UCQ$^{1,\neq}$s, which can be obtained by combining Theorem 4.20 with the above theorem.

Corollary 4.7. *The containment problem for UCQ$^{\neq}$s when the containing query q is a UCQ$^{1,\neq}$ is NP-complete. Moreover, it is* PTime*-complete when the containing query q is assumed to be fixed.*

Chapter 5

Complete Source-to-Ontology Rewritings

In this chapter, we study both the verification, and the computation problem for complete source-to-ontology rewritings. In what follows, given a syntactic object x such as a UCQ, an ontology, or a mapping, we denote by $\sigma(x)$ its size, i.e., the number of symbols needed to write it, with names of predicates, variables, etc. counting as one.

5.1 Verification Problem

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

Lemma 5.1. *$q_{\mathcal{O}}$ is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ if and only if $q_{\mathcal{S}} \sqsubseteq (\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \cup \mathsf{PerfRef}_{\mathcal{V}_{\mathcal{O}}^n,\Sigma})$, where $n = ar(q_{\mathcal{O}}) = ar(q_{\mathcal{S}})$.*

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

"**If part:**" Suppose, for the sake of contradiction, that $q_{\mathcal{O}}$ is not a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, that is, there exists an $\mathcal{S}$-database D consistent with Σ such that $q_{\mathcal{S}}^D \not\subseteq cert_{q_{\mathcal{O}},\Sigma}^D$. Since D is consistent with Σ, we have (i) $\mathsf{PerfRef}_{\mathcal{V}_{\mathcal{O}},\Sigma}^D = \emptyset$, which implies that (i) $\mathsf{PerfRef}_{\mathcal{V}_{\mathcal{O}}^n,\Sigma}^D = \emptyset$ and (ii) $cert_{q_{\mathcal{O}},\Sigma}^D = \mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}^D$. Therefore, for the $\mathcal{S}$-database D, we have that $q_{\mathcal{S}}^D \not\subseteq (\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \cup \mathsf{PerfRef}_{\mathcal{V}_{\mathcal{O}}^n,\Sigma})^D$. Thus, $q_{\mathcal{S}} \not\sqsubseteq (\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \cup \mathsf{PerfRef}_{\mathcal{V}_{\mathcal{O}}^n,\Sigma})$, as required. $\square$

The following theorem characterises the computational complexity of the verification problem for complete source-to-ontology rewritings.

Theorem 5.1. *The verification problem for complete source-to-ontology rewritings is NP-complete.*

Proof. As for the upper bound, by virtue of Lemma 5.1, it is sufficient to show how to check the containment $q_{\mathcal{S}} \sqsubseteq (\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},\Sigma})$ in NP, where $n = ar(q_{\mathcal{O}})$. In particular, for every disjunct q of $q_{\mathcal{S}}$, (i) we guess a query q' over $\mathcal{O}$ with the same arity of $q_{\mathcal{O}}$ and size at most the maximum between $\sigma(q_{\mathcal{O}})$ and $\sigma(\mathcal{V}^n_{\mathcal{O}})$, a sequence ρ of ontology assertions, a query q'' over $\mathcal{S}$ with the same arity of $q_{\mathcal{O}}$ and size at most $\sigma(\mathcal{M}) \times \sigma(q')$, and a function ϕ from the variables of q'' to the variables of q, and (ii) we check in polynomial time whether we can rewrite either $q_{\mathcal{O}}$ or $\mathcal{V}^n_{\mathcal{O}}$ into q' through ρ, q'' is in $\mathsf{MapRef}(q', \mathcal{M})$, and ϕ is a homomorphism from q'' to q.

As for the lower bound, the proof of NP-hardness is by a LogSpace reduction from the 3-colourability problem, which is NP-complete [Garey *et al.*, 1976]. 3-colourability is the problem of deciding, given an undirected graph $G = (V, E)$ with no loops, whether G is 3-colourable, i.e., whether there exists a function $f : V \to \{R, G, B\}$ such that $f(y_i) \neq f(y_j)$ for each $(y_i, y_j) \in E$.

Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification as follows: ontology $\mathcal{O}$ contains no axioms, i.e., $\mathcal{O} = \emptyset$, schema $\mathcal{S}$ involves a binary predicate E and three unary predicates s_R, s_G, and s_B, and finally the mapping $\mathcal{M} = \{m_1, m_2, m_3, m_4\}$ with:

$$
\begin{array}{lrcl}
m_1 : & E(x_1, x_2) & \to & P(x_1, x_2), \\
m_2 : & s_R(x) & \to & R(x), \\
m_3 : & s_G(x) & \to & G(x), \\
m_4 : & s_B(x) & \to & B(x),
\end{array}
$$

where R, G, and B are atomic concepts, whereas P is an atomic role.

Let $V = (y_1, y_2, \ldots, y_n)$, then we define a boolean CQ $q_{\mathcal{O}}$ over $\mathcal{O}$ as follows:

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

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

Observe that $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and $q_{\mathcal{S}}$ do not depend on the input of the 3-COLOURABILITY problem (i.e., $G = (V, E)$), whereas $q_{\mathcal{O}}$ can be constructed in LogSpace from it.

We now show that G is 3-colourable if and only if $q_{\mathcal{O}}$ is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. To begin observe that $\mathcal{V}_{\mathcal{O}} \equiv \bot$, and hence, for each $\mathcal{S}$-database D, we have that $cert^D_{q_{\mathcal{O}},\Sigma} = \mathsf{PerfRef}^D_{q_{\mathcal{O}},\Sigma}$, where $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ is the CQ over $\mathcal{S}$:

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

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

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

It can be readily seen that h consists in a homomorphism from $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ to $q_{\mathcal{S}}$. It follows that $q_{\mathcal{S}} \sqsubseteq \mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ which, due to Lemma 5.1, implies that $q_{\mathcal{O}}$ is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

"**If part:**" Suppose that G is not 3-colourable, that is, every possible function $f : V \to \{R, G, B\}$ is such that $f(y_i) = f(y_j)$ for some $(y_i, y_j) \in E$. It is not hard to see that this implies that there exists no homomorphism from $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ to $q_{\mathcal{S}}$. Therefore, since $\mathcal{V}_{\mathcal{O}} \equiv \bot$, we have that $q_{\mathcal{S}} \not\sqsubseteq (\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \cup \mathsf{PerfRef}_{\mathcal{V}^3_{\mathcal{O}},\Sigma})$ which, due to Lemma 5.1, implies that $q_{\mathcal{O}}$ is not a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. □

Note that the result of NP-hardness already holds when $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ is fixed (i.e., it does not depend on the input of the reduction) with $\mathcal{O}$ containing no axioms and $\mathcal{M}$ being both a pure GAV mapping and a LAV mapping, $q_{\mathcal{S}}$ is a fixed CQJFE, and finally $q_{\mathcal{O}}$ is a CQ.

5.2 Computation Problem

We now provide the algorithm $\mathsf{MinimallyComplete}$ for computing UCQ-minimally complete source-to-ontology rewritings.

Algorithm 5.1 $\mathsf{MinimallyComplete}$

Input:
OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$;
UCQ $q_{\mathcal{S}} = q^1_{\mathcal{S}} \cup \ldots \cup q^n_{\mathcal{S}}$ over $\mathcal{S}$, where $q^i_{\mathcal{S}} = \{\vec{t_i} \mid \exists \vec{y_i}.\phi_i(\vec{x_i}, \vec{y_i})\}$ for each $i \in [1, n]$

Output:
UCQ $q_{\mathcal{O}}$ over $\mathcal{O}$

1: $q_{\mathcal{O}} := \{\vec{t_1} \mid \exists \vec{y_1}.\mathcal{M}(q^1_{\mathcal{S}}) \wedge \top(\vec{x_1})\} \cup \ldots \cup \{\vec{t_n} \mid \vec{y_n}.\mathcal{M}(q^n_{\mathcal{S}}) \wedge \top(\vec{x_n})\}$, where $\vec{y_i}$ includes the set of existential variables of $q^i_{\mathcal{S}}$ occurring in $\mathcal{M}(q_{\mathcal{S}})$ plus the fresh existential variables introduced by $\mathcal{M}(q_{\mathcal{S}})$, for each $i \in [1, n]$
2: **return** $q_{\mathcal{O}}$

Informally, for each disjunct $q^i_{\mathcal{S}}$ of $q_{\mathcal{S}}$, the algorithm obtains a CQ by simply chasing (the incomplete $\mathcal{S}$-database associated to) $q^i_{\mathcal{S}}$ with respect to $\mathcal{M}$, using $\top$ to bind the distinguished variables that are not involved in the application of $\mathcal{M}$ to $q^i_{\mathcal{S}}$. Finally, the output query $q_{\mathcal{O}}$ is the union of all such CQs.

Example 5.1. Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the following OBDM specification:

- $\mathcal{O} = \emptyset$

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

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

Let the data service be expressed as the following UCQ $q_{\mathcal{S}}$ over $\mathcal{S}$:

$$\begin{array}{l} \{(x_1,x_2) \mid \exists y_1, y_2. s_1(x_1) \wedge s_2(x_1,y_1) \wedge s_2(y_2,x_2)\} \cup \\ \{(x_1,c_3) \mid \exists y_1, y_2. s_1(c_1) \wedge s_3(x_1,y_1) \wedge s_2(y_1,y_2)\}. \end{array}$$

One can verify that $\mathsf{MinimallyComplete}(\Sigma, q_{\mathcal{S}})$ returns the UCQ $q_{\mathcal{O}}$:

$$\begin{array}{l} \{(x_1,x_2) \mid \exists y_3. P_1(x_1,y_3) \wedge A_1(y_3) \wedge \top(x_2)\} \cup \\ \{(x_1,c_3) \mid \exists y_1, y_3. P_1(c_1,y_3) \wedge A_1(y_3) \wedge P_3(x_1,c_2) \wedge P_4(x_1,y_1)\}, \end{array}$$

which corresponds to the unique (up to equivalence w.r.t. Σ) UCQ-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. □

The following theorem establishes termination and correctness of the Minimally-Complete algorithm.

Theorem 5.2. *MinimallyComplete$(\Sigma, q_{\mathcal{S}})$ terminates and returns the unique (up to equivalence w.r.t. Σ) UCQ-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.*

Proof. Termination of the algorithm easily follows from the termination of the chase of a source instance (possibly containing variables) with respect to a GLAV mapping, or, equivalently, with respect to a set of source-to-target tgds [Fagin *et al.*, 2005a].

As for the correctness, we first show that the computed $q_{\mathcal{O}} = q^1_{\mathcal{O}} \cup \ldots \cup q^n_{\mathcal{O}}$, with $q^i_{\mathcal{O}} = \{\vec{t_i} \mid \exists \vec{y_i}. \mathcal{M}(q^i_{\mathcal{S}}) \wedge \top(\vec{x_i})\}$ for every $i \in [1,n]$, is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. By construction, for every $i \in [1,n]$, the CQ $q^i_{\mathcal{S}}$ corresponds to, or it is contained in, a disjunct of $\mathsf{MapRef}(q^i_{\mathcal{O}}, \mathcal{M})$. Thus, $q^i_{\mathcal{S}} \sqsubseteq \mathsf{PerfRef}_{q^i_{\mathcal{O}},\Sigma}$ holds for every $i \in [1,n]$. It follows that $q_{\mathcal{S}} \sqsubseteq \mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ which, due to Lemma 5.1, implies that $q_{\mathcal{O}}$ is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. We now show that $q_{\mathcal{O}}$ is actually the unique (up to equivalence w.r.t. Σ) UCQ-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, that is, each UCQ $q'_{\mathcal{O}}$ that is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ is such that $cert_{q_{\mathcal{O}},\Sigma} \sqsubseteq cert_{q'_{\mathcal{O}},\Sigma}$ (cf. Definition 3.7). We do this by way of contradiction.

Let $q'_{\mathcal{O}}$ be a UCQ such that $cert_{q_{\mathcal{O}},\Sigma} \not\sqsubseteq cert_{q'_{\mathcal{O}},\Sigma}$, that is, there exists an $\mathcal{S}$-database D consistent with Σ such that $cert^D_{q_{\mathcal{O}},\Sigma} \not\subseteq cert^D_{q'_{\mathcal{O}},\Sigma}$. It follows that there is a tuple of constant $\vec{c} = (c_1, \ldots, c_m)$ such that $\vec{c} \notin cert^D_{q'_{\mathcal{O}},\Sigma}$, but $\vec{c} \in cert^D_{q_{\mathcal{O}},\Sigma}$, i.e., $\vec{c} \in cert^D_{q^i_{\mathcal{O}},\Sigma}$ for some $i \in [1,n]$. Consider now the freezing of $q^i_{\mathcal{S}} = \{\vec{t_i} \mid \exists y_i. \phi_i(\vec{x_i}, \vec{y_i})\}$, i.e., the set $D_{q^i_{\mathcal{S}}}$ (here denoted by D_i) of all facts over $\mathcal{S}$ obtained from $\phi_i(\vec{x_i}, \vec{y_i})$ by replacing each variable $v \in \vec{x_i} \cup \vec{y_i}$ with a different fresh constant denoted by c_v. Let, moreover, $\vec{c^i}$ be the freezed tuple of constants $\vec{c^i} = (c^i_1, \ldots, c^i_m)$ where, for each $j \in [1,m]$, $c^i_j = t_j$ if t_j is a constant, and $c^i_j = c_x$ if $t_j = x$. We now prove that $\vec{c^i} \in {q^i_{\mathcal{S}}}^{D_i}$ and $\vec{c^i} \notin cert^{D_i}_{q'_{\mathcal{O}},\Sigma}$, thus showing that $q'_{\mathcal{O}}$ is not a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

Obviously, $\vec{c^i} \in {q^i_{\mathcal{S}}}^{D_i}$ trivially holds. Consider $\mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}}$, i.e., the canonical structure of $\mathcal{O}$ with respect to $\mathcal{M}$ and D. Since $\vec{c} \in cert^{D}_{q^i_{\mathcal{O}},\Sigma}$, there exists a homomorphism h from $q^i_{\mathcal{O}}$ to $\mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}}$ for which $h(\vec{t_i}) = \vec{c}$. Furthermore, due to the facts that $\mathcal{M}$ is a GLAV mapping and $\mathcal{O}$ is a *DL-Lite*$_{\mathcal{R}}$ ontology, and due to the fact that $\vec{c}$ is in the evaluation of $q^i_{\mathcal{O}}$ over $\mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}}$ (i.e., $\vec{c} \in {q^i_{\mathcal{O}}}^{\mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}}}$), by construction of $q^i_{\mathcal{O}}$ and D_i it is easy to see the existence of a function f from $\mathcal{C}^{\mathcal{M}(D_i)}_{\mathcal{O}}$ to $\mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}}$ for which (i) $f(c) = h(c) = c$ for each constant c occurring in $q^i_{\mathcal{O}}$, (ii) $f(c_v) = h(v)$ for each variable $v \in \vec{x_i} \cup \vec{y_i}$ of $q^i_{\mathcal{S}}$ occurring in $\mathcal{M}(q^i_{\mathcal{S}})$, and (iii) $f(\mathcal{C}^{\mathcal{M}(D_i)}_{\mathcal{O}}) \subseteq \mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}}$, where $f(\mathcal{C}^{\mathcal{M}(D_i)}_{\mathcal{O}})$ is the image of $\mathcal{C}^{\mathcal{M}(D_i)}_{\mathcal{O}}$ under f. Observe that $f(\vec{c^i}) = \vec{c}$, and, since D is consistent with Σ, D_i is consistent with Σ as well.

Due to the existence of this function f and the assumption that $\vec{c} \notin cert^{D}_{q'_{\mathcal{O}},\Sigma}$, we derive that there is no disjunct $q' = \{\vec{t'} \mid \exists y'.\phi'(\vec{x'}, \vec{y'})\}$ of $q'_{\mathcal{O}}$ for which there is a homomorphism h' from q' to $\mathcal{C}^{\mathcal{M}(D_i)}_{\mathcal{O}}$ such that $h'(\vec{t'}) = \vec{c^i}$ (otherwise, the composition function $f \circ h'$ would result in a homomorphism from q' to $\mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}}$ such that $f(h'(\vec{t'})) = \vec{c}$, and therefore this would contradict the assumption that $\vec{c} \notin cert^{D}_{q'_{\mathcal{O}},\Sigma}$). Thus, $\vec{c^i} \notin cert^{D_i}_{q'_{\mathcal{O}},\Sigma}$ as well. To conclude the proof, observe that D_i is an $\mathcal{S}$-database consistent with Σ for which $\vec{c^i} \in {q^i_{\mathcal{S}}}^{D_i}$ (and so $\vec{c^i} \in {q_{\mathcal{S}}^{D_i}}^i$) and $\vec{c^i} \notin cert^{D_i}_{q'_{\mathcal{O}},\Sigma}$, thus implying that $q'_{\mathcal{O}}$ is not a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. $\square$

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

Corollary 5.1. *The unique (up to equivalence w.r.t. Σ) UCQ-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ always exists. Furthermore, if $q_{\mathcal{S}}$ is a CQ, then it can be expressed as a CQ as well.*

Regarding the cost of the MinimallyComplete algorithm, we observe that, essentially, it applies the chase of each (possibly incomplete source instance associated to the) disjunct of $q_{\mathcal{S}}$ via the conjunction of atoms that appear in the left-hand side of the assertions in $\mathcal{M}$. This results in a running time that does not depend on $\mathcal{O}$ and $\mathcal{S}$, is exponential in $\sigma(\mathcal{M})$, and only polynomial in $\sigma(q_{\mathcal{S}})$.

Notice, however, that if $\mathcal{M}$ is a LAV mapping, then the application of the chase is feasible in polynomial time even in $\sigma(\mathcal{M})$ (indeed, in this case there is no conjunction of atoms to evaluate when applying the chase), and therefore the running time of the algorithm becomes polynomial in the size of the input of the problem.

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

We start by proving the following lemma.

Lemma 5.2. *If q_1 is a UCQJFE and q_2 is a UCQ, then the problem of checking whether $q_1 \sqsubseteq q_2$ is NP-complete.*

Proof. The membership in NP follows from the membership in NP of the more general case of containment between UCQs [Sagiv and Yannakakis, 1980].

We now show that the problem is NP-hard even when q_1 is a boolean CQJFE, q_2 is a boolean CQ, and both q_1 and q_2 use only binary predicates. We follow a proof strategy that is similar to the reduction illustrated in [Aho *et al.*, 1979]. In particular, we provide a LogSpace reduction from the 3-CNF problem, which is NP-complete [Karp, 1972]. 3-CNF is the problem of deciding, given a CNF formula $F = c_1 \wedge \ldots \wedge c_m$ on a set of variables $Y = \{y_1, \ldots, y_n\}$, whether F is *satisfiable*, that is, whether there exists a truth assignment $V = \{v_1, \ldots, v_n\}$ to the variables in Y that satisfies F. Each clause c_i is a disjunction of three literals, where each literal is either a variable $x \in X$ or its negated. For $i = 1, \ldots, m$, we denote by $z_{i,1}$, $z_{i,2}$, and $z_{i,3}$ the first, the second, and the third, respectively, variable appearing (either positive or negated) in clause c_i.

Let F be an instance of the 3-CNF problem. We now define a boolean CQJFE q_1 and a boolean CQ q_2, where both queries use only binary predicates.

As for q_1, it is the conjunction of the atoms appearing in its body: for each clause c_i of F, and for each of the seven satisfying truth assignments $A_{i,k} = \{v_1, v_2, v_3\}$ for c_i (where, for each $k = 1, \ldots, 7$, $A_{i,k}$ is a constant, and, for each $j = 1, 2, 3$, v_j is either the constant 1 or 0), the body of q_1 contains the atoms $s_{i,1}(A_{i,k}, v_1)$, $s_{i,2}(A_{i,k}, v_2)$, and $s_{i,3}(A_{i,k}, v_3)$.

As for q_2, it is the conjunction of the atoms appearing in its body: for each clause c_i of F, the body of q_2 contains the atoms $s_{i,1}(a_i, z_{i,1})$, $s_{i,2}(a_i, z_{i,2})$, and $s_{i,3}(a_i, z_{i,3})$, where a_i denotes a fresh existential variable.

To illustrate the reduction, consider the formula $F = (y_1 \vee y_2 \vee \neg y_3) \bigwedge (\neg y_1 \vee y_2 \vee \neg y_4)$. In this case, the reduction produces the following boolean CQJFE

$$\begin{aligned} q_1 = \{() \mid\ & \alpha_1(A_{1,1}, 0, 0, 0) \wedge \alpha_1(A_{1,2}, 0, 1, 0) \wedge \alpha_1(A_{1,3}, 0, 1, 1) \wedge \alpha_1(A_{1,4}, 1, 0, 0) \wedge \\ & \alpha_1(A_{1,5}, 1, 0, 1) \wedge \alpha_1(A_{1,6}, 1, 1, 0) \wedge \alpha_1(A_{1,7}, 1, 1, 1) \wedge \alpha_2(A_{2,1}, 0, 0, 0) \wedge \\ & \alpha_2(A_{2,2}, 0, 0, 1) \wedge \alpha_2(A_{2,3}, 0, 1, 0) \wedge \alpha_2(A_{2,4}, 0, 1, 1) \wedge \alpha_2(A_{2,5}, 1, 0, 0) \wedge \\ & \alpha_2(A_{2,6}, 1, 1, 0) \wedge \alpha_2(A_{2,7}, 1, 1, 1)\}, \end{aligned}$$

where an atom of the form $\alpha_i(x, y, z, w)$ is a shortcut for the conjunction of atoms $s_{i,1}(x, y) \wedge s_{i,2}(x, z) \wedge s_{i,3}(x, w)$.

Moreover, the reduction produces the following boolean CQ

$$\begin{aligned} q_2 = \{() \mid \exists a_1, a_2, y_1, y_2, y_3, y_4 . s_{1,1}(a_1, y_1) \wedge s_{1,2}(a_1, y_2) \wedge s_{1,3}(a_1, y_3) \wedge \\ s_{2,1}(a_2, y_1) \wedge s_{2,2}(a_2, y_2) \wedge s_{2,3}(a_2, y_4)\}. \end{aligned}$$

With the same arguments given in [Aho *et al.*, 1979], it is possible to prove that every truth assignment satisfying formula F consists in a homomorphism from q_2 to q_1 and vice versa.

In particular, from every truth assignment $V = \{v_1, \ldots, v_n\}$ to the variables in Y that satisfies F it is possible to derive a homomorphism h from q_2 to q_1 in the following way: (*i*) $h(x_j) = v_j$ for each $j \in [1, n]$, and (*ii*) $h(a_i) = A_{i,k}$ for each $i \in [1, m]$, where

$k \in [1,7]$ is an arbitrary value for which $\alpha_i(A_{i,k}, h(z_{i,1}), h(z_{i,2}), h(z_{i,3}))$ is an atom of q_1 (observe that, since by assumption clause c_i is satisfied under truth assignment V, at least one constant $A_{i,k}$ for some $k \in [1,7]$ must exists by construction).

Also, from every homomorphism h from q_2 to q_1, the truth assignment $V = \{h(y_1), \ldots, h(y_n)\}$ satisfies formula F by construction.

From the above considerations, it follows that F is satisfiable if and only if $q_1 \sqsubseteq q_2$. To conclude the proof, we observe that both the boolean CQJFE q_1 and the boolean CQ q_2 can be constructed in LogSpace from F. □

Proposition 5.1. *Assuming* PTime $\subsetneq$ *NP, the computation problem for complete source-to-ontology rewritings can not be solved in polynomial time.*

Proof. Let $q_1 = \{() \mid \exists \vec{y_1}.\psi(\vec{y_1})\}$ and $q_2 = \{() \mid \exists \vec{y_2}.\phi(\vec{y_2})\}$ be a boolean CQJFE and a boolean CQ, respectively. We define an OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ as follows: ontology $\mathcal{O}$ contains no axioms, schema $\mathcal{S}$ is composed of all predicates involved in $\psi(\vec{y_1})$ and in $\phi(\vec{y_2})$ plus an additional fresh unary predicate s, and finally the mapping $\mathcal{M}$ comprises only the following pure GAV mapping assertion:

$$\exists \vec{y_2}.s(x) \land \phi(\vec{y_2}) \to A(x),$$

where A is an atomic concept of $\mathcal{O}$.

We also define a boolean CQJFE over $\mathcal{S}$: $q_{\mathcal{S}} = \{() \mid \exists \vec{y_1}.\exists y.s(y) \land \psi(\vec{y_1})\}$, where y denotes a fresh existential variable occurring neither in $\vec{y_1}$ nor in $\vec{y_2}$.

Note that the unique (up to equivalence w.r.t. Σ) UCQ-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ is either the query $q_{\mathcal{O}} = \{() \mid \exists y.A(y)\}$ if it is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, or the query $q'_{\mathcal{O}} = \{() \mid \exists y.\top(y)\}$.

Specifically, we now prove that $q_{\mathcal{O}}$ is the unique (up to equivalence w.r.t. Σ) UCQ-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting if and only if $q_1 \sqsubseteq q_2$. Due to Lemma 5.1, $q_{\mathcal{O}}$ is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ if and only if $q_{\mathcal{S}} \sqsubseteq \mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \cup \mathsf{PerfRef}_{\mathcal{V}_{\mathcal{O}},\Sigma}$. Since $\mathcal{V}_{\mathcal{O}} \equiv \bot$, in this case we have that $q_{\mathcal{O}}$ is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ if and only if $q_{\mathcal{S}} \sqsubseteq \mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$. Notice, however, that $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} = \{() \mid \exists \vec{y_2}.\exists y.s(y) \land \phi(\vec{y_2})\}$, and therefore $q_{\mathcal{S}} \sqsubseteq \mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ if and only if $q_1 \sqsubseteq q_2$.

We have reduced the problem of checking whether $q_1 \sqsubseteq q_2$ for a boolean CQJFE q_1 and a boolean CQ q_2 to the problem of computing the unique (up to equivalence w.r.t. Σ) UCQ-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of a CQJFE $q_{\mathcal{S}}$, where both $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and $q_{\mathcal{S}}$ can be constructed in LogSpace from q_1 and q_2.

Thus, a polynomial time algorithm for computing UCQ-minimally complete source-to-ontology rewritings of CQJFEs $q_{\mathcal{S}}$ would imply a polynomial time algorithm for checking whether $q_1 \sqsubseteq q_2$, where q_1 is a CQJFE and q_2 is a CQ. Since by Lemma 5.2 we know that this latter containment problem is in general NP-hard, it follows that, unless PTime = NP, the computation problem for complete source-to-ontology rewritings can not be solved in polynomial time, even in the case of pure GAV mappings $\mathcal{M}$ and CQJFEs $q_{\mathcal{S}}$. □

5.3 Improving by means of Inequalities

In this section, we consider an extension of the scenario introduced in Section 3.2 in which $\mathcal{L}_{\mathcal{O}}$ is the class of UCQ$^{\neq}$s rather than UCQs. Specifically, we tackle the problem of computing UCQ$^{\neq}$-minimally complete source-to-ontology rewritings.

The next example shows that the usage of inequalities is an interesting feature to consider when providing abstractions of data services. In general, UCQ$^{\neq}$s provide better approximated complete source-to-ontology rewritings compared to UCQs.

Example 5.2. Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the following OBDM specification:

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

$$\begin{array}{rrcl} m_1: & s_1(x_1, x_2) & \rightarrow & P(x_1, x_2), \\ m_2: & s_1(x, x) & \rightarrow & A(x), \\ m_3: & s_2(x_1, x_2) & \rightarrow & P(x_1, x_2). \end{array}$$

Let the query over $\mathcal{S}$ be the CQ $q_{\mathcal{S}} = \{(x_1, x_2) \mid s_1(x_1, x_2)\}$.

Then, $\mathsf{MinimallyComplete}(\Sigma, q_{\mathcal{S}})$ returns the CQ $q_{\mathcal{O}} = \{(x_1, x_2) \mid P(x_1, x_2)\}$, which is the unique (up to equivalence w.r.t. Σ) UCQ-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. Consider, however, the UCQ$^{\neq}$ $q'_{\mathcal{O}} = \{(x_1, x_2) \mid P(x_1, x_2) \wedge x_1 \neq x_2\} \cup \{(x, x) \mid P(x, x) \wedge A(x)\}$. Consider any $\mathcal{S}$-database D and any tuple $(c_1, c_2) \in q_{\mathcal{S}}^D$. If $c_1 \neq c_2$, then $(c_1, c_2) \in cert^D_{q'_{\mathcal{O}},\Sigma}$ because of the first disjunct. Conversely, if $c_1 = c_2$, then $(c_1, c_2) \in cert^D_{q'_{\mathcal{O}},\Sigma}$ because of the second disjunct. This clearly proves that $q'_{\mathcal{O}}$ is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

Furthermore, notice that $cert_{q'_{\mathcal{O}},\Sigma} \sqsubset cert_{q_{\mathcal{O}},\Sigma}$ holds. On the one hand, $cert_{q'_{\mathcal{O}},\Sigma} \sqsubseteq cert_{q_{\mathcal{O}},\Sigma}$ is trivially true. On the other hand, for the $\mathcal{S}$-database $D = \{s_2(c, c)\}$, we have $cert^D_{q'_{\mathcal{O}},\Sigma} = \emptyset$, whereas $cert^D_{q_{\mathcal{O}},\Sigma} = \{(c, c)\}$, and therefore $cert^D_{q'_{\mathcal{O}},\Sigma} \subsetneq cert^D_{q_{\mathcal{O}},\Sigma}$. Thus, $q'_{\mathcal{O}}$ is a better complete approximation of $q_{\mathcal{S}}$ (w.r.t. Σ) compared to $q_{\mathcal{O}}$. □

The above example also suggests an algorithm for computing UCQ$^{\neq}$-minimally complete source-to-ontology rewritings. Informally, before of chasing (the incomplete $\mathcal{S}$-database associated to) the various disjuncts $q^i_{\mathcal{S}}$ of $q_{\mathcal{S}}$ with respect to $\mathcal{M}$, the basic idea is to first compute the so-called inequality saturation of $q_{\mathcal{S}}$ with respect to $\mathsf{conL}_{\mathcal{M}}$, and only after chasing each (incomplete $\mathcal{S}$-database associated to the) disjunct of the obtained query. Here, for a mapping $\mathcal{M}$, $\mathsf{conL}_{\mathcal{M}}$ denotes the set of all constants occurring in the left-hand side of mapping assertions in $\mathcal{M}$.

Roughly speaking, the *inequality saturation* of a UCQ $q_{\mathcal{S}}$ with respect to a set of constants con is a UCQ$^{\neq}$ obtained by replacing each CQ $q^i_{\mathcal{S}}$ of $q_{\mathcal{S}}$ with an equivalent UCQ$^{\neq}$. This latter is obtained from $q^i_{\mathcal{S}}$ by computing a CQ$^{\neq}$ for each possible unification between the terms in $\mathsf{ter}(q^i_{\mathcal{S}}) \cup \mathsf{con}$, and then imposing inequalities between the syntactically different remaining variables. Here, for a CQ q, $\mathsf{ter}(q)$ denotes the set of all terms (i.e., constants and variables) occurring in q.

We now present the algorithm $\mathsf{Saturate}$ that, given a UCQ Q and a set of constants con, returns a logically equivalent UCQ$^{\neq}$ representing the inequality saturation of Q with respect to con. Essentially, the core of the algorithm is the one illustrated in [Lembo *et al.*, 2015], but extended in order to deal with a set of constants con given as input. We include below the algorithm for the sake of completeness.

In the algorithm, two terms t_1 and t_2 are *compatible* if t_1 and t_2 denote distinct terms and at least one of them is a variable. Furthermore, for a query q, $q[t_1/t_2]$

denotes the query obtained from q by replacing every occurrence (even in the target list) of the term t_1 in q with the term t_2 (obviously, if one of the two terms is a constant, then we always assume that t_2 is the constant and t_1 is the variable).

Algorithm 5.2 Saturate

Input:
  UCQ Q;
  set of constants con
Output:
  UCQ$^{\neq}$ Q'

repeat
  $Q'' := Q$
  for each CQ $q \in Q''$ **do**
    for each pair of compatible terms t_1, t_2 in $\mathsf{ter}(q) \cup \mathsf{con}$ **do**
      $Q := Q \cup q[t_1/t_2]$
    end for
  end for
until $Q'' = Q$
$Q' := \emptyset$
for each $q \in Q''$ **do**
  for each pair of compatible t_1, t_2 in $\mathsf{ter}(q) \cup \mathsf{con}$ **do**
    Add the inequality atom $t_1 \neq t_2$ in conjunction to the body of q
  end for
  $Q' := Q' \cup q$
end for
return Q'

For a UCQ Q and a set of constants con, $\mathsf{Saturate}(Q, \mathsf{con})$ first computes the UCQ Q'' by unifying compatible terms in each query $q \in Q$ in all possible ways. Then, for any query $q \in Q''$ and for each pair of terms t_1 and t_2 that are syntactically different and compatible, it adds the inequality atom $t_1 \neq t_2$ to the body of q.

The following example illustrates an execution of the $\mathsf{Saturate}$ algorithm.

Example 5.3. Let $q_{\mathcal{S}}$ be the UCQ $q_{\mathcal{S}} = q^1_{\mathcal{S}} \cup q^2_{\mathcal{S}}$, where $q^1_{\mathcal{S}} = \{(x) \mid \exists y. s_1(x, y)\}$ and $q^2_{\mathcal{S}} = \{(x) \mid s_2(x, c_2)\}$, and let $\mathsf{con} = \{c_1\}$. One can verify that $\mathsf{Saturate}(q_{\mathcal{S}}, \mathsf{con})$ returns the UCQ$^{\neq}$ $Q' = q^{1,1}_{\mathcal{S}} \cup q^{1,2}_{\mathcal{S}} \cup q^{1,3}_{\mathcal{S}} \cup q^{1,4}_{\mathcal{S}} \cup q^{1,5}_{\mathcal{S}} \cup q^{2,1}_{\mathcal{S}} \cup q^{2,2}_{\mathcal{S}} \cup q^{2,3}_{\mathcal{S}}$, where:

- $q^{1,1}_{\mathcal{S}} = \{(x) \mid \exists y. s_1(x, y) \wedge x \neq y \wedge x \neq c_1 \wedge y \neq c_1\}$;
- $q^{1,2}_{\mathcal{S}} = \{(x) \mid s_1(x, x) \wedge x \neq c_1\}$;
- $q^{1,3}_{\mathcal{S}} = \{(c_1) \mid \exists y. s_1(c_1, y) \wedge y \neq c_1\}$;
- $q^{1,4}_{\mathcal{S}} = \{(x) \mid s_1(x, c_1) \wedge x \neq c_1\}$;
- $q^{1,5}_{\mathcal{S}} = \{(c_1) \mid s_1(c_1, c_1)\}$;
- $q^{2,1}_{\mathcal{S}} = \{(x) \mid s_2(x, c_2) \wedge x \neq c_1 \wedge x \neq c_2\}$;

- $q_{\mathcal{S}}^{2,2} = \{(c_1) \mid s_2(c_1, c_2)\}$;
- $q_{\mathcal{S}}^{2,3} = \{(c_2) \mid s_2(c_2, c_2)\}$. $\square$

We are now ready to focus on the problem of computing UCQ$^{\neq}$-minimally complete source-to-ontology rewritings, and present algorithm $\mathsf{MinimallyCompleteIneq}$.

In the algorithm, we write each CQ$^{\neq}$ q as $q = \{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y}) \wedge \xi(\vec{x}, \vec{y})\}$, where $\phi(\vec{x}, \vec{y})$ is the conjunction of the atoms occurring in the body of q that are not inequality atoms, whereas $\xi(\vec{x}, \vec{y})$ is the conjunction of all the inequality atoms occurring in the body of q. Furthermore, $\mathcal{M}(q)$ for a CQ$^{\neq}$ q is computed by simply ignoring the inequality atoms of q, and thus treating it as a CQ. Finally, $\xi(\vec{x}, \vec{\mathscr{y}})$ is the conjunction of the inequality atoms obtained from $\xi(\vec{x}, \vec{y})$ by removing all those atoms of the form $y \neq t$ and $t \neq y$ in which y is an existential variable occurring in $\vec{y}$ but not in $\vec{\mathscr{y}}$ (i.e., not in $\mathcal{M}(q)$), and t is any other possible term.

Algorithm 5.3 $\mathsf{MinimallyCompleteIneq}$

Input:
OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$;
UCQ $q_{\mathcal{S}}$ over $\mathcal{S}$
Output:
UCQ$^{\neq}$ $q_{\mathcal{O}}$ over $\mathcal{O}$

$q_{\mathcal{O}} = \emptyset$
for each CQ$^{\neq}$ $q \in \mathsf{Saturate}(q_{\mathcal{S}}, \mathsf{conL}_{\mathcal{M}})$ **do**
  Let $q = \{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y}) \wedge \xi(\vec{x}, \vec{y})\}$
  $q_{\mathcal{O}} := q_{\mathcal{O}} \cup \{\vec{t} \mid \exists \vec{\mathscr{y}}.\mathcal{M}(q) \wedge \top(\vec{x}) \wedge \xi(\vec{x}, \vec{\mathscr{y}})\}$
end for
return $q_{\mathcal{O}}$

Informally, the algorithm first computes the inequality saturation of $q_{\mathcal{S}}$ with respect to $\mathsf{conL}_{\mathcal{M}}$. Then, for each CQ$^{\neq}$ $q \in \mathsf{Saturate}(q_{\mathcal{S}}, \mathsf{conL}_{\mathcal{M}})$ the algorithm obtains a CQ$^{\neq}$ by chasing (the incomplete $\mathcal{S}$-database associated to) the conjunctive part of q with respect to $\mathcal{M}$, using $\top$ to bind the distinguished variables that are not involved in the application of $\mathcal{M}$, and adding the inequality atoms of q to the so obtained query, where the inequality atoms involving existential variables not appearing in $\mathcal{M}(q)$ are removed. Finally, the output UCQ$^{\neq}$ $q_{\mathcal{O}}$ is the union of all such CQ$^{\neq}$s.

Example 5.4. Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the following OBDM specification:

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

$$\begin{array}{rrcl} m_1: & \exists y. s_1(x,y) & \rightarrow & \text{AdvertisingCompany}(x), \\ m_2: & s_1(c_1,x) & \rightarrow & \text{PublicCompany}(x), \\ m_3: & s_2(x_1,x_2) & \rightarrow & \text{Controls}(x_1,x_2), \\ m_4: & s_2(x,x) & \rightarrow & \text{SelfHoldingCompany}(x), \\ m_5: & s_3(x_1,x_2) & \rightarrow & \text{Controls}(x_1,x_2); \\ m_6: & s_4(x) & \rightarrow & \text{AdvertisingCompany}(x). \end{array}$$

Let $q_{\mathcal{S}}$ be the UCQ illustrated in Example 5.3. Since $\mathsf{conL}_{\mathcal{M}} = \{c_1\}$, $\mathsf{Saturate}(q_{\mathcal{S}}, \mathsf{conL}_{\mathcal{M}})$ is the UCQ$^{\neq}$ $Q' = q_{\mathcal{S}}^{1,1} \cup q_{\mathcal{S}}^{1,2} \cup q_{\mathcal{S}}^{1,3} \cup q_{\mathcal{S}}^{1,4} \cup q_{\mathcal{S}}^{1,5} \cup q_{\mathcal{S}}^{2,1} \cup q_{\mathcal{S}}^{2,2} \cup q_{\mathcal{S}}^{2,3}$ illustrated in Example 5.3. One can therefore verify that $\mathsf{MinimallyCompleteIneq}(\Sigma, q_{\mathcal{S}})$ returns the UCQ$^{\neq}$ $q_{\mathcal{O}} = q_{\mathcal{O}}^{1,1} \cup q_{\mathcal{O}}^{1,2} \cup q_{\mathcal{O}}^{1,3} \cup q_{\mathcal{O}}^{1,4} \cup q_{\mathcal{O}}^{1,5} \cup q_{\mathcal{O}}^{2,1} \cup q_{\mathcal{O}}^{2,2} \cup q_{\mathcal{O}}^{2,3}$, where:

- $q_{\mathcal{O}}^{1,1} = q_{\mathcal{O}}^{1,2} = \{(x) \mid \text{AdvertisingCompany}(x) \wedge x \neq c_1\}$;
- $q_{\mathcal{O}}^{1,3} = \{(c_1) \mid \exists y. \text{AdvertisingCompany}(c_1) \wedge \text{PublicCompany}(y) \wedge y \neq c_1\}$;
- $q_{\mathcal{O}}^{1,4} = \{(x) \mid \text{AdvertisingCompany}(x) \wedge x \neq c_1\}$;
- $q_{\mathcal{O}}^{1,5} = \{(c_1) \mid \text{AdvertisingCompany}(c_1) \wedge \text{PublicCompany}(c_1)\}$;
- $q_{\mathcal{O}}^{2,1} = \{(x) \mid \text{Controls}(x, c_2) \wedge x \neq c_1 \wedge x \neq c_2\}$;
- $q_{\mathcal{O}}^{2,2} = \{(c_1) \mid \text{Controls}(c_1, c_2)\}$;
- $q_{\mathcal{S}}^{2,3} = \{(c_2) \mid \text{Controls}(c_2, c_2) \wedge \text{SelfHoldingCompany}(c_2)\}$,

which corresponds to the unique (up to equivalence w.r.t. Σ) UCQ$^{\neq}$-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ rewriting of $q_{\mathcal{S}}$. □

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

Theorem 5.3. *MinimallyCompleteIneq$(\Sigma, q_{\mathcal{S}})$ terminates and returns the unique (up to equivalence w.r.t. Σ) UCQ$^{\neq}$-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.*

Proof. Termination of the algorithm follows from (i) the termination of the chase of a source instance (possibly containing variables) with respect to a GLAV mapping; (ii) the termination of $\mathsf{Saturate}(q_{\mathcal{S}}, \mathsf{conL}_{\mathcal{M}})$, which is guaranteed by the fact that $q_{\mathcal{S}}$ is a disjunction of a finite number of CQs in which there is a finite number of atoms and terms, and the fact that $\mathsf{conL}_{\mathcal{M}}$ is a set of a finite number of constants.

As for the correctness, we first show that the computed $q_{\mathcal{O}}$ is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. Let $\vec{c} = (c_1, \ldots, c_n)$ be any tuple of constants such that $\vec{c} \in q_{\mathcal{S}}^{D}$. We now prove that $\vec{c} \in cert_{q_{\mathcal{O}},\Sigma}^{D}$. Clearly, $\vec{c} \in q_{\mathcal{S}}^{D}$ implies the presence of a UCQ$^{\neq}$ $q \in \mathsf{Saturate}(q_{\mathcal{S}}, \mathsf{conL}_{\mathcal{M}})$ where $q = \{\vec{t} \mid \exists \vec{y}. \phi(\vec{x}, \vec{y}) \wedge \xi(\vec{x}, \vec{y})\}$ for which $\vec{c} \in q^{D}$, i.e., there is a homomorphism h from the set of atoms occurring in $\phi(\vec{x}, \vec{y})$ to D for which (i) $h(\vec{t}) = \vec{c}$ and (ii) $h(\alpha) \neq h(\beta)$ for each inequality atom $\alpha \neq \beta$ occurring in $\xi(\vec{x}, \vec{y})$.

Consider now the CQ$^{\neq}$ $q' = \{\vec{t} \mid \exists \vec{y}. \mathcal{M}(q) \wedge \top(\vec{x}) \wedge \xi(\vec{x}, \vec{y})\}$, which is by construction a disjunct of $q_{\mathcal{O}}$. Clearly, due to the existence of h, there must be a homomorphism h' from the set of all atoms occurring in $\mathcal{M}(q)$ and $\top(\vec{x})$ to $\mathcal{M}(D)$ for which $h'(t) = h(t)$ for each term t occurring in q. This implies that (i) $h'(\vec{t}) = \vec{c}$;

(*ii*) $h'(y)$ is a constant for each existential variable of q' which was also in q; and (*iii*) since atoms in $\xi(\vec{x}, \vec{\mathcal{y}})$ are a subset of those in $\xi(\vec{x}, \vec{y})$, we have $h'(\alpha) \neq h'(\beta)$ for each inequality atom $\alpha \neq \beta$ occurring in $\xi(\vec{x}, \vec{\mathcal{y}})$, where both $h'(\alpha)$ and $h'(\beta)$ are constants due to (*ii*). Since we are adopting the UNA, we have that $(h'(\alpha))^{\mathcal{I}} \neq (h'(\beta))^{\mathcal{I}}$ for each inequality atom $\alpha \neq \beta$ occurring in $\xi(\vec{x}, \vec{\mathcal{y}})$ and for each model $\mathcal{I} \in Mod_D(\Sigma)$.

Due to the existence of homomorphism h' from atoms occurring in $\mathcal{M}(q)$ and $\top(\vec{x})$ to $\mathcal{M}(D)$ satisfying (*i*), (*ii*), and (*iii*), and the above observation regarding the UNA, we derive that $(c_1^{\mathcal{I}}, \ldots, c_n^{\mathcal{I}}) \in q'^{\mathcal{I}}$ for each model $\mathcal{I} \in Mod_D(\Sigma)$. Therefore, by definition of certain answers, $\vec{c} \in cert^D_{q',\Sigma}$. Thus, $\vec{c} \in cert^D_{q_{\mathcal{O}},\Sigma}$, as required.

We now show that $q_{\mathcal{O}}$ is actually the unique (up to equivalence w.r.t. Σ) UCQ$^{\neq}$-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, that is, each UCQ$^{\neq}$ $q'_{\mathcal{O}}$ that is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ is such that $cert_{q_{\mathcal{O}},\Sigma} \sqsubseteq cert_{q'_{\mathcal{O}},\Sigma}$ (cf. Definition 3.7). We do this by way of contradiction.

Let $q'_{\mathcal{O}}$ be a UCQ$^{\neq}$ such that $cert_{q_{\mathcal{O}},\Sigma} \not\sqsubseteq cert_{q'_{\mathcal{O}},\Sigma}$, that is, there exists an $\mathcal{S}$-database D consistent with Σ such that $cert^D_{q_{\mathcal{O}},\Sigma} \not\subseteq cert^D_{q'_{\mathcal{O}},\Sigma}$. It follows that there is a tuple of constant $\vec{c} = (c_1, \ldots, c_n)$ such that $\vec{c} \notin cert^D_{q'_{\mathcal{O}},\Sigma}$, but $\vec{c} \in cert^D_{q_{\mathcal{O}},\Sigma}$. By the definition of certain answers, there exists a model $\mathcal{I} \in Mod_D(\Sigma)$ such that $(c_1^{\mathcal{I}}, \ldots, c_n^{\mathcal{I}}) \notin {q'_{\mathcal{O}}}^{\mathcal{I}}$, whereas $(c_1^{\mathcal{I}}, \ldots, c_n^{\mathcal{I}}) \in q_{\mathcal{O}}^{\mathcal{I}}$. We now show the existence of an $\mathcal{S}$-database D' for which (*i*) $\vec{c} \in q_{\mathcal{S}}^{D'}$, and (*ii*) $\mathcal{I} \in Mod_{D'}(\Sigma)$ (and therefore $\vec{c} \notin cert^{D'}_{q'_{\mathcal{O}},\Sigma}$), thus proving that $q'_{\mathcal{O}}$ is not a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

Since $(c_1^{\mathcal{I}}, \ldots, c_n^{\mathcal{I}}) \in q_{\mathcal{O}}^{\mathcal{I}}$, there is a disjunct in $q_{\mathcal{O}}$ of the form $\{\vec{t} \mid \exists \vec{\mathcal{y}}.\mathcal{M}(q) \wedge \top(\vec{x}) \wedge \xi(\vec{x}, \vec{\mathcal{y}})\}$ for which there is a homomorphism h from atoms occurring in $\mathcal{M}(q)$ and $\top(\vec{x})$ to $\mathcal{I}$ satisfying (*i*) $h(\vec{t}) = \vec{c}$ and (*ii*) $h(\alpha) \neq h(\beta)$ for each inequality atom $\alpha \neq \beta$ occurring in $\xi(\vec{x}, \vec{\mathcal{y}})$. Observe that $q \in \mathsf{Saturate}(q_{\mathcal{S}}, \mathsf{conL}_{\mathcal{M}})$ is a CQ$^{\neq}$ of the form $\{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y}) \wedge \xi(\vec{x}, \vec{y})\}$. Consider the $\mathcal{S}$-database $D' = h'(\phi(\vec{x}, \vec{y}))$, where h' extends h by assigning a different fresh constant to each existential variable $y \in \vec{y}$ not occurring in $\vec{\mathcal{y}}$ (i.e., not occurring in $\mathcal{M}(q)$).

Clearly, we have that $\vec{c} \in q^{D'}$. Furthermore, due to the existence of the homomorphism h from atoms occurring in $\mathcal{M}(q)$ and $\top(\vec{x})$ to $\mathcal{I}$ and by construction of D', we have $\langle D', \mathcal{I} \rangle \models \mathcal{M}$. Notice that this implies $\mathcal{I} \in Mod_{D'}(\Sigma)$. Indeed, on the one hand $\langle D', \mathcal{I} \rangle \models \mathcal{M}$, and, on the other hand, $\mathcal{I} \models \mathcal{O}$ due to the initial assumption that $\mathcal{I} \in Mod_D(\Sigma)$. But then, $\vec{c} \in q^{D'}$ for a query $q \in \mathsf{Saturate}(q_{\mathcal{S}}, \mathsf{conL}_{\mathcal{M}})$ implies that $\vec{c} \in q_{\mathcal{S}}^{D'}$, and $(c_1^{\mathcal{I}}, \ldots, c_n^{\mathcal{I}}) \notin {q'_{\mathcal{O}}}^{\mathcal{I}}$ for a model $\mathcal{I} \in Mod_{D'}(\Sigma)$ implies that $\vec{c} \notin cert^{D'}_{q'_{\mathcal{O}},\Sigma}$. It follows that, for the $\mathcal{S}$-database D' consistent with Σ, $\vec{c} \in q^{D'}$ and $\vec{c} \notin cert^{D'}_{q'_{\mathcal{O}},\Sigma}$. Thus, $q'_{\mathcal{O}}$ is not a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, as required. $\square$

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

Corollary 5.2. *The unique (up to equivalence w.r.t.* Σ*) UCQ*$^{\neq}$*-minimally complete* $\mathcal{S}$*-to-*$\mathcal{O}$ Σ*-rewriting of* $q_{\mathcal{S}}$ *always exists.*

Differently from the $\mathsf{MinimallyComplete}$ algorithm, the running time of the $\mathsf{MinimallyCompleteIneq}$ algorithm becomes exponential in $\sigma(q_{\mathcal{S}})$. This is due to the computation of the inequality saturation of $q_{\mathcal{S}}$ with respect to $\mathsf{conL}_{\mathcal{M}}$, which is done by means of the $\mathsf{Saturate}$ algorithm that, in general, even for a CQ Q and an empty set of constants as input, it produces a number of queries that is exponential in the number of terms occurring in Q. As a result, the MinimallyCompleteIneq algorithm returns, in general, a UCQ$^{\neq}$ having exponentially many disjuncts with respect to the number of terms occurring in $q_{\mathcal{S}}$. Finally, we point out that it is straightforward to construct cases where the number of disjuncts of the unique (up to equivalence w.r.t. Σ) UCQ$^{\neq}$-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ is necessarily exponential in the number of terms occurring in $q_{\mathcal{S}}$.

5.4 Dropping the UNA

Let us now study complete source-to-ontology rewritings when the UNA is not adopted. Clearly, since *DL-Lite*$_{\mathcal{R}}$ is insensitive to the adoption of the UNA for UCQ answering, from Theorem 5.2 we easily derive the following corollary.

Corollary 5.3. *When the UNA is not adopted,* MinimallyComplete$(\Sigma, q_{\mathcal{S}})$ *terminates and returns the unique (up to equivalence w.r.t.* Σ*) UCQ-minimally complete* $\mathcal{S}$*-to-*$\mathcal{O}$ Σ*-rewriting of* $q_{\mathcal{S}}$*.*

Differently from the UCQ case, the next example shows that a UCQ$^{\neq}$ $q_{\mathcal{O}}$ that is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ when the UNA is adopted is not necessarily so when the UNA is not adopted.

Example 5.5. Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, $q_{\mathcal{S}}$, and $q'_{\mathcal{O}}$ be the OBDM specification, the CQ over $\mathcal{S}$, and the UCQ$^{\neq}$ over $\mathcal{O}$, respectively, illustrated in Example 5.2.

Consider the $\mathcal{S}$-database $D = \{s_1(c_1, c_2)\}$, and the interpretation $\mathcal{I}$ for $\langle \Sigma, D \rangle$ with $c_1^{\mathcal{I}} = c_2^{\mathcal{I}} = c$, $P^{\mathcal{I}} = \{(c, c)\}$, and $A^{\mathcal{I}} = \emptyset$. Clearly, by definition of mapping satisfaction (cf. Subsection 2.6.3), we have $\langle D, \mathcal{I} \rangle \models \mathcal{M}$. Since $\mathcal{I} \models \mathcal{O}$ trivially holds because $\mathcal{O} = \emptyset$, we have $\mathcal{I} \in Mod_D(\Sigma)$ (when the UNA is not adopted). Notice, however, that ${q'_{\mathcal{O}}}^{\mathcal{I}} = \emptyset$ because the inequality atom of the first disjunct is not satisfied. Thus, $(c_1, c_2) \notin cert^D_{q'_{\mathcal{O}}, \Sigma}$, whereas $(c_1, c_2) \in q^D_{\mathcal{S}}$. Therefore, when the UNA is not adopted, $q'_{\mathcal{O}}$ is not a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

As we will see, the unique (up to equivalence w.r.t. Σ) UCQ$^{\neq}$-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ is the UCQ $q_{\mathcal{O}}$ illustrated again in Example 5.2. □

Quite interestingly, the next theorem proves that, when the UNA is not adopted, for each OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and query $q_{\mathcal{S}}$ of our scenario under consideration, the unique (up to equivalence w.r.t. Σ) UCQ$^{\neq}$-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ always coincides with the unique (up to equivalence w.r.t. Σ) UCQ-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

Theorem 5.4. *When the UNA is not adopted,* MinimallyComplete$(\Sigma, q_{\mathcal{S}})$ *terminates and returns the unique (up to equivalence w.r.t.* Σ*) UCQ$^{\neq}$-minimally complete* $\mathcal{S}$*-to-*$\mathcal{O}$ Σ*-rewriting of* $q_{\mathcal{S}}$*.*

Proof. Termination is already proven in Theorem 5.2.

The fact that the computed query $q_{\mathcal{O}}$ is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ when the UNA is not adopted follows from the facts that it is so when the UNA is adopted (cf. Theorem 5.2) and the insensitiveness of the UNA for UCQ answering. We now prove that, when the UNA is not adopted, $q_{\mathcal{O}}$ is actually the unique (up to equivalence w.r.t. Σ) UCQ$^{\neq}$-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, that is, each UCQ$^{\neq}$ $q'_{\mathcal{O}}$ that is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ is such that $cert_{q_{\mathcal{O}},\Sigma} \sqsubseteq cert_{q'_{\mathcal{O}},\Sigma}$ (cf. Definition 3.7). We do this by way of contradiction.

Let $q'_{\mathcal{O}}$ be a UCQ$^{\neq}$ such that $cert_{q_{\mathcal{O}},\Sigma} \not\sqsubseteq cert_{q'_{\mathcal{O}},\Sigma}$, that is, there exists an $\mathcal{S}$-database D consistent with Σ such that $cert^{D}_{q_{\mathcal{O}},\Sigma} \not\subseteq cert^{D}_{q'_{\mathcal{O}},\Sigma}$. It follows that there is a tuple of constant $\vec{c} = (c_1, \ldots, c_m)$ such that $\vec{c} \notin cert^{D}_{q'_{\mathcal{O}},\Sigma}$, but $\vec{c} \in cert^{D}_{q_{\mathcal{O}},\Sigma}$. By the definition of certain answers, there exists a model $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle \in Mod_D(\Sigma)$ such that $(c_1^{\mathcal{I}}, \ldots, c_m^{\mathcal{I}}) \notin {q'_{\mathcal{O}}}^{\mathcal{I}}$, whereas $(c_1^{\mathcal{I}}, \ldots, c_n^{\mathcal{I}}) \in q_{\mathcal{O}}^{\mathcal{I}}$. We now show the existence of an $\mathcal{S}$-database D' for which (i) $\vec{c} \in q_{\mathcal{S}}^{D'}$, and (ii) $\mathcal{I} \in Mod_{D'}(\Sigma)$ (and therefore $\vec{c} \notin cert^{D'}_{q'_{\mathcal{O}},\Sigma}$), thus proving that $q'_{\mathcal{O}}$ is not a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

Since $(c_1^{\mathcal{I}}, \ldots, c_m^{\mathcal{I}}) \in q_{\mathcal{O}}^{\mathcal{I}}$, there is a homomorphism h from a CQ $q^i_{\mathcal{O}}$ of $q_{\mathcal{O}}$ for some $i \in [1, n]$ to $\mathcal{I}$, where $q^i_{\mathcal{O}} = \{\vec{t_i} \mid \exists \vec{y_i}.\mathcal{M}(q^i_{\mathcal{S}}) \wedge \top(\vec{x_i})\}$. Consider now the freezing of $q^i_{\mathcal{S}} = \{\vec{t_i} \mid \exists y_i.\phi_i(\vec{x_i}, \vec{y_i})\}$, i.e., the set $D_{q^i_{\mathcal{S}}}$ (here denoted by D') of all facts over $\mathcal{S}$ obtained from $\phi_i(\vec{x_i}, \vec{y_i})$ by replacing each variable $v \in \vec{x_i} \cup \vec{y_i}$ with a fresh constant denoted by c_v. Let, moreover, $\vec{c'}$ be the freezed tuple of constants $\vec{c'} = (c'_1, \ldots, c'_m)$ where, for each $j \in [1, m]$, $c'_j = t_j$ if t_j is a constant, and $c'_j = c_x$ if $t_j = x$.

Let $\mathcal{I}' = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}'} \rangle$ be the interpretation for $\langle \Sigma, D' \rangle$ with the same domain of $\mathcal{I}$ and the interpretation function $\cdot^{\mathcal{I}'}$ such that (i) $c^{\mathcal{I}'} = c^{\mathcal{I}}$ for each constant c occurring in $q^i_{\mathcal{S}}$; (ii) $c_{\mathcal{y}}^{\mathcal{I}'} = h(\mathcal{y})$, for each existential variable $\mathcal{y} \in \vec{\mathcal{y}_i}$ occurring in $q^i_{\mathcal{O}}$; $c_x^{\mathcal{I}'} = h(x)$ for each distinguished variable $x \in \vec{x_i}$ occurring in $q^i_{\mathcal{S}}$; (iv) $c_y^{\mathcal{I}'}$ is any object of $\Delta^{\mathcal{I}}$, for each existential variable $y \in \vec{y_i}$ not occurring in $\mathcal{M}(q^i_{\mathcal{S}})$; and (v) the extensions of atomic concepts and atomic roles of $\mathcal{O}$ is the same as in the interpretation $\mathcal{I}$. Observe that $({c'_1}^{\mathcal{I}'}, \ldots, {c'_m}^{\mathcal{I}'}) = (c_1^{\mathcal{I}}, \ldots, c_m^{\mathcal{I}})$.

Obviously, $\vec{c'} \in {q^i_{\mathcal{S}}}^{D'}$ trivially holds. Furthermore, due to the existence of the homomorphism h from atoms occurring in $\mathcal{M}(q^i_{\mathcal{S}})$ and $\top(\vec{x})$ to $\mathcal{I}$, and by construction of D' and $\mathcal{I}'$, we have $\langle D', \mathcal{I}' \rangle \models \mathcal{M}$. Notice that this implies $\mathcal{I}' \in Mod_{D'}(\Sigma)$. Indeed, on the one hand $\langle D', \mathcal{I}' \rangle \models \mathcal{M}$, and, on the other hand, $\mathcal{I}' \models \mathcal{O}$ due to the initial assumption that $\mathcal{I} \in Mod_D(\Sigma)$ and the fact that $\mathcal{I}$ and $\mathcal{I}'$ have the same extensions of atomic concepts and atomic roles. But then, $(c_1^{\mathcal{I}}, \ldots, c_m^{\mathcal{I}}) \notin {q'_{\mathcal{O}}}^{\mathcal{I}}$ implies $({c'_1}^{\mathcal{I}'}, \ldots, {c'_m}^{\mathcal{I}'}) \notin {q'_{\mathcal{O}}}^{\mathcal{I}'}$ since $\mathcal{I}$ and $\mathcal{I}'$ have the same extension of atomic concepts and atomic roles and constants of $q^i_{\mathcal{S}}$ are interpreted in the same way in $\mathcal{I}$ and $\mathcal{I}'$. Thus, $({c'_1}^{\mathcal{I}'}, \ldots, {c'_m}^{\mathcal{I}'}) \notin {q'_{\mathcal{O}}}^{\mathcal{I}'}$ for a model $\mathcal{I}' \in Mod_{D'}(\Sigma)$ implies $\vec{c'} \notin cert^{D'}_{q'_{\mathcal{O}},\Sigma}$.

It follows that, for the $\mathcal{S}$-database D' consistent with Σ, we have $\vec{c'} \in {q^i_{\mathcal{S}}}^{D'}$ (and therefore $\vec{c'} \in q_{\mathcal{S}}^{D'}$) and $\vec{c'} \notin cert^{D'}_{q'_{\mathcal{O}},\Sigma}$. This allows us to conclude that $q'_{\mathcal{O}}$ is not a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, as required. □

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

Corollary 5.4. *When the UNA is not adopted, the unique (up to equivalence w.r.t.* Σ*) UCQ*$^{\neq}$*-minimally complete* $\mathcal{S}$*-to-*$\mathcal{O}$ Σ*-rewriting of* $q_{\mathcal{S}}$ *always exists and can be expressed as a UCQ. Furthermore, if* $q_{\mathcal{S}}$ *is a CQ, then it can be expressed as a CQ as well.*

Using techniques developed on Chapter 4, we conclude this chapter with a result on a particular instance of the verification problem when the UNA is not adopted.

Theorem 5.5. *When the UNA is not adopted, the problem of checking whether a CQ$^{\neq,b}$ $q_{\mathcal{O}}$ is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of a UCQ $q_{\mathcal{S}}$ is NP-complete.*

Proof. Observe that Theorem 5.1 holds regardless of whether the UNA is adopted or not. Thus, NP-hardness trivially follows from the NP-hardness result of Theorem 5.1, which holds even when $q_{\mathcal{O}}$ is a CQ.

Notice that, due to Theorem 4.4, when the UNA is not adopted, for a CQ$^{\neq,b}$ $q_{\mathcal{O}}$ over $\mathcal{O}$ of arity n, the UCQ $\mathsf{NoUNAPerfRef}_{q_{\mathcal{O}},\Sigma} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},\Sigma}$ over $\mathcal{S}$ is the perfect $\mathcal{O}$-to-$\mathcal{S}$ Σ-rewriting of $q_{\mathcal{O}}$, i.e., $cert^D_{q_{\mathcal{O}},\Sigma} = (\mathsf{NoUNAPerfRef}_{q_{\mathcal{O}},\Sigma} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},\Sigma})^D$ for each $\mathcal{S}$-database D. So, analogously to Lemma 5.1, it is easy to see that, when the UNA is not adopted, $q_{\mathcal{O}}$ is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ if and only if $q_{\mathcal{S}} \sqsubseteq (\mathsf{NoUNAPerfRef}_{q_{\mathcal{O}},\Sigma} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},\Sigma})$, where $n = ar(q_{\mathcal{O}}) = ar(q_{\mathcal{S}})$.

As for the upper bound, it is therefore sufficient to show how to check the containment $q_{\mathcal{S}} \sqsubseteq (\mathsf{NoUNAPerfRef}_{q_{\mathcal{O}},\Sigma} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},\Sigma})$ in NP. For this, it is enough to slightly extend the nondeterministic algorithm illustrated in the upper bound proof of Theorem 5.1 in the following way: (i) we replace the guessed sequence ρ of ontology assertions with the sequence ρ', where ρ' extends ρ by possibly including some disjointness assertions of $\mathcal{O}$ used to rewrite the inequality atoms of $q_{\mathcal{O}}$; (ii) Before of checking in polynomial time whether we can apply ρ, we preliminarily check, again in polynomial time, whether we can rewrite inequality atoms of $q_{\mathcal{O}}$ through the additional assertions of ρ' (obtaining so a CQ in $\lambda(q_{\mathcal{O}}, \mathcal{O})$, cf. Section 4.2). $\square$

Chapter 6

Sound Source-to-Ontology Rewritings

In this chapter, we study both the verification, and the computation problem for sound source-to-ontology rewritings.

6.1 Verification Problem

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

Lemma 6.1. *$q_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ if and only if $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \sqsubseteq (q_{\mathcal{S}} \ \cup \ \mathsf{PerfRef}_{\mathcal{V}^{n}_{\mathcal{O}},\Sigma})$, where $n = ar(q_{\mathcal{O}}) = ar(q_{\mathcal{S}})$.*

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

"**If part:**" Suppose, for the sake of contradiction, that $q_{\mathcal{O}}$ is not a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, that is, there exists an $\mathcal{S}$-database D consistent with Σ such that $cert^{D}_{q_{\mathcal{O}},\Sigma} \not\subseteq q^{D}_{\mathcal{S}}$. Since D is consistent with Σ, we have (i) $\mathsf{PerfRef}^{D}_{\mathcal{V}_{\mathcal{O}},\Sigma} = \emptyset$, which implies that (i) $\mathsf{PerfRef}^{D}_{\mathcal{V}^{n}_{\mathcal{O}},\Sigma} = \emptyset$ and (ii) $cert^{D}_{q_{\mathcal{O}},\Sigma} = \mathsf{PerfRef}^{D}_{q_{\mathcal{O}},\Sigma}$. Therefore, for the $\mathcal{S}$-database D, we have that $\mathsf{PerfRef}^{D}_{q_{\mathcal{O}},\Sigma} \not\subseteq (q_{\mathcal{S}} \ \cup \ \mathsf{PerfRef}_{\mathcal{V}^{n}_{\mathcal{O}},\Sigma})^{D}$. Thus, $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \not\sqsubseteq q_{\mathcal{S}} \ \cup \ \mathsf{PerfRef}_{\mathcal{V}^{n}_{\mathcal{O}},\Sigma}$, as required. □

The following theorem characterises the computational complexity of the verification problem for sound source-to-ontology rewritings.

Theorem 6.1. *The verification problem for sound source-to-ontology rewritings is Π^{p}_{2}-complete.*

Proof. As for the upper bound, by virtue of Lemma 6.1, it is sufficient to show how to check the containment $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \sqsubseteq (q_{\mathcal{S}} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},\Sigma})$ in Π^p_2, where $n = ar(q_{\mathcal{O}})$. In particular, checking whether $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \not\sqsubseteq (q_{\mathcal{S}} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},\Sigma})$ can be done in Σ^p_2 in the following way: (*i*) we guess a CQ q_1 over $\mathcal{S}$ with the same arity of $q_{\mathcal{O}}$ and size at most $\sigma(\mathcal{M}) \times \sigma(q_{\mathcal{O}})$, and (*ii*) with an NP-oracle, similarly to what described in Theorem 5.1, we first check whether q_1 corresponds to, or it is contained in, a disjunct of $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$, i.e., whether $q_1 \sqsubseteq \mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$, and then whether $q_1 \not\sqsubseteq (q_{\mathcal{S}} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},\Sigma})$, again using the method mentioned in Theorem 5.1.

As for the lower bound, the proof of Π^p_2-hardness is by a LogSpace reduction from the $\forall\exists$-CNF problem, which is Π^p_2-complete [Stockmeyer, 1976]. $\forall\exists$-CNF is the problem of deciding, given a 3-CNF formula $F = c_1 \wedge \ldots \wedge c_p$ on a set of variables $Y = \{y_1, \ldots, y_m\} \cup X = \{x_1, \ldots, x_n\}$ such that the variables in Y (respectively, X) are universally (respectively, existentially) quantified, whether F is true, i.e., whether for each truth assignment to the variables in Y, there exists a truth assignment to the variables in X that satisfies F. Each clause c_i is a disjunction of three literals, where each literal is either a variable $z \in Y \cup X$ or its negated. For $i = 1, \ldots, p$, we denote by $z_{i,1}$, $z_{i,2}$, $z_{i,3}$ the first, the second, and the third, respectively, variable appearing (either positive or negated) in clause c_i.

We follow a proof strategy that is similar to [Millstein *et al.*, 2003, Theorem 3.3], in which the reduction can be seen as an extension of the one provided in Lemma 5.2.

In particular, we define an OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ with $\mathcal{O}$ containing no axioms, and $\mathcal{S}$ and $\mathcal{M}$ as follows. For each clause c_i of F, schema $\mathcal{S}$ contains three binary predicates, namely $s_{i,1}$, $s_{i,2}$, and $s_{i,3}$. Intuitively, the first argument of all three predicates represents the clause c_i, while, for each $j = 1, 2, 3$, the second argument of predicate $s_{i,j}$ represents the variable $z_{i,j}$. Additionally, $\mathcal{S}$ also contains m unary predicates e_i, one for each universally quantified variable y_i, and two further unary predicates, namely *zero* and *one*. The mapping $\mathcal{M}$ is composed of two parts.

The first part simply mirrors each relation $s_{i,j}$ into the atomic role $R_{i,j}$ of $\mathcal{O}$ through the mapping assertion $s_{i,j}(x_1, x_2) \rightarrow R_{i,j}(x_1, x_2)$, for each $i = 1, \ldots, n$ and for each $j = 1, 2, 3$.

The second part of mapping $\mathcal{M}$ mirrors each relation e_i into the atomic concept W_i of $\mathcal{O}$ through the mapping assertion $e_i(x) \rightarrow W_i(x)$, for each $i = 1, \ldots, m$. Finally, in the second part of $\mathcal{M}$ there are two further mapping assertions representing the possible truth value (either 0 or 1) for the universally quantified variables in Y: $zero(x) \rightarrow H_1(x, 0) \wedge \ldots \wedge H_m(x, 0)$, and $one(x) \rightarrow H_1(x, 1) \wedge \ldots \wedge H_m(x, 1)$, where 0 and 1 are constants, and, for each $i = 1, \ldots, m$, H_i is an atomic role of $\mathcal{O}$.

We define the CQ $q_{\mathcal{S}}$ as the conjunction of the atoms appearing in its body: (*i*) for each clause c_i of F, the body of $q_{\mathcal{S}}$ contains the atoms $s_{i,1}(a_i, z_{i,1})$, $s_{i,2}(a_i, z_{i,2})$, and $s_{i,3}(a_i, z_{i,3})$, where a_i denotes a fresh existential variable, and (*ii*) for each universally quantified variable $y_i \in Y$, there is also the atom $e_i(y_i)$.

Analogously, we define the CQ $q_{\mathcal{O}}$ as the conjunction of the atoms appearing in its body: (*i*) for each clause c_i of F, and for each of the seven satisfying truth assignments $A_{i,k} = \{v_1, v_2, v_3\}$ for c_i (where, for each $k = 1, \ldots, 7$, $A_{i,k}$ is a constant, and, for each $j = 1, 2, 3$, v_j is either the constant 1 or the constant 0), the body of $q_{\mathcal{O}}$ contains the atoms $R_{i,1}(A_{i,k}, v_1)$, $R_{i,2}(A_{i,k}, v_2)$, and $R_{i,3}(A_{i,k}, v_3)$, and (*ii*) for each universally quantified variable $y_i \in Y$, there are also the atoms $W_i(y_i)$ and $H_i(u_i, y_i)$, where u_i denotes a fresh existential variable.

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

To illustrate the reduction, we use the following formula: $F = (x_1 \vee x_2 \vee y_1) \bigwedge (\neg x_1 \vee \neg x_2 \vee \neg y_2)$. In this case, the reduction would produce the mapping $\mathcal{M}$ composed of the following mapping assertions:

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

and the following CQs $q_{\mathcal{S}}$ and $q_{\mathcal{O}}$:

$$
\begin{aligned}
q_{\mathcal{S}} = \{() \mid \exists a_1, a_2, x_1, x_2, y_1, y_2 . & s_{1,1}(a_1, x_1) \wedge s_{1,2}(a_1, x_2) \wedge s_{1,3}(a_1, y_1) \wedge \\
& s_{2,1}(a_2, x_1) \wedge s_{2,2}(a_2, x_2) \wedge s_{2,3}(a_2, y_2) \wedge \\
& e_1(y_1) \wedge e_2(y_2)\};
\end{aligned}
$$

$$
\begin{aligned}
q_{\mathcal{O}} = \{() \mid \exists u_1, u_2, y_1, y_2 . & \beta_1(A_{1,1}, 0, 0, 1) \wedge \beta_1(A_{1,2}, 0, 1, 0) \wedge \beta_1(A_{1,3}, 0, 1, 1) \wedge \\
& \beta_1(A_{1,4}, 1, 0, 0) \wedge \beta_1(A_{1,5}, 1, 0, 1) \wedge \beta_1(A_{1,6}, 1, 1, 0) \wedge \\
& \beta_1(A_{1,7}, 1, 1, 1) \wedge \beta_2(A_{2,1}, 0, 0, 0) \wedge \beta_2(A_{2,2}, 0, 0, 1) \wedge \\
& \beta_2(A_{2,3}, 0, 1, 0) \wedge \beta_2(A_{2,4}, 0, 1, 1) \wedge \beta_2(A_{2,5}, 1, 0, 0) \wedge \\
& \beta_2(A_{2,6}, 1, 0, 1) \wedge \beta_2(A_{2,7}, 1, 1, 0) \wedge \\
& W_1(y_1) \wedge H_1(u_1, y_1) \wedge W_2(y_2) \wedge H_2(u_2, y_2)\},
\end{aligned}
$$

where an atom of the form $\beta_i(x, y, z, w)$ stands for the conjunction of atoms $R_{i,1}(x, y) \wedge R_{i,2}(x, z) \wedge R_{i,3}(x, w)$.

Observe that $\mathcal{V}_{\mathcal{O}} \equiv \bot$ and therefore, for each $\mathcal{S}$-database D, we have that $cert^D_{q_{\mathcal{O}},\Sigma} = \mathsf{PerfRef}^D_{q_{\mathcal{O}},\Sigma}$. Note, moreover, that in all the cases where F is without universally quantified variables, i.e., a 3-CNF instance of the more general $\forall\exists$-CNF problem, in $q_{\mathcal{S}}$ there are no atoms of the form $e_i(y_i)$, and in $q_{\mathcal{O}}$ there are no atoms of the form $W_i(y_i)$ and $H_i(u_i, y_i)$. In such cases, $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ corresponds to the CQ q over $\mathcal{S}$ obtained from $q_{\mathcal{O}}$ by unfolding atoms $R_{i,j}(x, y)$ with $s_{i,j}(x, y)$. Using the same arguments provided in Lemma 5.2, it can be shown that F is satisfiable if and only if $q \sqsubseteq q_{\mathcal{S}}$, thus implying that this is a valid reduction from the 3-CNF problem to the problem of checking whether $q_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

We now address the more general case when F is a $\forall\exists$-CNF formula, and prove that F is true if and only if $q_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

"**Only-if part:**" Suppose that F is true, that is, for every truth assignment to the variables in Y, there exists a truth assignment to the variables in X that satisfies F. Note that every disjunct of $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ (totally, 2^m) corresponds to an assignment to the variables in Y depending on the choice done for unfolding the atoms $H_1(u_1, y_1), \ldots, H_m(u_m, y_m)$, where each $H_i(u_i, y_i)$ is unfolded either with the atom $zero(u_i)$ (thus, forcing $y_i = 0$), or with the atom $one(u_i)$ (thus, forcing $y_i = 1$). This implies that in each disjunct of $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ appears either the atom $e_i(0)$ or the atom $e_i(1)$, for each $i = 1, \ldots, m$.

For instance, in the running example, if the atom $H_1(u_1, y_1)$ is unfolded with $zero(u_1) \land y_1 = 0$ and the atom $H_2(u_2, y_2)$ is unfolded with $one(u_1) \land y_2 = 1$, then the disjunct of $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ obtained is the following CQ: $\{() \mid \exists u_1, u_2.\beta_1(A_{1,1}, 0, 0, 1) \land \beta_1(A_{1,2}, 0, 1, 0) \land \beta_1(A_{1,3}, 0, 1, 1) \land \beta_1(A_{1,4}, 1, 0, 0) \land \beta_1(A_{1,5}, 1, 0, 1) \land \beta_1(A_{1,6}, 1, 1, 0) \land \beta_1(A_{1,7}, 1, 1, 1) \land \beta_2(A_{2,1}, 0, 0, 0) \land \beta_2(A_{2,2}, 0, 0, 1) \land \beta_2(A_{2,3}, 0, 1, 0) \land \beta_2(A_{2,4}, 0, 1, 1) \land \beta_2(A_{2,5}, 1, 0, 0) \land \beta_2(A_{2,6}, 1, 0, 1) \land \beta_2(A_{2,7}, 1, 1, 0) \land e_1(0) \land zero(u_1) \land e_2(1) \land one(u_2)\}$, where an atom of the form $\beta_i(x, y, z, w)$ stands for the conjunction of atoms $s_{i,1}(x, y) \land s_{i,2}(x, z) \land s_{i,3}(x, w)$.

Since, however, for every possible truth assignment to the variables in Y there exists an assignment to the variables in X that satisfies F, it can be readily seen that this is equivalent to the fact that there is a homomorphism from $q_{\mathcal{S}}$ to each possible disjunct of $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$. It follows that $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \sqsubseteq q_{\mathcal{S}}$ which, due to Lemma 6.1, implies that $q_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

"**If part:**" Suppose that F is not true, that is, there exists a truth assignment to the variables in Y such that every possible truth assignment to the variables in X does not satisfy F. Let $V = \{v_1, \ldots, v_m\}$ be the assignment to the variables in $Y = \{y_1, \ldots, y_m\}$ that makes F not satisfiable. Let, moreover, q be the disjunct of $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ obtained by unfolding, for each $i = 1, \ldots, m$, the atom $H_i(u_i, y_i)$ of $q_{\mathcal{O}}$ with $zero(u_i) \land y_i = 0$ if v_i is 0, and with $one(u_i) \land y_i = 1$ otherwise (i.e., $v_i = 1$). Since F is not satisfiable when substituting variable y_i with value v_i for each $i = 1, \ldots, m$, this implies that there is no homomorphism from $q_{\mathcal{S}}$ to q. It follows that $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \not\sqsubseteq q_{\mathcal{S}}$, and, since $\mathcal{V}_{\mathcal{O}} \equiv \bot$, we also have that $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \not\sqsubseteq (q_{\mathcal{S}} \cup \mathsf{PerfRef}_{\mathcal{V}_{\mathcal{O}},\Sigma})$ which, due to Lemma 6.1, implies that $q_{\mathcal{O}}$ is not a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

As a last consideration, we observe that the same proof works even with a reduction where $\mathcal{M}$ is a pure GAV mapping but not a LAV mapping, rather than both a GAV mapping and a LAV mapping as in the above case. In particular, it is straightforward to verify that for this it is sufficient to apply the following changes at the above reduction:

- for each $i = 1, \ldots, m$, now consider e_i as a binary predicate;
- replace the second part of the mapping $\mathcal{M}$ by including, for each $i = 1, \ldots, m$, the following two pure GAV mapping assertions: $e_i(x, 0) \to W_i(x)$ and $e_i(x, 1) \to W_i(x)$;
- for each $i = 1, \ldots, m$, replace in $q_{\mathcal{S}}$ the atom $e_i(y_i)$ with the atom $e_i(u_i, y_i)$, where u_i is a fresh existential variable;
- replace in $q_{\mathcal{O}}$ the conjunction of atoms $W_i(y_i) \land H_i(u_i, y_i)$ with the atom $W_i(u_i)$.

□

Note that the result of Π_2^p-hardness already holds when both $q_{\mathcal{S}}$ and $q_{\mathcal{O}}$ are boolean CQs, and $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ is such that the ontology $\mathcal{O}$ contains no axioms and $\mathcal{M}$ is either both a GAV mapping and a LAV mapping, or a pure GAV mapping.

It is an interesting open problem to derive the computational complexity of the verification problem for sound source-to-ontology rewritings when $\mathcal{M}$ is both a pure GAV mapping and a LAV mapping.

6.2 Computation Problem

We now address the problem of computing UCQ-maximally sound source-to-ontology rewritings. Our main result is that there are many cases where UCQ-maximally sound source-to-ontology rewritings are not guaranteed to exist.

In order to illustrate the result, from the general scenario introduced in Section 3.2, we introduce two restricted ones, namely *restricted scenario for CQJFEs* and *restricted scenario for UCQJFEs*. In both such restrained scenarios, the setting for OBDM specifications is obtained from the general one by limiting the DL ontology language to *DL-Lite*$_{\mathsf{RDFS}}$ rather than *DL-Lite*$_{\mathcal{R}}$, and limiting the mapping language to follow the pure GAV approach rather than the GLAV approach.

The difference between the two restricted scenarios is in the query language $\mathcal{L}_{\mathcal{S}}$ allowed for expressing data services, where in the former $\mathcal{L}_{\mathcal{S}}$ denotes the class of CQJFEs, whereas in the latter $\mathcal{L}_{\mathcal{S}}$ denotes the class of UCQJFEs.

We now show that, surprisingly, as soon as we try to expand the restricted scenario for CQJFEs either by extending its associated specific setting for OBDM specifications, or by extending the query language $\mathcal{L}_{\mathcal{S}}$ to CQs rather than CQJFEs, we lose the guarantee of the existence of UCQ-maximally sound source-to-ontology rewritings of queries over $\mathcal{S}$.

Theorem 6.2. *UCQ-maximally sound source-to-ontology rewritings of a query $q_{\mathcal{S}}$ may not exist if we extend the restricted scenario for CQJFEs with one of the following features:*

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

Proof. 1 This case already follows from the proof of Corollary 3.3. We now prove a stronger version, where only one existential variable occurs in the body of the CQ $q_{\mathcal{S}}$. Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the following OBDM specification of the restricted setting:

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

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

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

Let $q_{\mathcal{S}}$ be the following boolean CQ over $\mathcal{S}$: $q_{\mathcal{S}} = \{() \mid \exists y. s_1(y) \wedge s_2(y)\}$.

Observe that the CQ $q'_{\mathcal{O}} = \{() \mid \exists y_1, y_2. A_1(y_1) \wedge A_2(y_2)\}$ is not a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, because the query $q'_{\mathcal{S}} = \{() \mid \exists y_1, y_2. s_1(y_1) \wedge s_2(y_2) \wedge s_4(y_2)\}$ is a disjunct of $\mathsf{PerfRef}_{q'_{\mathcal{O}},\Sigma}$ (in fact, the only one) such that $q'_{\mathcal{S}} \not\sqsubseteq q_{\mathcal{S}}$ (cf. Lemma 6.1).

In order to continue the proof, we now introduce a pattern for an infinite number of CQs over $\mathcal{O}$ and related technical lemmata. Specifically, for every $i \geq 0$, let $q^i_{\mathcal{O}}$ be the following CQ over $\mathcal{O}$:

- if $i = 0$, then
$$q^0_{\mathcal{O}} = \{() \mid \exists y_0. A_1(y_0) \wedge A_2(y_0)\}.$$

- if $i \geq 1$, then
$$q^i_{\mathcal{O}} = \{() \mid \exists y_0, \ldots, y_i. A_1(y_0) \wedge (\bigwedge_{j=0}^{j=i-1} P(y_j, y_{j+1})) \wedge A_2(y_i)\}.$$

Lemma 6.2. *For every $i \geq 0$, we have that $q^i_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.*

Proof. As for $q^0_{\mathcal{O}}$, its perfect $\mathcal{O}$-to-$\mathcal{S}$ Σ-rewriting is the CQ $\mathsf{PerfRef}_{q^0_{\mathcal{O}},\Sigma} = \{() \mid \exists y_0. s_1(y_0) \wedge s_2(y_0) \wedge s_4(y_0)\}$, which is clearly contained in $q_{\mathcal{S}}$. Thus, due to Lemma 6.1, we can conclude that $q^0_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

Consider now $q^i_{\mathcal{O}}$, for each $i \geq 0$. Observe that $\mathsf{PerfRef}_{q^i_{\mathcal{O}},\Sigma}$ is a union of CQs, where the body of each CQ contains: (i) the atom $s_1(y_0)$ originating from $A_1(y_0)$; (ii) the conjunction of atoms $s_2(y_i) \wedge s_4(y_i)$ originating from $A_2(y_i)$; and (iii) for every $j \in [0, i-1]$, either the conjunction of atoms $s_2(y_j) \wedge s_3(y_j, y_{j+1})$ or the conjunction of atoms $s_1(y_{j+1}) \wedge s_5(y_j, y_{j+1})$ originating from $P(y_j, y_{j+1})$ using the mapping assertions m_2 and m_3, respectively. It follows that each disjunct of $\mathsf{PerfRef}_{q^i_{\mathcal{O}},\Sigma}$ contains the conjunction of atoms $s_1(y_k) \wedge s_2(y_k)$, for at least one $k \in [0, i]$. This implies that each disjunct of $\mathsf{PerfRef}_{q^i_{\mathcal{O}},\Sigma}$ is contained in $q_{\mathcal{S}}$, and therefore $\mathsf{PerfRef}_{q^i_{\mathcal{O}},\Sigma} \sqsubseteq q_{\mathcal{S}}$. Thus, due to Lemma 6.1, we can conclude that $q^i_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. $\square$

Lemma 6.3. *For every pair of natural numbers $i, k \geq 0$ with $i \neq k$, we have that both $cert_{q^i_{\mathcal{O}},\Sigma} \not\sqsubseteq cert_{q^k_{\mathcal{O}},\Sigma}$ and $cert_{q^k_{\mathcal{O}},\Sigma} \not\sqsubseteq cert_{q^i_{\mathcal{O}},\Sigma}$ hold.*

Proof. Let $i, k > 0$ (the case in which either i or k is 0 can be proven analogously) be any pair of natural numbers such that $i \neq k$, and consider the CQs $q^i_{\mathcal{O}}$ and $q^k_{\mathcal{O}}$. To prove the claim, it is sufficient to exhibit a disjunct $q^i_{\mathcal{S}}$ of $\mathsf{PerfRef}_{q^i_{\mathcal{O}},\Sigma}$ and a disjunct $q^k_{\mathcal{S}}$ of $\mathsf{PerfRef}_{q^k_{\mathcal{O}},\Sigma}$ such that both $q^i_{\mathcal{S}} \not\sqsubseteq \mathsf{PerfRef}_{q^k_{\mathcal{O}},\Sigma}$ and $q^k_{\mathcal{S}} \not\sqsubseteq \mathsf{PerfRef}_{q^i_{\mathcal{O}},\Sigma}$ hold. Let $q^i_{\mathcal{S}}$ (respectively, $q^k_{\mathcal{S}}$) be the disjunct of $\mathsf{PerfRef}_{q^i_{\mathcal{O}},\Sigma}$ (respectively, $\mathsf{PerfRef}_{q^k_{\mathcal{O}},\Sigma}$) obtained by unfolding the atom $A_1(y_0)$ with $s_1(y_0)$, the atom $A_2(y_i)$ (respectively, $A_2(y_k)$)

with $s_2(y_i) \wedge s_4(y_i)$ (respectively, $s_2(y_k) \wedge s_4(y_k)$), and all the atoms $P(y_j, y_{j+1})$, for $j \in [0, i-1]$ (respectively, $j \in [0, k-1]$), with $s_2(y_j) \wedge s_3(y_j, y_{j+1})$. It is immediate to see that each possible other disjunct $q_{\mathcal{S}}^{k'}$ (respectively, $q_{\mathcal{S}}^{i'}$) of $\mathsf{PerfRef}_{q_{\mathcal{O}}^k,\Sigma}$ (respectively, $\mathsf{PerfRef}_{q_{\mathcal{O}}^i,\Sigma}$) different from $q_{\mathcal{S}}^k$ (respectively, $q_{\mathcal{S}}^i$) is such that $q_{\mathcal{S}}^i \not\sqsubseteq q_{\mathcal{S}}^{k'}$ (respectively, $q_{\mathcal{S}}^k \not\sqsubseteq q_{\mathcal{S}}^{i'}$), because $q_{\mathcal{S}}^{k'}$ (respectively, $q_{\mathcal{S}}^{i'}$) contains at least an atom with s_5 as predicate name, whereas $q_{\mathcal{S}}^i$ (respectively, $q_{\mathcal{S}}^k$) does not.

Thus, in order to prove that $q_{\mathcal{S}}^i \not\sqsubseteq \mathsf{PerfRef}_{q_{\mathcal{O}}^k,\Sigma}$ (respectively, $q_{\mathcal{S}}^k \not\sqsubseteq \mathsf{PerfRef}_{q_{\mathcal{O}}^i,\Sigma}$) hold, it is enough to show that $q_{\mathcal{S}}^i \not\sqsubseteq q_{\mathcal{S}}^k$ (respectively, $q_{\mathcal{S}}^k \not\sqsubseteq q_{\mathcal{S}}^i$) hold. Consider the disjuncts $q_{\mathcal{S}}^i$ and $q_{\mathcal{S}}^k$ obtained as described above, that is:

$$q_{\mathcal{S}}^i = \{() \mid \exists y_0, \ldots, y_i . s_1(y_0) \wedge (\bigwedge_{j=0}^{j=i-1} s_2(y_j) \wedge s_3(y_j, y_{j+1})) \wedge s_2(y_i) \wedge s_4(y_i)\}$$

$$q_{\mathcal{S}}^k = \{() \mid \exists y_0, \ldots, y_k . s_1(y_0) \wedge (\bigwedge_{j=0}^{j=k-1} s_2(y_j) \wedge s_3(y_j, y_{j+1})) \wedge s_2(y_k) \wedge s_4(y_k)\}$$

Let $k > i$ (the case $i > k$ is specular). Since (i) both queries are rooted at $s_1(y_0)$; and (ii) in $q_{\mathcal{S}}^i$ there is the conjunction of atoms $s_2(y_i) \wedge s_4(y_i)$ whereas in $q_{\mathcal{S}}^k$ there is $s_2(y_i)$ but not $s_4(y_i)$, one can easily verify that this implies $q_{\mathcal{S}}^k \not\sqsubseteq q_{\mathcal{S}}^i$. On the other hand, since (i) both queries are rooted at $s_1(y_0)$; and (ii) in $q_{\mathcal{S}}^k$ there is the conjunction of atoms $s_3(y_i, y_{i+1}) \wedge s_2(y_{i+1})$ which is not occurring in $q_{\mathcal{S}}^i$, one can easily verify that this implies $q_{\mathcal{S}}^i \not\sqsubseteq q_{\mathcal{S}}^k$. It follows that both $q_{\mathcal{S}}^k \not\sqsubseteq q_{\mathcal{S}}^i$ and $q_{\mathcal{S}}^i \not\sqsubseteq q_{\mathcal{S}}^k$ hold, as required. $\square$

Before going further, we introduce a fair assumption. In what follows, without loss of generality, when we say that a CQ $q'_{\mathcal{O}}$ over $\mathcal{O}$ is (i) a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, and (ii) the body of $q'_{\mathcal{O}}$ is a conjunction of n atoms, we implicitly assume that all the n atoms occurring in the body of $q'_{\mathcal{O}}$ are *relevant*, i.e., if we remove any atom from the body of $q'_{\mathcal{O}}$, then we obtain a CQ that is not a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. Clearly, when seeking for (U)CQ-maximally sound source-to-ontology rewritings of queries over the source schema, one can always limit the attention to only (unions of) CQs whose body is the conjunction of relevant atoms. Furthermore, for the given Σ and $q_{\mathcal{S}}$, we can limit the attention to only CQs with no constants in their body.

Lemma 6.4. *For every $n \geq 2$, if a CQ $q'_{\mathcal{O}}$ over $\mathcal{O}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ and the body of $q'_{\mathcal{O}}$ is the conjunction of n atoms, then $q'_{\mathcal{O}} \equiv q_{\mathcal{O}}^i$, where $i = n-2$.*

Proof. We prove the claim by induction on the number of atoms n.

Base step ($n = 2$): Let $q'_{\mathcal{O}}$ be a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ whose body is the conjunction of only two atoms. Since $q'_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, it must be the case that each disjunct q' of $\mathsf{PerfRef}_{q'_{\mathcal{O}},\Sigma}$ is such that $q' \sqsubseteq q_{\mathcal{S}}$. This implies that in the body of each q' there must be at least a conjunction of atoms of the form $s_1(y') \wedge s_2(y')$ for some existential variable y'. Furthermore, since by assumption the body of $q'_{\mathcal{S}}$ is the conjunction of only two atoms, by inspecting the mapping assertions in $\mathcal{M}$, one can easily verify that the only possibility for $q'_{\mathcal{O}}$ to be a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ is that $q'_{\mathcal{O}} = \{() \mid \exists y' . A_1(y') \wedge A_2(y')\}$ for some existential variable y'. It follows that $q'_{\mathcal{O}} \equiv q_{\mathcal{O}}^0$, as required.

Inductive step: Let $q'_{\mathcal{O}}$ be a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ whose body is the conjunction of n atoms. We now prove that $q'_{\mathcal{O}} \equiv q^i_{\mathcal{O}}$, where $i = n-2$. To start, let $\overline{q_{\mathcal{O}}}$ be the CQ obtained from $q'_{\mathcal{O}}$ by removing one atom F from the body of $q'_{\mathcal{O}}$. Due to the fair assumption that the body $q'_{\mathcal{O}}$ is the conjunction of only relevant atoms, we get that $\overline{q_{\mathcal{O}}}$ is not a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. Let the removed atom be of the form $F = P(z, z')$ (the cases $F = A(z)$ and $F = B(z')$ are easier and can be proven following a similar line of reasoning) for some existential variables z, z' not necessarily distinct, and consider the two unfoldings of the atom $P(z, z')$, namely $s_2(z) \wedge s_3(z, z')$ and $s_1(z') \wedge s_5(z, z')$. Since $q'_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ whereas $\overline{q_{\mathcal{O}}}$ is not, by construction we have that each possible disjunct of both $\mathsf{PerfRef}_{q'_{\mathcal{O}},\Sigma}$ and $\mathsf{PerfRef}_{\overline{q_{\mathcal{O}}},\Sigma}$ contains both the atom $s_1(z)$ and the atom $s_2(z')$ (if not, we easily get a contradiction on the fact that $q'_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$). Consider now the following two queries:

- Let $q^l_{\mathcal{O}}$ be the CQ obtained from $q'_{\mathcal{O}}$ by (i) removing the atom $A_2(z'')$ (for some existential variable z'', such atom must occur in the body of $q'_{\mathcal{O}}$, otherwise we easily get a contradiction on the fact that $q'_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$), and (ii) replacing the above discussed atom $P(z, z')$ with the atom $A_2(z)$.
- Let $q^r_{\mathcal{O}}$ be the CQ obtained from $q'_{\mathcal{O}}$ by (i) removing the atom $A_1(z'')$ (for some existential variable z'', such atom must occur in the body of $q'_{\mathcal{O}}$, otherwise we easily get a contradiction on the fact that $q'_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$), and (ii) replacing the above discussed atom $P(z, z')$ with the atom $A_1(z')$.

Since as discussed above each disjunct of $\mathsf{PerfRef}_{\overline{q_{\mathcal{O}}},\Sigma}$ contains the atom $s_1(z)$ (respectively, $s_2(z')$), we get that $q^l_{\mathcal{O}}$ (respectively, $q^r_{\mathcal{O}}$) is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ because the unfolding of the atom $A_2(z)$ (respectively, $A_1(z')$) is $s_2(z) \wedge s_4(z)$ (respectively, $s_1(z')$), and therefore there is the join $s_1(z) \wedge s_2(z)$ (respectively, $s_1(z') \wedge s_2(z')$) in each disjunct of $\mathsf{PerfRef}_{q^l_{\mathcal{O}},\Sigma}$ (respectively, $\mathsf{PerfRef}_{q^r_{\mathcal{O}},\Sigma}$).

Notice, however, that both $q^l_{\mathcal{O}}$ and $q^r_{\mathcal{O}}$ are CQs over $\mathcal{O}$ whose body is the conjunction of at most $n-1$ atoms. Since they are both sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewritings of $q_{\mathcal{S}}$, by the inductive hypothesis, we derive that

$$q^l_{\mathcal{O}} \equiv \{() \mid \exists y^l_0, \ldots, y^l_{k+1}, z. A_1(y^l_0) \wedge (\bigwedge_{j=0}^{j=k} P(y^l_j, y^l_{j+1})) \wedge P(y^l_{j+1}, z) \wedge A_2(z)\};$$

$$q^r_{\mathcal{O}} \equiv \{() \mid \exists z', y^r_0, \ldots, y^r_{m+1}. A_1(z') \wedge P(z', y^r_0) \wedge (\bigwedge_{j=0}^{j=m} P(y^r_0, y^r_{j+1})) \wedge A_2(y^r_{m+1})\},$$

where $0 \leq k, m \leq i-1$ (we recall that $i = n-2$). By conjoining $q^l_{\mathcal{O}}$ with $q^r_{\mathcal{O}}$ and replacing the conjunction $A_2(z) \wedge A_1(z')$ with the original atom $P(z, z')$, we get that

$$q'_{\mathcal{O}} \equiv \{() \mid \exists y^l_0, \ldots, y^l_{k+1}, z, z', y^r_0, \ldots, y^r_{m+1}. A_1(y^l_0) \wedge (\bigwedge_{j=0}^{j=k} P(y^l_j, y^l_{j+1})) \wedge P(y^l_{j+1}, z) \wedge$$

$$\wedge P(z, z') \wedge P(z', y^r_0) \wedge (\bigwedge_{j=0}^{j=m} P(y^r_0, y^r_{j+1})) \wedge A_2(y^r_{m+1})\},$$

where, since by assumption the body of $q'_{\mathcal{O}}$ is the conjunction of n relevant atoms, one can easily verify that $q'_{\mathcal{O}} \equiv q^i_{\mathcal{O}}$, as required. □

Using the above lemmata, we are now able to prove that no UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ exists. Suppose, for the sake of contradiction, the existence of a UCQ $q'_{\mathcal{O}} = {q'_{\mathcal{O}}}^1 \cup \ldots \cup {q'_{\mathcal{O}}}^n$ which is a UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. For every $j \in [1,n]$, let k^j be the number of atoms occurring in the body of the CQ ${q'_{\mathcal{O}}}^j$. By Lemma 6.4, we derive that ${q'_{\mathcal{O}}}^j \equiv q_{\mathcal{O}}^{k_j-2}$, for each $j \in [1,n]$. It follows that $q'_{\mathcal{O}} \equiv q''_{\mathcal{O}}$, where $q''_{\mathcal{O}} = \bigcup_{j\in[1,n]} q_{\mathcal{O}}^{k_j-2}$. Consider now the query $q'''_{\mathcal{O}} = q''_{\mathcal{O}} \cup q^l_{\mathcal{O}}$, where $l \geq 0$ is an arbitrary number such that $l \neq k_j - 2$ for each $j \in [1,n]$. Observe that, by Lemma 6.3, we have that $cert_{q^l_{\mathcal{O}},\Sigma} \not\sqsubseteq cert_{q_{\mathcal{O}}^{k_j-2},\Sigma}$, for each $j \in [1,n]$. This implies that $cert_{q''_{\mathcal{O}},\Sigma} \sqsubset cert_{q'''_{\mathcal{O}},\Sigma}$, and therefore, since $q'_{\mathcal{O}} \equiv q''_{\mathcal{O}}$, we also have that $cert_{q'_{\mathcal{O}},\Sigma} \sqsubset cert_{q'''_{\mathcal{O}},\Sigma}$. Furthermore, since by Lemma 6.2 each disjunct of $q'''_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, we can finally conclude that $q'''_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ such that $cert_{q'_{\mathcal{O}},\Sigma} \sqsubset cert_{q'''_{\mathcal{O}},\Sigma}$. Thus, following Definition 3.4, this is clearly a contradiction on the fact that $q'_{\mathcal{O}}$ is a UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, as required.

For a more accurate characterisation about the illustrated OBDM specification Σ and CQ $q_{\mathcal{S}}$, let us define an *infinite* union of CQs $q_{\mathcal{O}}$ over $\mathcal{O}$ as follows:

$$q_{\mathcal{O}} = \bigcup_{i\geq 0} q^i_{\mathcal{O}}.$$

From the foregoing lemmata and considerations, one can immediately derive that $q_{\mathcal{O}}$ is a maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of Datalog queries. $\square$

2 Consider the following OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, where:

- $\mathcal{O} = \{\ A_1 \sqsubseteq \neg A_1, A_2 \sqsubseteq \neg A_2\ \}$
- $\mathcal{S} = \{\ s_1, s_2, s_3, s_4\ \}$
- $\mathcal{M} = \{\ m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8\ \}$, where:

$$\begin{array}{lrcl}
m_1: & s_1(x_1,x_2) & \to & P_1(x_1,x_2),\\
m_2: & s_2(x_1,x_2) & \to & P_1(x_1,x_2),\\
m_3: & s_2(x_1,x_2) & \to & P_2(x_1,x_2),\\
m_4: & s_3(x_1,x_2) & \to & P_2(x_1,x_2),\\
m_5: & s_3(x_1,x_2) & \to & P_3(x_1,x_2),\\
m_6: & s_4(x_1,x_2) & \to & P_3(x_1,x_2),\\
m_7: & \exists y_1,y_2. s_2(y_1,x) \wedge s_3(x,y_2) & \to & A_1(x),\\
m_8: & \exists y_1,y_2. s_2(x,y_1) \wedge s_4(y_2,x) & \to & A_2(x).
\end{array}$$

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

Since $\mathcal{O}$ contains the disjointness assertions $A_1 \sqsubseteq \neg A_1$ and $A_2 \sqsubseteq \neg A_2$, the violation query for $\mathcal{O}$ is $\mathcal{V}_{\mathcal{O}} = \{() \mid \exists y. A_1(y)\} \cup \{() \mid \exists y. A_2(y)\}$, and therefore $\mathcal{V}^2_{\mathcal{O}} = \{(x_1,x_2) \mid \exists y. A_1(y) \wedge \top(x_1,x_2)\} \cup \{() \mid \exists y. A_2(y) \wedge \top(x_1,x_2)\}$. By looking at the mapping assertions m_7 and m_8 occurring in $\mathcal{M}$, this means that each query of arity 2 over $\mathcal{S}$ that has a join either between the second component of s_2 and the first component of s_3, or between the first component of s_2 and the second component of s_4 is contained in $\mathsf{PerfRef}_{\mathcal{V}^2_{\mathcal{O}},\Sigma}$.

Observe that the CQ $q'_{\mathcal{O}} = \{(x_1, x_2) \mid P_1(x_1, x_2)\}$ is not a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, because the query $q'_{\mathcal{S}} = \{(x_1, x_2) \mid s_2(x_1, x_2)\}$ is a disjunct of $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ such that $q'_{\mathcal{S}} \not\sqsubseteq (q_{\mathcal{S}} \cup \mathsf{PerfRef}_{\mathcal{V}^2_{\mathcal{O}},\Sigma})$ (cf. Lemma 6.1).

In order to continue the proof, we now introduce a pattern for an infinite number of CQs over $\mathcal{O}$ and related technical lemmata. Specifically, for every $i \geq 0$, let $q^i_{\mathcal{O}}$ be the following CQ over $\mathcal{O}$:

- if $i = 0$, then $q^0_{\mathcal{O}}$ is the following CQ:

$$\{(x_1, x_2) \mid P_1(x_1, x_2) \wedge P_3(x_2, x_1)\}.$$

- if $i = 1$, then $q^1_{\mathcal{O}}$ is the following CQ:

$$\{(x_1, x_2) \mid \exists y_1.P_1(x_1, x_2) \wedge P_2(x_2, y_1) \wedge P_3(y_1, x_1)\}.$$

- if $i \geq 2$, then $q^i_{\mathcal{O}}$ is the following CQ:

$$\{(x_1, x_2) \mid \exists y_1, \ldots, y_i.P_1(x_1, x_2) \wedge P_2(x_2, y_1) \wedge (\bigwedge_{j=1}^{j=i-1} P_2(y_j, y_{j+1})) \wedge P_3(y_i, x_1)\}.$$

Lemma 6.5. *For every $i \geq 0$, we have that $q^i_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.*

Proof. Due to Lemma 6.1, it is enough to show that $\mathsf{PerfRef}_{q^i_{\mathcal{O}},\Sigma} \sqsubseteq (q_{\mathcal{S}} \cup \mathsf{PerfRef}_{\mathcal{V}^2_{\mathcal{O}},\Sigma})$, i.e., each disjunct $q'_{\mathcal{S}}$ of $\mathsf{PerfRef}_{q^i_{\mathcal{O}},\Sigma}$ is such that either $q'_{\mathcal{S}} \sqsubseteq q_{\mathcal{S}}$ or $q'_{\mathcal{S}} \sqsubseteq \mathsf{PerfRef}_{\mathcal{V}^2_{\mathcal{O}},\Sigma}$, for every $i \geq 0$.

For each $i \geq 0$, the body of $q^i_{\mathcal{O}}$ contains the atom $P_1(x_1, x_2)$ which, by looking at the mapping assertions in $\mathcal{M}$, can be unfolded with either the atom $s_1(x_1, x_2)$ or the atom $s_2(x_1, x_2)$. Half of the disjuncts of $\mathsf{PerfRef}_{q^i_{\mathcal{O}},\Sigma}$ (the ones in which $P_1(x_1, x_2)$ is unfolded with $s_1(x_1, x_2)$) are therefore easily contained in the CQJFE $q_{\mathcal{S}} = \{(x_1, x_2) \mid s_1(x_1, x_2)\}$. To prove that $\mathsf{PerfRef}_{q^i_{\mathcal{O}},\Sigma} \sqsubseteq (q_{\mathcal{S}} \cup \mathsf{PerfRef}_{\mathcal{V}^2_{\mathcal{O}},\Sigma})$, we now show that the other half of the disjuncts of $\mathsf{PerfRef}_{q^i_{\mathcal{O}},\Sigma}$ (the ones in which the atom $P_1(x_1, x_2)$ is unfolded with $s_2(x_1, x_2)$) are contained in $\mathsf{PerfRef}_{\mathcal{V}^2_{\mathcal{O}},\Sigma}$.

As for $q^0_{\mathcal{O}}$, consider the disjuncts of $\mathsf{PerfRef}_{q^0_{\mathcal{O}},\Sigma}$ in which the atom $P_1(x_1, x_2)$ is unfolded with the atom $s_2(x_1, x_2)$, namely ${q'_{\mathcal{S}}}^1 = \{(x_1, x_2) \mid s_2(x_1, x_2) \wedge s_3(x_2, x_1)\}$ and ${q'_{\mathcal{S}}}^2 = \{(x_1, x_2) \mid s_2(x_1, x_2) \wedge s_4(x_2, x_1)\}$. Since in ${q'_{\mathcal{S}}}^1$ there is the join between the second component of s_2 and the first component of s_3, and since in ${q'_{\mathcal{S}}}^2$ there is the join between the first component of s_2 and the second component of s_4, we derive that they are both contained in $\mathsf{PerfRef}_{\mathcal{V}^2_{\mathcal{O}},\Sigma}$, as required.

As for $q^1_{\mathcal{O}}$, consider the disjuncts of $\mathsf{PerfRef}_{q^1_{\mathcal{O}},\Sigma}$ in which the atom $P_1(x_1, x_2)$ is unfolded with the atom $s_2(x_1, x_2)$, namely ${q'_{\mathcal{S}}}^1 = \{(x_1, x_2) \mid \exists y_1.s_2(x_1, x_2) \wedge s_2(x_2, y_1) \wedge s_3(y_1, x_1)\}$, ${q'_{\mathcal{S}}}^2 = \{(x_1, x_2) \mid \exists y_1.s_2(x_1, x_2) \wedge s_2(x_2, y_1) \wedge s_4(y_1, x_1)\}$, ${q'_{\mathcal{S}}}^3 = \{(x_1, x_2) \mid \exists y_1.s_2(x_1, x_2) \wedge s_3(x_2, y_1) \wedge s_3(y_1, x_1)\}$ and ${q'_{\mathcal{S}}}^4 = \{(x_1, x_2) \mid \exists y_1.s_2(x_1, x_2) \wedge s_3(x_2, y_1) \wedge s_4(y_1, x_1)\}$. One can verify that in each disjunct ${q'_{\mathcal{S}}}^i$, for $i \in [1, 4]$, there is the join either between the second component of s_2 and the first component of s_3, or between the first component of s_2 and the second component of s_4, and therefore they are contained in $\mathsf{PerfRef}_{\mathcal{V}^2_{\mathcal{O}},\Sigma}$, as required.

Let us now consider $q^i_{\mathcal{O}} = \{(x_1, x_2) \mid \exists y_1, \ldots, y_i . P_1(x_1, x_2) \wedge P_2(x_2, y_1) \wedge (\bigwedge_{j=1}^{j=i-1} P_2(y_j, y_{j+1})) \wedge P_3(y_i, x_1)\}$, for any $i \geq 2$, and its disjuncts $\mathsf{PerfRef}_{q^i_{\mathcal{O}},\Sigma}$ over $\mathcal{S}$ in which the atom $P_1(x_1, x_2)$ is unfolded with the atom $s_2(x_1, x_2)$. Consider first all the possible disjuncts where at least an atom of the form $P_2(z, z')$ occurring in $q^i_{\mathcal{O}}$ is unfolded with the atom $s_3(z, z')$. We now prove that they are all contained in $\mathsf{PerfRef}_{\mathcal{V}^2_{\mathcal{O}},\Sigma}$. If $P_2(x_1, y_1)$ is unfolded with the atom $s_3(x_2, y_1)$, then there is the presence of $s_2(x_1, x_2) \wedge s_3(x_2, y_1)$ (i.e., a join between the second component of s_2 and the first component of s_3). Analogously, let the atom $P_2(x_2, y_1)$ be unfolded with the atom $s_2(x_2, y_1)$ but the atom $P_2(y_1, y_2)$ be unfolded with the atom $s_3(y_1, y_2)$. Again, there is a join between the second component of s_2 and the first component of s_3, this time on the variable y_1. Now, let $k \in [2, i-1]$ be the number such that the atom $P_2(y_k, y_{k+1})$ is unfolded with the atom $s_3(y_k, y_{k+1})$ but all the atoms $P_2(y_l, y_{l+1})$, for $l < k$, are unfolded with $s_2(y_l, y_{l+1})$. Once again, there is a join between the second component of s_2 and the first component of s_3, this time on the variable y_k.

To conclude the proof, it remains to address the case of the other disjuncts in which all the atoms of the form $P_2(z, z')$ occurring in $q^i_{\mathcal{O}}$ are unfolded with $s_3(z, z')$. In particular, note that such disjuncts contain both the atom $s_2(x_1, x_2)$ (obtained from $P_1(x_1, x_2)$) and the atom $s_2(y_{i-1}, y_i)$ (obtained from $P_2(y_{i-1}, y_i)$). There are two possible cases: either the atom $P_3(y_i, x_1)$ is unfolded with $s_3(y_i, x_1)$ or with $s_4(y_i, x_1)$. In the former case, there is a join between the second component of s_2 and the first component of s_3, this time on the variable y_i. In the latter case, there is a join between the first component of s_2 and the second component of s_4. Thus, we can conclude that all the disjuncts of $q^i_{\mathcal{O}}$ in which the atom $P_1(x_1, x_2)$ is unfolded with the atom $s_2(x_1, x_2)$ are contained in $\mathsf{PerfRef}_{\mathcal{V}^2_{\mathcal{O}},\Sigma}$, as required. □

Lemma 6.6. *For every pair of natural numbers $i, k \geq 0$ with $i \neq k$, we have that both $cert_{q^i_{\mathcal{O}},\Sigma} \not\sqsubseteq cert_{q^k_{\mathcal{O}},\Sigma}$ and $cert_{q^k_{\mathcal{O}},\Sigma} \not\sqsubseteq cert_{q^i_{\mathcal{O}},\Sigma}$ hold.*

Proof. Let $i, k \geq 2$ (the case in which either i or k is less than 2 can be proven analogously) be any pair of natural numbers such that $i \neq k$, and consider the CQs $q^i_{\mathcal{O}}$ and $q^k_{\mathcal{O}}$. To prove the claim, it is sufficient to exhibit a disjunct $q^i_{\mathcal{S}}$ of $\mathsf{PerfRef}_{q^i_{\mathcal{O}},\Sigma}$ and a disjunct $q^k_{\mathcal{S}}$ of $\mathsf{PerfRef}_{q^k_{\mathcal{O}},\Sigma}$ such that both $q^i_{\mathcal{S}} \not\sqsubseteq (\mathsf{PerfRef}_{q^k_{\mathcal{O}},\Sigma} \cup \mathsf{PerfRef}_{\mathcal{V}^2_{\mathcal{O}},\Sigma})$ and $q^k_{\mathcal{S}} \not\sqsubseteq (\mathsf{PerfRef}_{q^i_{\mathcal{O}},\Sigma} \cup \mathsf{PerfRef}_{\mathcal{V}^2_{\mathcal{O}},\Sigma})$ hold. Let $q^i_{\mathcal{S}}$ (respectively, $q^k_{\mathcal{S}}$) be the disjunct obtained by unfolding the atom $P_1(x_1, x_2)$ with $s_1(x_1, x_2)$, the atom $P_3(y_i, x_1)$ (respectively, $P_3(y_k, x_1)$) with $s_3(y_i, x_1)$ (respectively, $s_3(y_k, x_1)$), and all the atoms $P(y_j, y_{j+1})$, for $j \in [1, i-1]$ (respectively, $j \in [1, k-1]$), with $s_3(y_j, y_{j+1})$. Some immediate observations follow: (i) both $q^i_{\mathcal{S}} \not\sqsubseteq \mathsf{PerfRef}_{\mathcal{V}^2_{\mathcal{O}},\Sigma}$ and $q^k_{\mathcal{S}} \not\sqsubseteq \mathsf{PerfRef}_{\mathcal{V}^2_{\mathcal{O}},\Sigma}$ hold; (ii) all the disjuncts $q^{k'}_{\mathcal{S}}$ (respectively, $q^{i'}_{\mathcal{S}}$) of $\mathsf{PerfRef}_{q^k_{\mathcal{O}},\Sigma}$ (respectively, $\mathsf{PerfRef}_{q^i_{\mathcal{O}},\Sigma}$) obtained by unfolding the atom $P_1(x_1, x_2)$ with $s_2(x_1, x_2)$ are such that $q^i_{\mathcal{S}} \not\sqsubseteq q^{k'}_{\mathcal{S}}$ (respectively, $q^k_{\mathcal{S}} \not\sqsubseteq q^{i'}_{\mathcal{S}}$), because $q^{k'}_{\mathcal{S}}$ (respectively, $q^{i'}_{\mathcal{S}}$) contains the atom $s_2(x_1, x_2)$, whereas $q^i_{\mathcal{S}}$ (respectively, $q^k_{\mathcal{S}}$) does not; (iii) all the disjuncts $q^{k'}_{\mathcal{S}}$ (respectively, $q^{i'}_{\mathcal{S}}$) of $\mathsf{PerfRef}_{q^k_{\mathcal{O}},\Sigma}$ (respectively, $\mathsf{PerfRef}_{q^i_{\mathcal{O}},\Sigma}$) obtained by unfolding the atom $P_3(y_k, x_1)$ (respectively, $P_3(y_i, x_1)$) with $s_2(y_k, x_1)$ (respectively, $s_2(y_i, x_1)$) are such that $q^i_{\mathcal{S}} \not\sqsubseteq q^{k'}_{\mathcal{S}}$ (respectively, $q^k_{\mathcal{S}} \not\sqsubseteq q^{i'}_{\mathcal{S}}$), because $q^{k'}_{\mathcal{S}}$ (respectively, $q^{i'}_{\mathcal{S}}$) contains the atom $s_2(y_k, x_1)$ (respectively, $s_2(y_i, x_1)$), whereas $q^i_{\mathcal{S}}$ (respectively, $q^k_{\mathcal{S}}$) does not; (iv) all the disjuncts $q^{k'}_{\mathcal{S}}$ (respectively, $q^{i'}_{\mathcal{S}}$) of $\mathsf{PerfRef}_{q^k_{\mathcal{O}},\Sigma}$ (respectively, $\mathsf{PerfRef}_{q^i_{\mathcal{O}},\Sigma}$) obtained by unfolding the atom $P_2(y_j, y_{j+1})$, for some $j \in [1, k-1]$ (respectively, $j \in [1, i-1]$), with $s_2(y_j, y_{j+1})$ are such that $q^i_{\mathcal{S}} \not\sqsubseteq q^{k'}_{\mathcal{S}}$ (respectively, $q^k_{\mathcal{S}} \not\sqsubseteq q^{i'}_{\mathcal{S}}$), because $q^{k'}_{\mathcal{S}}$ (respectively, $q^{i'}_{\mathcal{S}}$) contains the atom $s_2(y_j, y_{j+1})$, whereas $q^i_{\mathcal{S}}$ (respectively, $q^k_{\mathcal{S}}$) does not.

It follows that, in order to prove that $q^i_{\mathcal{S}} \not\sqsubseteq \mathsf{PerfRef}_{q^k_{\mathcal{O}},\Sigma}$ (respectively, $q^k_{\mathcal{S}} \not\sqsubseteq \mathsf{PerfRef}_{q^i_{\mathcal{O}},\Sigma}$) hold, it is enough to show that $q^i_{\mathcal{S}} \not\sqsubseteq q^k_{\mathcal{S}}$ (respectively, $q^k_{\mathcal{S}} \not\sqsubseteq q^i_{\mathcal{S}}$) hold. Consider the disjuncts $q^i_{\mathcal{S}}$ and $q^k_{\mathcal{S}}$ obtained as described above, that is:

$$q^i_{\mathcal{S}} = \{() \mid \exists y_1, \ldots, y_i . s_2(x_1, x_2) \wedge s_3(x_2, y_1) \wedge (\bigwedge_{j=1}^{j=i-1} s_2(y_j, y_{j+1})) \wedge s_3(y_i, x_1)\}$$

$$q^k_{\mathcal{S}} = \{() \mid \exists y_1, \ldots, y_k . s_2(x_1, x_2) \wedge s_3(x_2, y_1) \wedge (\bigwedge_{j=1}^{j=k-1} s_2(y_j, y_{j+1})) \wedge s_3(y_k, x_1)\}$$

One can easily verify that, if either $i > k$ or $k > i$ hold, then both $q^i_{\mathcal{S}} \not\sqsubseteq q^k_{\mathcal{S}}$ and $q^k_{\mathcal{S}} \not\sqsubseteq q^i_{\mathcal{S}}$ hold, as required. $\square$

Lemma 6.7. *If a CQ $q'_{\mathcal{O}}$ over $\mathcal{O}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, then $q'_{\mathcal{O}} \sqsubseteq q^i_{\mathcal{O}}$, for some $i \geq 0$.*

Proof. Let a CQ $q'_{\mathcal{O}} = \{(t_1, t_2) \mid \exists \vec{y'} . \phi(t_1, t_2, \vec{y'})\}$ be a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, where the target list (t_1, t_2) comprises two (not necessarily distinct) terms t_1 and t_2. Clearly, since $q'_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, by inspecting the mapping assertions in $\mathcal{M}$, it is easy to see that $\phi(t_1, t_2, \vec{y'})$ must contain the atom $P_1(t_1, t_2)$ (unless $\phi(t_1, t_2, \vec{y'}) \equiv \bot$, from which the claim trivially follows). Due to the mapping assertion m_2, however, there are disjuncts of $\mathsf{PerfRef}_{q'_{\mathcal{O}},\Sigma}$ that are not contained in $q_{\mathcal{S}}$, because the atom $P_1(t_1, t_2)$ can be unfolded with $s_2(t_1, t_2)$. It follows that the only possibility for $q'_{\mathcal{O}}$ to be a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting is that those disjuncts must be contained in $\mathsf{PerfRef}_{\mathcal{V}^2_{\mathcal{O}},\Sigma}$, i.e., they must have a join either between the second component of s_2 and the first component of s_3, or between the first component of s_2 and the second component of s_4.

By inspecting again the mapping, one can verify that at least an atom of the form $P_3(t_i, t_1)$, for some term t_i, must occur in $\phi(t_1, t_2, \vec{y'})$ (without loss of generality, we can assume that only one of such atom occurs in $\phi(t_1, t_2, \vec{y'})$). Indeed, if this is not the case, then the disjunct $q'_{\mathcal{S}}$ of $\mathsf{PerfRef}_{q'_{\mathcal{O}},\Sigma}$ in which the atom $P_1(t_1, t_2)$ is unfolded with $s_2(t_1, t_2)$, all the atoms of the form $P_2(z, z')$ (for any pair of terms z, z') are unfolded with $s_2(z, z')$, and all the atoms of the form $P_3(z_i, z')$ (for any pair of terms z_i, z', where $z' \neq t_1$) are unfolded with $s_4(z_i, z')$ is such that $q'_{\mathcal{S}} \not\sqsubseteq q_{\mathcal{S}} \cup \mathsf{PerfRef}_{\mathcal{V}^2_{\mathcal{O}},\Sigma}$, thus contradicting the fact that $q'_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. Observe that there are two possible cases for the atom $P_3(t_i, t_1)$: either $t_i = t_2$ or not. In the former case, we trivially have that $q'_{\mathcal{O}} \sqsubseteq q^0_{\mathcal{O}}$, as required. Thus, in what follows, we explore the latter case. In the case that there is also an atom of the form $P_2(t_2, t_i)$ in the body $\phi(t_1, t_2, \vec{y'})$ of $q'_{\mathcal{O}}$, one can easily verify that $q'_{\mathcal{O}} \sqsubseteq q^1_{\mathcal{O}}$, as required.

Let us now consider the case in which $t_i \neq t_2$ and $P_2(t_2, t_i)$ does not occur. Specifically, we prove that in the body $\phi(t_1, t_2, \vec{y'})$ of $q'_{\mathcal{O}}$ there must necessarily exist a conjunction of atoms of the form $P_2(t_2, z_1) \wedge P_2(z_1, z_2) \wedge \ldots \wedge P_2(z_{i-2}, z_{i-1}) \wedge P_2(z_{i-1}, t_i)$ for some terms $z_1, z_2, \ldots, z_{i-1}$ and for some number $i \geq 2$. Suppose, for the sake of contradiction, that such conjunction of atoms does not occur in the body of $q'_{\mathcal{O}}$, i.e., at least one of the following two conditions hold: (i) the atom $P_2(t_2, z_1)$ for some term z_1 is missing; (ii) for all the possible conjunction of atoms of the form $P_2(t_2, z_1) \wedge P_2(z_1, z_2) \wedge \ldots \wedge P_2(z_{i-2}, z_{i-1}) \wedge P_2(z_{i-2}, z_{i-1})$ for terms $z_1, z_2, \ldots, z_{i-1}$ and for a number $i \geq 2$, the atom $P_2(z_{i-1}, t_i)$ is missing in $\phi(t_1, t_2, \vec{y'})$.

If (i) holds, then consider the disjunct $q'_{\mathcal{S}}$ of $\mathsf{PerfRef}_{q'_{\mathcal{O}},\Sigma}$ obtained by unfolding the atom $P_1(t_1, t_2)$ with $s_2(t_1, t_2)$, all the atoms of the form $P_2(z, z')$ (for any pair of terms z, z' with $z \neq t_2$) with $s_3(z, z')$, and the atom $P_3(t_i, t_1)$ with $s_3(t_i, t_1)$. Obviously, we have that $q'_{\mathcal{S}} \not\sqsubseteq q_{\mathcal{S}}$ and, moreover, since $s_3(t_2, z)$ does not occur in the body of $q'_{\mathcal{S}}$ for any term z, we easily get that $q'_{\mathcal{S}} \not\sqsubseteq \mathsf{PerfRef}_{\mathcal{V}^2_{\mathcal{O}},\Sigma}$. Thus, $q'_{\mathcal{S}} \not\sqsubseteq q_{\mathcal{S}} \cup \mathsf{PerfRef}_{\mathcal{V}^2_{\mathcal{O}},\Sigma}$, which contradicts the fact that $q'_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

If (ii) holds, then consider the disjunct $q'_{\mathcal{S}}$ of $\mathsf{PerfRef}_{q'_{\mathcal{O}},\Sigma}$ obtained by unfolding the atom $P_1(t_1, t_2)$ with $s_2(t_1, t_2)$, all the possible conjunction of atoms of the form $P_2(t_2, z_1) \wedge P_2(z_1, z_2) \wedge \ldots \wedge P_2(z_{i-2}, z_{i-1})$, for terms $z_1, z_2, \ldots, z_{i-1}$ and for a number $i \geq 2$, with $s_2(t_2, z_1) \wedge s_2(z_1, z_2) \wedge \ldots \wedge s_2(z_{i-2}, z_{i-1})$, all the other possible atoms $P_2(z, z')$ (for any pair of terms z, z') not included in the previous case with $s_3(z, z')$, and the atom $P_3(t_i, t_1)$ with $s_3(t_i, t_1)$. Obviously, we have that $q'_{\mathcal{S}} \not\sqsubseteq q_{\mathcal{S}}$ and, moreover, since $s_2(z, t_i)$ does not occur in the body of $q'_{\mathcal{S}}$ for any term z, one can easily verify that $q'_{\mathcal{S}} \not\sqsubseteq \mathsf{PerfRef}_{\mathcal{V}^2_{\mathcal{O}},\Sigma}$. Thus, $q'_{\mathcal{S}} \not\sqsubseteq q_{\mathcal{S}} \cup \mathsf{PerfRef}_{\mathcal{V}^2_{\mathcal{O}},\Sigma}$, which contradicts the fact that $q'_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

As a consequence of the above considerations, we derive that the body of each $q'_{\mathcal{O}}$ that is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting must contains the conjunction of atoms $P_1(t_1, t_2) \wedge P_2(t_2, z_1) \wedge P_2(z_1, z_2) \wedge \ldots \wedge P_2(z_{i-2}, z_{i-1}) \wedge P_2(z_{i-1}, t_i) \wedge P_3(t_i, t_1)$, where (t_1, t_2) is the target list of $q'_{\mathcal{O}}$ and i is a number such that $i \geq 2$. But then, we easily get that $q'_{\mathcal{O}}$ is such that $q'_{\mathcal{O}} \sqsubseteq q^i_{\mathcal{O}}$, as required. □

Using the above lemmata, we are now able to prove that no UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ exists. Suppose, for the sake of contradiction, the existence of a UCQ $q'_{\mathcal{O}} = {q'_{\mathcal{O}}}^1 \cup \ldots \cup {q'_{\mathcal{O}}}^n$ which is a UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. For each $j \in [1, n]$, by Lemma 6.7, we know the existence of a number i_j for which ${q'_{\mathcal{O}}}^j \sqsubseteq q^{i_j}_{\mathcal{O}}$. It follows that $q'_{\mathcal{O}} \sqsubseteq q''_{\mathcal{O}}$, where $q''_{\mathcal{O}} = \bigcup_{j \in [1,n]} q^{i_j}_{\mathcal{O}}$. Consider now the query $q'''_{\mathcal{O}} = q''_{\mathcal{O}} \cup q^l_{\mathcal{O}}$, where $l \geq 0$ is an arbitrary number such that $l \neq i_j$ for each $j \in [1, n]$. Observe that, by Lemma 6.6, we have that $cert_{q^l_{\mathcal{O}},\Sigma} \not\sqsubseteq cert_{q^{i_j}_{\mathcal{O}},\Sigma}$, for each $j \in [1, n]$. This implies that $cert_{q''_{\mathcal{O}},\Sigma} \sqsubset cert_{q'''_{\mathcal{O}},\Sigma}$, and therefore, since $q'_{\mathcal{O}} \sqsubseteq q''_{\mathcal{O}}$, we also have that $cert_{q'_{\mathcal{O}},\Sigma} \sqsubset cert_{q'''_{\mathcal{O}},\Sigma}$. Furthermore, since by Lemma 6.5 each disjunct of $q'''_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, we can finally conclude that $q'''_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting such that $cert_{q'_{\mathcal{O}},\Sigma} \sqsubset cert_{q'''_{\mathcal{O}},\Sigma}$. Thus, following Definition 3.4, this is clearly a contradiction on the fact that $q'_{\mathcal{O}}$ is a UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, as required.

For a more accurate characterisation about the illustrated OBDM specification Σ and CQJFE $q_{\mathcal{S}}$, let us define an infinite union of CQs $q_{\mathcal{O}}$ over $\mathcal{O}$ as follows:

$$q_{\mathcal{O}} = \bigcup_{i \geq 0} q^i_{\mathcal{O}}.$$

From the foregoing lemmata and considerations, one can immediately derive that $q_{\mathcal{O}}$ is a maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of Datalog queries. □

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

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

$$\begin{array}{lrcl} m_1: & s_1(x_1, x_2) & \rightarrow & P(x_1, x_2), \\ m_2: & s_2(x) & \rightarrow & A(x). \end{array}$$

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

Observe that the CQ $q'_{\mathcal{O}} = \{(x) \mid \exists y . P(x, y)\}$ is not a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, because the query $q'_{\mathcal{S}} = \{(x) \mid s_2(x)\}$ is a disjunct of $\mathsf{PerfRef}_{q'_{\mathcal{O}}, \Sigma}$ (obtained by rewriting $q'_{\mathcal{O}}$ with the ontology assertion $A \sqsubseteq \exists P$, thus obtaining the CQ $\{(x) \mid A(x)\}$, and then by unfolding this latter with respect to $\mathcal{M}$) such that $q'_{\mathcal{S}} \not\sqsubseteq q_{\mathcal{S}}$.

In order to continue the proof, we now introduce two patterns for an infinite number of CQs over $\mathcal{O}$ and related technical lemmata. The first one ranges over natural numbers. Specifically, for every $i \geq 0$, let $q^i_{\mathcal{O}}$ be the following CQ over $\mathcal{O}$:

- if $i = 0$, then
$$q^0_{\mathcal{O}} = \{(x) \mid \exists y_0, y_1 . P(x, y_0) \wedge P(y_0, y_1)\}$$
- if $i \geq 1$, then
$$q^i_{\mathcal{O}} = \{(x) \mid \exists y_0, \ldots, y_{2i+1} . P(x, y_0) \wedge (\bigwedge_{i=1}^{i} P(y_{2i-1}, y_{2i-2}) \wedge P(y_{2i-1}, y_{2i})) \wedge \\ \wedge P(y_{2i}, y_{2i+1})\}$$

For instance, with $i = 3$, we have $q^3_{\mathcal{O}} = \{(x) \mid \exists y_0, \ldots, y_7 . P(x, y_0) \wedge P(y_1, y_0) \wedge P(y_1, y_2) \wedge P(y_3, y_2) \wedge P(y_3, y_4) \wedge P(y_5, y_4) \wedge P(y_5, y_6) \wedge P(y_6, y_7)\}$.

The second one ranges over natural numbers and constants $c \in \mathsf{Const}$. Specifically, for every constant $c \in \mathsf{Const}$ and for every $i \geq 0$, let $q^{c,i}_{\mathcal{O}}$ be the following CQ over $\mathcal{O}$:

- if $i = 0$, then
$$q^{c,0}_{\mathcal{O}} = \{(x) \mid P(x, c)\}$$
- if $i = 1$, then
$$q^{c,1}_{\mathcal{O}} = \{(x) \mid \exists y_0, y_1 . P(x, y_0) \wedge P(y_1, y_0) \wedge P(y_1, c)\}$$
- if $i \geq 2$, then
$$q^{c,i}_{\mathcal{O}} = \{(x) \mid \exists y_0, \ldots, y_{2i-1} . P(x, y_0) \wedge (\bigwedge_{i=1}^{i-1} P(y_{2i-1}, y_{2i-2}) \wedge P(y_{2i-1}, y_{2i})) \wedge \\ \wedge P(y_{2i-1}, y_{2i-2}) \wedge P(y_{2i-1}, c)\}$$

For instance, with $i = 3$ and any $c \in \mathsf{Const}$, we have $q^{c,3}_{\mathcal{O}} = \{(x) \mid \exists y_0, \ldots, y_7 . P(x, y_0) \wedge P(y_1, y_0) \wedge P(y_1, y_2) \wedge P(y_3, y_2) \wedge P(y_3, y_4) \wedge P(y_5, y_4) \wedge P(y_5, c)\}$.

Before delving into the technical lemmata, we introduce some considerations on the rewriting of each of these above illustrated CQs with respect to the ontology $\mathcal{O}$. Consider $q^0_{\mathcal{O}}$ (respectively, $q^{c,0}_{\mathcal{O}}$ for some constant $c \in \mathsf{Const}$). Since the existential variable y_0 is in join because of the occurrence of the atom $P(y_0, y_1)$ (respectively, the atom $P(x, c)$ has a constant in its second argument), when rewriting $q^0_{\mathcal{O}}$ (respectively, $q^{c,0}_{\mathcal{O}}$) with respect to $\mathcal{O}$ by means of the $\mathsf{PerfectRef}$ algorithm, the inclusion assertion $A \sqsubseteq \exists P$ is not applicable to the atom $P(x, y_0)$ (respectively, $P(x, c)$). Thus, $\mathsf{PerfectRef}(\mathcal{O}, q^0_{\mathcal{O}}) = q^0_{\mathcal{O}}$ (respectively, $\mathsf{PerfectRef}(\mathcal{O}, q^{c,0}_{\mathcal{O}}) = q^{c,0}_{\mathcal{O}}$).

Consider now $q^1_{\mathcal{O}}$ (respectively, $q^{c,1}_{\mathcal{O}}$ for some constant $c \in \mathsf{Const}$). Here, the existential variable y_0 of the atom $P(x, y_0)$ is in join because of the occurrence of the atom $P(y_1, y_0)$, and therefore the inclusion assertion $A \sqsubseteq \exists P$ is not applicable to the atom $P(x, y_0)$. Furthermore, the existential variable y_2 of the atom $P(y_1, y_2)$ is in join because of the occurrence of the atom $P(y_2, y_3)$ (respectively, the atom $P(y_1, c)$ has a constant in its second argument), and therefore the inclusion assertion $A \sqsubseteq \exists P$ is not applicable either to both the atoms $P(y_1, y_2)$ and $P(y_2, y_3)$ (respectively, to the atom $P(y_1, c)$). Notice, however, that when rewriting $q^1_{\mathcal{O}}$ (respectively, $q^{c,1}_{\mathcal{O}}$) with respect to $\mathcal{O}$, the $\mathsf{PerfectRef}$ algorithm produces a new CQ obtained by *unifying* $P(x, y_0)$ and $P(y_1, y_0)$, i.e., it produces a new CQ in which $x = y_1$. One can easily verify that such a new CQ is logically equivalent to $q^0_{\mathcal{O}}$ (respectively, $q^{c,0}_{\mathcal{O}}$), and therefore the inclusion assertion $A \sqsubseteq \exists P$ is not applicable to this new CQ, too. Thus, $\mathsf{PerfectRef}(\mathcal{O}, q^1_{\mathcal{O}}) = q^1_{\mathcal{O}} \cup \mathsf{PerfectRef}(\mathcal{O}, q^0_{\mathcal{O}}) = q^1_{\mathcal{O}} \cup q^0_{\mathcal{O}}$ (respectively, $\mathsf{PerfectRef}(\mathcal{O}, q^{c,1}_{\mathcal{O}}) = q^{c,1}_{\mathcal{O}} \cup \mathsf{PerfectRef}(\mathcal{O}, q^{c,0}_{\mathcal{O}}) = q^{c,1}_{\mathcal{O}} \cup q^{c,0}_{\mathcal{O}}$).

Let us consider the CQ $q^i_{\mathcal{O}}$ (respectively, $q^{c,i}_{\mathcal{O}}$ for some constant $c \in \mathsf{Const}$), for any natural number $i \geq 2$. As before, the existential variable y_0 of the atom $P(x, y_0)$ is in join because of the occurrence of the atom $P(y_1, y_0)$, and therefore the inclusion assertion $A \sqsubseteq \exists P$ is not applicable to the atom $P(x, y_0)$. Furthermore, for each even $j \in [2, 2i-2]$, the existential variable y_j is in join because occurs in the second argument of both the atom $P(y_{j-1}, y_j)$ and the atom $P(y_{j+1}, y_j)$, and therefore the inclusion assertion $A \sqsubseteq \exists P$ is not applicable either to both the atoms $P(y_{j-1}, y_j)$ and $P(y_{j+1}, y_j)$. Finally, the existential variable y_{2i} of the atom $P(y_{2i-1}, y_{2i})$ is in join because of the occurrence of the atom $P(y_{2i}, y_{2i+1})$ (respectively, the atom $P(y_{2i-1}, c)$ has a constant in its second argument), and therefore, once again, the inclusion assertion $A \sqsubseteq \exists P$ is not applicable either to both the atoms $P(y_{2i-1}, y_{2i})$ and $P(y_{2i}, y_{2i+1})$ (respectively, to the atom $P(y_{2i-1}, c)$). Note that when rewriting $q^i_{\mathcal{O}}$ (respectively, $q^{c,i}_{\mathcal{O}}$) with respect to $\mathcal{O}$, the $\mathsf{PerfectRef}$ algorithm, by means of the *unification step*, considers the following set of equalities $E = \{x = y_1, y_1 = y_3, y_3 = y_5, \dots, y_{2i-3} = y_{2i-1}\}$. Specifically, for each set of equalities e in the power set of E (i.e., for each $e \in \mathfrak{P}(E)$), a new CQ is obtained from $q^i_{\mathcal{O}}$ in which all the equalities in e are applied. One can easily verify that each CQ obtained in this way is logically equivalent to $q^{i-|e|}_{\mathcal{O}}$ (respectively, $q^{c,i-|e|}_{\mathcal{O}}$), where $|e|$ is the cardinality of e (i.e., the number of equalities in e). So, with a trivial induction on i, it can be proven that the inclusion assertion $A \sqsubseteq \exists P$ is not applicable to these new CQs, too. Thus, for any $i \geq 2$, we have $\mathsf{PerfectRef}(\mathcal{O}, q^i_{\mathcal{O}}) = q^i_{\mathcal{O}} \cup \mathsf{PerfectRef}(\mathcal{O}, q^{i-1}_{\mathcal{O}}) = q^i_{\mathcal{O}} \cup q^{i-1}_{\mathcal{O}} \cup \mathsf{PerfectRef}(\mathcal{O}, q^{i-2}_{\mathcal{O}}) = \dots = q^i_{\mathcal{O}} \cup q^{i-1}_{\mathcal{O}} \cup q^{i-2}_{\mathcal{O}} \cup \dots \cup q^0_{\mathcal{O}}$ (respectively, $\mathsf{PerfectRef}(\mathcal{O}, q^{c,i}_{\mathcal{O}}) = q^{c,i}_{\mathcal{O}} \cup \mathsf{PerfectRef}(\mathcal{O}, q^{c,i-1}_{\mathcal{O}}) = q^{c,i}_{\mathcal{O}} \cup q^{c,i-1}_{\mathcal{O}} \cup \mathsf{PerfectRef}(\mathcal{O}, q^{c,i-2}_{\mathcal{O}}) = \dots = q^{c,i}_{\mathcal{O}} \cup q^{c,i-1}_{\mathcal{O}} \cup q^{c,i-2}_{\mathcal{O}} \cup \dots \cup q^{c,0}_{\mathcal{O}}$).

Lemma 6.8. *For every $i \geq 0$ and for every constant $c \in \mathsf{Const}$, we have that both $q^i_{\mathcal{O}}$ and $q^{c,i}_{\mathcal{O}}$ are sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewritings of $q_{\mathcal{S}}$.*

Proof. Consider any $i \geq 0$ and any constant $c \in \mathsf{Const}$. From the above considerations, when rewriting the CQs $q^i_{\mathcal{O}}$ and $q^{c,i}_{\mathcal{O}}$ with respect to the ontology $\mathcal{O}$, the $\mathsf{PerfectRef}$ algorithm never applies the inclusion assertion $A \sqsubseteq \exists P$. So, each possible CQ in $\mathsf{PerfectRef}(\mathcal{O}, q^i_{\mathcal{O}})$ and each possible CQ in $\mathsf{PerfectRef}(\mathcal{O}, q^{c,i}_{\mathcal{O}})$ will contain the atom $P(x, y_0)$ (or the atom $P(x, c)$ in $\mathsf{PerfectRef}(\mathcal{O}, q^{c,0}_{\mathcal{O}})$ if $i = 0$). By construction of the mapping $\mathcal{M}$, since any atom of the form $P(z, z')$ (for any pair of not necessarily distinct terms z, z') can be unfolded only with $s_1(z, z')$, we derive that each possible CQ in $\mathsf{PerfRef}_{q^i_{\mathcal{O}},\Sigma}$ and each possible CQ in $\mathsf{PerfRef}_{q^{c,i}_{\mathcal{O}},\Sigma}$ will contain the atom $s_1(x, y_0)$ (or the atom $s_1(x, c)$ in $\mathsf{PerfRef}_{q^{c,0}_{\mathcal{O}},\mathcal{O}}$ if $i = 0$). Thus, both $\mathsf{PerfRef}_{q^i_{\mathcal{O}},\Sigma} \sqsubseteq q_{\mathcal{S}}$ and $\mathsf{PerfRef}_{q^{c,i}_{\mathcal{O}},\Sigma} \sqsubseteq q_{\mathcal{S}}$ trivially hold, and therefore, due to Lemma 6.1, we derive that both $q^i_{\mathcal{O}}$ and $q^{c,i}_{\mathcal{O}}$ are sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewritings of $q_{\mathcal{S}}$, as required. □

Lemma 6.9. *For every $i \geq 0$ and for every constant $c \in \mathsf{Const}$, we have that both $cert_{q^i_{\mathcal{O}},\Sigma} \sqsubset cert_{q^{i+1}_{\mathcal{O}},\Sigma}$ and $cert_{q^{c,i}_{\mathcal{O}},\Sigma} \sqsubset cert_{q^{c,i+1}_{\mathcal{O}},\Sigma}$ hold.*

Proof. Let $i \geq 0$ be any natural number, and consider the CQs $q^i_{\mathcal{O}}$ and $q^{i+1}_{\mathcal{O}}$ (respectively, the CQs $q^{c,i}_{\mathcal{O}}$ and $q^{c,i+1}_{\mathcal{O}}$ for any constant $c \in \mathsf{Const}$). As already observed, we have $q^i_{\mathcal{O}} \in \mathsf{PerfectRef}(\mathcal{O}, q^{i+1}_{\mathcal{O}})$ (respectively, $q^{c,i}_{\mathcal{O}} \in \mathsf{PerfectRef}(\mathcal{O}, q^{c,i+1}_{\mathcal{O}})$). So, any possible disjunct of $\mathsf{PerfRef}_{q^i_{\mathcal{O}},\Sigma}$ (respectively, $\mathsf{PerfRef}_{q^{c,i}_{\mathcal{O}},\Sigma}$) occurs also in $\mathsf{PerfRef}_{q^{i+1}_{\mathcal{O}},\Sigma}$ (respectively, $\mathsf{PerfRef}_{q^{c,i+1}_{\mathcal{O}},\Sigma}$). It follows that $cert_{q^i_{\mathcal{O}},\Sigma} \sqsubseteq cert_{q^{i+1}_{\mathcal{O}},\Sigma}$ (respectively, $cert_{q^{c,i}_{\mathcal{O}},\Sigma} \sqsubseteq cert_{q^{c,i+1}_{\mathcal{O}},\Sigma}$). In order to prove that actually $cert_{q^i_{\mathcal{O}},\Sigma} \sqsubset cert_{q^{i+1}_{\mathcal{O}},\Sigma}$ (respectively, $cert_{q^{c,i}_{\mathcal{O}},\Sigma} \sqsubset cert_{q^{c,i+1}_{\mathcal{O}},\Sigma}$) hold, it is therefore enough to exhibit a disjunct $q^{i+1}_{\mathcal{S}}$ of $\mathsf{PerfRef}_{q^{i+1}_{\mathcal{O}},\Sigma}$ (respectively, $\mathsf{PerfRef}_{q^{c,i+1}_{\mathcal{O}},\Sigma}$) for which $q^{i+1}_{\mathcal{S}} \not\sqsubseteq \mathsf{PerfRef}_{q^i_{\mathcal{O}},\Sigma}$ (respectively, $q^{i+1}_{\mathcal{S}} \not\sqsubseteq \mathsf{PerfRef}_{q^{c,i}_{\mathcal{O}},\Sigma}$). As already observed, all the possible CQs in $\mathsf{PerfectRef}(\mathcal{O}, q^i_{\mathcal{O}})$ (respectively, $\mathsf{PerfectRef}(\mathcal{O}, q^{c,i}_{\mathcal{O}})$) are obtained by applying some equalities to the variables occurring in $q^i_{\mathcal{O}}$ (respectively, $q^{c,i}_{\mathcal{O}}$).

Consider now the CQ $q^{i+1}_{\mathcal{O}}$ (respectively, $q^{c,i+1}_{\mathcal{O}}$). From the above consideration, one can easily verify that, for each possible CQ $q'_{\mathcal{O}}$ in $\mathsf{PerfectRef}(\mathcal{O}, q^i_{\mathcal{O}})$ (respectively, $\mathsf{PerfectRef}(\mathcal{O}, q^{c,i}_{\mathcal{O}})$), there is no homomorphism from $q'_{\mathcal{O}}$ to $q^{i+1}_{\mathcal{O}}$ (respectively, $q^{c,i+1}_{\mathcal{O}}$). This is because the variable x occurring in both CQs under examination is a distinguished variable, and therefore every candidate homomorphism h must have $h(x) = x$. It follows that $q^{i+1}_{\mathcal{O}} \not\sqsubseteq \mathsf{PerfectRef}(\mathcal{O}, q^i_{\mathcal{O}})$ (respectively, $q^{c,i+1}_{\mathcal{O}} \not\sqsubseteq \mathsf{PerfectRef}(\mathcal{O}, q^{c,i}_{\mathcal{O}})$).

Furthermore, as already observed, the inclusion assertion $A \sqsubseteq \exists P$ is never applied, and thus no atom with predicate name A occurs in the body of the CQs of $\mathsf{PerfectRef}(\mathcal{O}, q^{i+1}_{\mathcal{O}})$ (respectively, $\mathsf{PerfectRef}(\mathcal{O}, q^{c,i+1}_{\mathcal{O}})$). So, all the disjuncts of $\mathsf{PerfRef}_{q^{i+1}_{\mathcal{O}},\Sigma}$ (respectively, $\mathsf{PerfRef}_{q^{c,i+1}_{\mathcal{O}},\Sigma}$) are obtained from the CQs in $\mathsf{PerfectRef}(\mathcal{O}, q^{i+1}_{\mathcal{O}})$ (respectively, $\mathsf{PerfectRef}(\mathcal{O}, q^{c,i+1}_{\mathcal{O}})$) by just unfolding the atoms in their body, which amounts to replace each predicate name P with s_1.

It follows that the disjunct $q^{i+1}_{\mathcal{S}}$ of $\mathsf{PerfRef}_{q^{i+1}_{\mathcal{O}},\Sigma}$ (respectively, $\mathsf{PerfRef}_{q^{c,i+1}_{\mathcal{O}},\Sigma}$) obtained from the CQ $q^{i+1}_{\mathcal{O}}$ (respectively, $q^{c,i+1}_{\mathcal{O}}$) by unfolding each atom in its body is such that $q^{i+1}_{\mathcal{S}} \not\sqsubseteq \mathsf{PerfRef}_{q^i_{\mathcal{O}},\Sigma}$ (respectively, $q^{i+1}_{\mathcal{S}} \not\sqsubseteq \mathsf{PerfRef}_{q^{c,i}_{\mathcal{O}},\Sigma}$), as required. □

Lemma 6.10. *If a CQ $q'_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, then, for some $i \geq 0$, either $q'_{\mathcal{O}} \sqsubseteq q^{i}_{\mathcal{O}}$, or $q'_{\mathcal{O}} \sqsubseteq q^{c,i}_{\mathcal{O}}$ for some constant $c \in \mathsf{Const}$.*

Proof. Let a CQ $q'_{\mathcal{O}} = \{(t) \mid \exists \vec{y'}.\phi(t, \vec{y'})\}$ be a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, where the target list is composed of a single term t (i.e., either a distinguished variable or a constant). Clearly, since $q'_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, by inspecting the ontology $\mathcal{O}$ and the mapping $\mathcal{M}$, it is easy to see that $\phi(t, \vec{y'})$ must contain an atom of the form $P(t, t_0)$ for a term t_0 (unless $\phi(t, \vec{y'}) \equiv \bot$, from which the claim trivially follows). Without loss of generality, we can assume that only one of such atom occurs in $\phi(t, \vec{y'})$. There are three possible different cases for the term t_0: (i) t_0 is a constant; (ii) t_0 is an existential variable and $t_0 = t$; (iii) t_0 is an existential variable and $t_0 \neq t$. In case (i), we trivially have that $q'_{\mathcal{O}} \sqsubseteq q^{t_0,0}_{\mathcal{O}}$. On the other hand, in case (ii), we trivially have that $q'_{\mathcal{O}} \sqsubseteq q^{0}_{\mathcal{O}}$. So, in what follows we explore case (iii).

By induction on natural numbers $i \geq 1$, we now prove that either one among $q'_{\mathcal{O}} \sqsubseteq q^{i}_{\mathcal{O}}$ or $q'_{\mathcal{O}} \sqsubseteq q^{c,i}_{\mathcal{O}}$ for some constant $c \in \mathsf{Const}$ holds, or $\phi(t, \vec{y'})$ contains a conjunction of atom of the form $P(t_{2i-1}, t_{2i-2}) \wedge P(t_{2i-1}, t_{2i})$ where t_{2i} is an existential variable and $t_{2i} \neq t_{2i-1}$.

Base step ($i = 1$): Observe that the variable t_0 of the atom $P(t, t_0)$ must be in join with some other atom. Indeed, if not, $\mathsf{PerfectRef}(\mathcal{O}, q'_{\mathcal{O}})$ can apply the inclusion assertion $A \sqsubseteq \exists P$ to the atom $P(t, t_0)$ of $q'_{\mathcal{O}}$, thus producing a new CQ in which there is the atom $A(t)$ instead of $P(t, t_0)$, and when rewriting this new CQ with respect to $\mathcal{M}$, the result is a disjunct of $\mathsf{PerfRef}_{q'_{\mathcal{O}},\Sigma}$ that is not contained in $q_{\mathcal{S}}$, thus contradicting the fact that $q'_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

There are two possible cases: either there is an atom of the form $P(t_0, t_1)$ for some term t_1, or there is a conjunction of atoms of the form $P(t_1, t_0) \wedge P(t_1, t_2)$ for some pair of terms t_1, t_2. Indeed, if otherwise only the atom $P(t_1, t_0)$ occurs (and $P(t_1, t_2)$ for some term t_2 does not occur), the same reasoning delineated above applies, after that the unification step of $\mathsf{PerfectRef}(\mathcal{O}, q'_{\mathcal{O}})$ unifies term t_1 with term t, thus reducing the conjunction of atoms $P(t, t_0) \wedge P(t_1, t_0)$ to only $P(t, t_0)$.

In the former case, we trivially have that $q'_{\mathcal{O}} \sqsubseteq q^{0}_{\mathcal{O}}$, and therefore $q'_{\mathcal{O}} \sqsubseteq q^{1}_{\mathcal{O}}$ (observe that $q^{l}_{\mathcal{O}} \sqsubseteq q^{m}_{\mathcal{O}}$ for any pair of natural numbers $l, m \geq 0$ with $l \leq m$). In the latter case, there are three possibilities: (i) t_2 is a constant; (ii) t_2 is an existential variable and $t_2 = t_1$; (iii) t_2 is an existential variable and $t_2 \neq t_1$. In case (i), we trivially have that $q'_{\mathcal{O}} \sqsubseteq q^{t_2,1}_{\mathcal{O}}$. On the other hand, in case (ii), we have that $q'_{\mathcal{O}} \sqsubseteq q^{1}_{\mathcal{O}}$ with the homomorphism h from $q^{1}_{\mathcal{O}}$ to $q'_{\mathcal{O}}$ in which $h(x) = t$, $h(y_0) = t_0$, $h(y_1) = y_1$, and $h(y_2) = h(y_3) = t_1 = t_2$. In case (iii), there is the conjunction of atoms $P(t_{2i-1}, t_{2i-2}) \wedge P(t_{2i-1}, t_{2i})$ where t_{2i} is an existential variable and $t_{2i} \neq t_{2i-1}$. So, the base step of the induction is verified.

Inductive step: By the inductive hypothesis, we derive that either there is a natural number $k \leq i - 1$ for which one among $q'_{\mathcal{O}} \sqsubseteq q^{k}_{\mathcal{O}}$ or $q'_{\mathcal{O}} \sqsubseteq q^{c,k}_{\mathcal{O}}$ for some constant $c \in \mathsf{Const}$ holds, or $\phi(t, \vec{y'})$ contains a conjunction of atoms of the form $\bigwedge_{j=1}^{i-1} P(t_{2j-1}, t_{2j-2}) \wedge P(t_{2j-1}, t_{2j})$ where, for each $j \in [1, i-1]$, term t_{2j} is an existential variable and $t_{2j} \neq t_{2j-1}$. Consider the former case. As already observed, for any $i \geq 0$ and for any $k \leq i - 1$, the query $q^{k}_{\mathcal{O}}$ (respectively, $q^{c,k}_{\mathcal{O}}$ for any constant $c \in \mathsf{Const}$) can be obtained from $q^{i}_{\mathcal{O}}$ (respectively, $q^{c,i}_{\mathcal{O}}$) by applying some set of equalities between the variables in $q^{i}_{\mathcal{O}}$ (respectively, $q^{c,i}_{\mathcal{O}}$). This implies that $q^{k}_{\mathcal{O}} \sqsubseteq q^{i}_{\mathcal{O}}$

(respectively, $q_{\mathcal{O}}^{c,k} \sqsubseteq q_{\mathcal{O}}^{c,i}$). It follows that, if one among $q'_{\mathcal{O}} \sqsubseteq q_{\mathcal{O}}^{k}$ or $q'_{\mathcal{O}} \sqsubseteq q_{\mathcal{O}}^{c,k}$ for some constant $c \in \mathsf{Const}$ holds, then, since $q_{\mathcal{O}}^{k} \sqsubseteq q_{\mathcal{O}}^{i}$ (respectively, $q_{\mathcal{O}}^{c,k} \sqsubseteq q_{\mathcal{O}}^{c,i}$), we have that either one among $q'_{\mathcal{O}} \sqsubseteq q_{\mathcal{O}}^{i}$ or $q'_{\mathcal{O}} \sqsubseteq q_{\mathcal{O}}^{c,i}$ holds as well.

Consider the latter case. Observe that the variable t_{2i-2} of the atom $P(t_{2i-3}, t_{2i-2})$ must be in join with some other atom. Indeed, if not, then $\mathsf{PerfectRef}(\mathcal{O}, q'_{\mathcal{O}})$ can apply the inclusion assertion $A \sqsubseteq \exists P$ to the atom $P(t, t_0)$ of $q'_{\mathcal{O}}$ after unifying the variables t and t_{2j-1} for each $j \in [1, i-1]$ (i.e., after applying the set of equalities $e = \{t = t_1, t_1 = t_3, \ldots, t_{2i-5} = t_{2i-3}\}$). In this way, a new CQ is produced in which there is the atom $A(t)$ instead of $P(t, t_0)$ (notice that this would happen because, for each $j \in [0, i-1]$, term t_{2j} is an existential variable such that $t_{2j} \neq t_{2j-1}$), and when rewriting this new CQ with respect to $\mathcal{M}$, the result is a disjunct of $\mathsf{PerfRef}_{q'_{\mathcal{O}},\Sigma}$ that is not contained in $q_{\mathcal{S}}$, thus contradicting the fact that $q'_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. We can now proceed in a very similar way to the base step case ($i = 1$). We write it for the sake of completeness.

There are two possible cases: either there is an atom $P(t_{2i-2}, t_{2i-1})$ for some term t_{2i-1}, or there is a conjunction of atoms $P(t_{2i-1}, t_{2i-2}) \wedge P(t_{2i-1}, t_{2i})$ for some pair of terms t_{2i-1}, t_{2i}. In the former case, one can verify that, by construction, $q'_{\mathcal{O}} \sqsubseteq q_{\mathcal{O}}^{i-1}$, and therefore $q'_{\mathcal{O}} \sqsubseteq q_{\mathcal{O}}^{i}$. In the latter case, there are three possibilities: (i) t_{2i} is a constant; (ii) t_{2i} is an existential variable and $t_{2i} = t_{2i-1}$; (iii) t_{2i} is an existential variable and $t_{2i} \neq t_{2i-1}$. In case (i), we trivially have that $q'_{\mathcal{O}} \sqsubseteq q_{\mathcal{O}}^{t_{2i},i}$. On the other hand, in case (ii), we have that $q'_{\mathcal{O}} \sqsubseteq q_{\mathcal{O}}^{i}$ with the homomorphism h from $q_{\mathcal{O}}^{i}$ to $q'_{\mathcal{O}}$ in which $h(x) = t$, $h(y_j) = t_j$ for each $j \in [0, 2i-1]$, and $h(y_{2i}) = h(y_{2i+1}) = t_{2i-1} = t_{2i}$. In case ($iii$), there is the conjunction of atoms $P(t_{2i-1}, t_{2i-2}) \wedge P(t_{2i-1}, t_{2i})$ where t_{2i} is an existential variable and $t_{2i} \neq t_{2i-1}$. So, even the induction step is verified.

Since by assumption $q'_{\mathcal{O}}$ is a CQ, and so its body consists of a *finite* set of atoms, from the above induction we derive that there must be a natural number i for which one among $q'_{\mathcal{O}} \sqsubseteq q_{\mathcal{O}}^{i}$ or $q'_{\mathcal{O}} \sqsubseteq q_{\mathcal{O}}^{c,i}$ for some constant $c \in \mathsf{Const}$ holds, as required. $\square$

Using the above lemmata, we are now able to prove that no UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ exists. Suppose, for the sake of contradiction, the existence of a UCQ $q'_{\mathcal{O}} = {q'_{\mathcal{O}}}^{1} \cup \ldots \cup {q'_{\mathcal{O}}}^{n}$ which is a UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. For each $j \in [1, n]$, by Lemma 6.10, we know the existence of a number i_j for which either ${q'_{\mathcal{O}}}^{j} \sqsubseteq q_{\mathcal{O}}^{i_j}$, or ${q'_{\mathcal{O}}}^{j} \sqsubseteq q_{\mathcal{O}}^{c,i_j}$ for some constant $c \in \mathsf{Const}$ holds. It follows that $q'_{\mathcal{O}} \sqsubseteq q''_{\mathcal{O}} = {q''_{\mathcal{O}}}^{1} \cup \ldots \cup {q''_{\mathcal{O}}}^{n}$, where, for each $j \in [1, n]$, the CQ ${q''_{\mathcal{O}}}^{j}$ corresponds to $q_{\mathcal{O}}^{i_j}$ if ${q'_{\mathcal{O}}}^{j} \sqsubseteq q_{\mathcal{O}}^{i_j}$, and to $q_{\mathcal{O}}^{c,i_j}$ if ${q'_{\mathcal{O}}}^{j} \sqsubseteq q_{\mathcal{O}}^{c,i_j}$. Consider now the query $q'''_{\mathcal{O}} = {q'''_{\mathcal{O}}}^{1} \cup \ldots \cup {q'''_{\mathcal{O}}}^{n}$ where, for each $j \in [1, n]$, the CQ ${q'''_{\mathcal{O}}}^{j}$ corresponds to $q_{\mathcal{O}}^{i_j+1}$ if ${q'_{\mathcal{O}}}^{j} \sqsubseteq q_{\mathcal{O}}^{i_j}$, and to $q_{\mathcal{O}}^{c,i_j+1}$ if ${q'_{\mathcal{O}}}^{j} \sqsubseteq q_{\mathcal{O}}^{c,i_j}$. Observe that, by Lemma 6.9, we have $cert_{{q''_{\mathcal{O}}}^{j},\Sigma} \sqsubset cert_{{q'''_{\mathcal{O}}}^{j},\Sigma}$ for each $j \in [1, n]$. This implies that $cert_{q''_{\mathcal{O}},\Sigma} \sqsubset cert_{q'''_{\mathcal{O}},\Sigma}$, and therefore, since $q'_{\mathcal{O}} \sqsubseteq q''_{\mathcal{O}}$, we also have that $cert_{q'_{\mathcal{O}},\Sigma} \sqsubset cert_{q'''_{\mathcal{O}},\Sigma}$. Furthermore, since by Lemma 6.8 each disjunct of $q'''_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, we can finally conclude that $q'''_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ such that $cert_{q'_{\mathcal{O}},\Sigma} \sqsubset cert_{q'''_{\mathcal{O}},\Sigma}$. Thus, following Definition 3.4, this is clearly a contradiction on the fact that $q'_{\mathcal{O}}$ is a UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, as required.

Two interesting considerations follow:

(i) For a more accurate characterisation about the illustrated OBDM specification Σ and CQ $q_{\mathcal{S}}$, we refer the reader to Chapter 9 (specifically to Section 9.4), where we will see that in this case even a perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ does exist, and can be expressed in a target query language equipped with an epistemic operator.

(*ii*) We point out that the same negative result applies also to CQJFEs $q_{\mathcal{S}}$ with no existential variables at all in their body (i.e., the so-called *full conjunctive queries*). To see this, it is sufficient to consider a slight modification of the above illustrated OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and CQJFE $q_{\mathcal{S}}$. Specifically, let $\Sigma_f = \langle \mathcal{O}_f, \mathcal{S}_f, \mathcal{M}_f \rangle$ be the following OBDM specification:

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

$$\begin{array}{lrcl} m_1: & s_1(x_1) \wedge s_3(x_1, x_2) & \rightarrow & P(x_1, x_2), \\ m_2: & s_2(x) & \rightarrow & A(x). \end{array}$$

Consider now the full conjunctive query $q^f_{\mathcal{S}} = \{(x) \mid s_1(x)\}$. One can easily verify that all the steps in the above proof remain valid even if we replace the OBDM specification Σ with Σ_f and the query $q_{\mathcal{S}}$ with $q^f_{\mathcal{S}}$. Thus, the above proof also shows that no UCQ-maximally sound $\mathcal{S}_f$-to-$\mathcal{O}_f$ Σ_f-rewriting of $q^f_{\mathcal{S}}$ exists. $\square$

4 The proof is based on the idea that an assertion of the form $A \sqsubseteq \exists P$ entailed by a *DL-Lite*$_{\mathcal{R}}$ ontology $\mathcal{O}$ can be simulated using LAV mapping assertions. In our case, consider the OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ illustrated in the proof of point 3, and let $\Sigma' = \langle \mathcal{O}', \mathcal{S}, \mathcal{M}' \rangle$ be the OBDM specification in which $\mathcal{O}' = \emptyset$ ($\mathcal{O}'$ and $\mathcal{O}$ share the same alphabet) and $\mathcal{M}' = \mathcal{M} \cup \{ \ m_3 : s_2(x) \rightarrow \exists y. P(x, y) \ \}$. It can be readily seen that Σ' is *query-preserving with respect to* Σ [Calì *et al.*, 2002; Lenzerini, 2002], that is, $cert^D_{q,\Sigma} = cert^D_{q,\Sigma'}$ for every query q over $\mathcal{O}$ (equivalently, over $\mathcal{O}'$) and for every $\mathcal{S}$-database D.

Therefore, a formal proof of this case can be obtained from the above proof of point 3 by replacing the OBDM specification Σ with the OBDM specification Σ'.

Furthermore, it is clear that the same reasoning can be done with the OBDM specification Σ_f, and thus also in this case the same negative result applies also when the queries over $\mathcal{S}$ are full conjunctive queries. $\square$

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

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

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

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

Observe that the CQ $q'_{\mathcal{O}} = \{(x_1, x_2) \mid P(x_1, x_2)\}$ is not a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, because the query $q'_{\mathcal{S}} = \{(x, x) \mid s_2(x)\}$ is a disjunct of $\mathsf{PerfRef}_{q'_{\mathcal{O}},\Sigma}$ such that $q'_{\mathcal{S}} \not\sqsubseteq q_{\mathcal{S}}$.

Lemma 6.11. *If a CQ $q_\mathcal{O}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$, then either $q_\mathcal{O} \equiv \bot$ or the target list of $q_\mathcal{O}$ does not include any distinguished variable.*

Proof. We prove the claim by contradiction. Let $q_\mathcal{O} = \{(t_1, t_2) \mid \exists \vec{y}.\phi(t_1, t_2, \vec{y})\}$ be a CQ over $\mathcal{O}$ such that $\phi(t_1, t_2, \vec{y}) \not\equiv \bot$, i.e., $\phi(t_1, t_2, \vec{y})$ is a conjunction of atoms with P as predicate name, and at least one among t_1 and t_2 (which are not necessarily different) is a distinguished variable. There are two possible cases: either $P(t_1, t_2)$ is an atom occurring in $\phi(t_1, t_2, \vec{y})$, or not.

In the former case, consider the disjunct $q'_\mathcal{S} = \{(t, t) \mid \exists \vec{y}.\phi'(t, \vec{y})\}$ of $\mathsf{PerfRef}_{q_\mathcal{O},\Sigma}$ obtained by unfolding (i) each atom of the form $P(z, z)$ for a term z with $s_2(z)$, (ii) each atom $P(t_1, t_2)$ with $s_2(t)$, thus imposing the equality $t_1 = t_2$ (where $t = t_i$ if t_i is a constant for $i = 1$ or $i = 2$, otherwise t denotes a fresh term), and (iii) each remaining atom $P(z_1, z_2)$ for distinct terms z_1 and z_2 such that either $z_1 \neq t_1$ or $z_2 \neq t_2$ with $s_1(z_1, z_2)$. Observe that, since by assumption one among t_1 and t_2 is a distinguished variable, such equality is never an equality between different constant.

Consider now each possible function f from terms of $q_\mathcal{S}$ to $q'_\mathcal{S}$ for which $f(x_1) = t$ and $f(x_2) = t$. By construction, in $q'_\mathcal{S}$ there is no atom of the form $s_1(t, t)$, and therefore $s_1(f(x_1), f(x_2))$ does not occur in the body of $q'_\mathcal{S}$, where $s_1(x_1, x_2)$ is the atom occurring in $q_\mathcal{S}$. It follows that there is no homomorphism from $q_\mathcal{S}$ to $q'_\mathcal{S}$ which, due to [Chandra and Merlin, 1977], implies that $q'_\mathcal{S} \not\sqsubseteq q_\mathcal{S}$. Since $q'_\mathcal{S}$ is a disjunct of $\mathsf{PerfRef}_{q_\mathcal{O},\Sigma}$ such that $q'_\mathcal{S} \not\sqsubseteq q_\mathcal{S}$, we get that $\mathsf{PerfRef}_{q_\mathcal{O},\Sigma} \not\sqsubseteq q_\mathcal{S}$ which, due to Lemma 6.1, implies that $q_\mathcal{O}$ is not a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$, as required.

In the latter case, consider the disjunct $q'_\mathcal{S} = \{(t_1, t_2) \mid \exists \vec{y}.\phi'(t_1, t_2, \vec{y})\}$ obtained by unfolding each atom $P(z_1, z_2)$ occurring in $q_\mathcal{O}$ with $s_1(z_1, z_2)$. Since $P(t_1, t_2)$ never occurs in the body of $q_\mathcal{O}$, we have that each possible function f from terms of $q_\mathcal{S}$ to $q'_\mathcal{S}$ for which $f(x_1) = t_1$ and $f(x_2) = t_2$ is such that $s_1(f(x_1), f(x_2))$ does not occur in the body of $q'_\mathcal{S}$. With the same arguments given in the previous case, we obtain that $q_\mathcal{O}$ is not a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$, as required. □

From the above lemma, we easily derive that each possible candidate CQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$ is of the form $\{(c_1, c_2) \mid P(c_1, c_2)\}$, where c_1, c_2 are constants in Const. With this observation at hand, we are now ready to prove that no UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$ exists. Suppose, for the sake of contradiction, the existence of a UCQ $q'_\mathcal{O} = {q'_\mathcal{O}}^1 \cup \ldots \cup {q'_\mathcal{O}}^n$ which is a UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$. Consider now the query $q''_\mathcal{O} = q'_\mathcal{O} \cup {q'_\mathcal{O}}^{n+1}$, where ${q'_\mathcal{O}}^{n+1} = \{(c, c') \mid P(c, c')\}$ with c, c' being distinct fresh constants of Const not occurring anywhere else in the disjuncts of $q'_\mathcal{O}$. Obviously, we have that $cert_{{q'_\mathcal{O}}^{n+1},\Sigma} \not\sqsubseteq cert_{{q'_\mathcal{O}}^{i},\Sigma}$, for each $i \in [1, n]$. This implies that $cert_{q'_\mathcal{O},\Sigma} \sqsubset cert_{q''_\mathcal{O},\Sigma}$. Furthermore, since ${q'_\mathcal{O}}^{n+1}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$, we can finally conclude that $q''_\mathcal{O}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$ such that $cert_{q'_\mathcal{O},\Sigma} \sqsubset cert_{q''_\mathcal{O},\Sigma}$. Thus, following Definition 3.4, this is clearly a contradiction on the fact that $q'_\mathcal{O}$ is a UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$, as required.

For a more accurate characterisation about the illustrated OBDM specification Σ and CQ $q_\mathcal{S}$, the CQ$^{\neq}$ $q_\mathcal{O} = \{(x_1, x_2) \mid P(x_1, x_2) \wedge x_1 \neq x_2\}$ over $\mathcal{O}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$. More precisely, one can verify that $q_\mathcal{O}$ is the (unique up to equivalence w.r.t. Σ) UCQ$^{\neq}$-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$. □

Chapter 7

Perfect Source-to-Ontology Rewritings

With the results of previous chapters at hand, we now study both the verification, and the computation problem for perfect source-to-ontology rewritings.

7.1 Verification Problem

We remind the reader that, by definition, $q_{\mathcal{O}}$ is a perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ rewriting of $q_{\mathcal{S}}$ if and only if it is both a sound, and a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. Thus, by combining Lemma 6.1 and Lemma 5.1, we immediately get the following.

Corollary 7.1. *$q_{\mathcal{O}}$ is a perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ if and only if both $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \sqsubseteq (q_{\mathcal{S}} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},\Sigma})$ and $q_{\mathcal{S}} \sqsubseteq (\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},\Sigma})$ hold, where $n = ar(q_{\mathcal{O}}) = ar(q_{\mathcal{S}})$.*

As already discussed in Subsection 3.3.1, for OBDM specifications where inconsistencies can not arise, the notion of perfect source-to-ontology rewriting coincides with the notion of realization considered in [Lutz *et al.*, 2018]. Since the problem of checking whether $q_{\mathcal{O}}$ is a realization of $q_{\mathcal{S}}$ in Σ is in general Π^p_2-hard even when $\mathcal{O}$ contains no assertions (i.e., $\mathcal{O} = \emptyset$), $\mathcal{M}$ is a pure GAV mapping, and both $q_{\mathcal{S}}$ and $q_{\mathcal{O}}$ are boolean CQs [Lutz *et al.*, 2018, Theorem 11], and since in those cases the two notions are equivalent, such lower bound applies also to our notion.

The following theorem characterises the computational complexity of the verification problem for perfect source-to-ontology rewritings.

Theorem 7.1. *The verification problem for perfect source-to-ontology rewritings is Π^p_2-complete.*

Proof. As for the upper bound, by virtue of Corollary 7.1, it is sufficient to show how to check the following two containments in Π^p_2, where $n = ar(q_{\mathcal{O}}) = ar(q_{\mathcal{S}})$: (i) $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \sqsubseteq (q_{\mathcal{S}} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},\Sigma})$, which holds if and only if $q_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$; and (ii) $q_{\mathcal{S}} \sqsubseteq (\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},\Sigma})$, which holds if and only if $q_{\mathcal{O}}$ is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. By Theorem 6.1, the former can be verified in Π^p_2, and, by Theorem 5.1, the latter can be verified even in NP.

As for the lower bound, it follows from [Lutz *et al.*, 2018, Theorem 11]. $\square$

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

7.2 Computation Problem

As for computation, consider any OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and UCQ $q_{\mathcal{S}}$ over $\mathcal{S}$. We have that (*i*) the unique (up to equivalence w.r.t. Σ) UCQ-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ always exists and can be computed by means of the $\mathsf{MinimallyComplete}$ algorithm (cf. Theorem 5.2); and (*ii*) by construction (see Definition 3.5), either this latter is also a sound, and therefore a perfect, $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, or no UCQ-perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ exists.

With these observations at hand, we can easily derive the algorithm $\mathsf{Perfect}$ together with its termination and correctness.

Algorithm 7.1 $\mathsf{Perfect}$

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

Output:
either a UCQ $q_{\mathcal{O}}$ over $\mathcal{O}$, or report that "no UCQ-perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ exists"

$q_{\mathcal{O}} := \mathsf{MinimallyComplete}(\Sigma, q_{\mathcal{S}})$
if $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \sqsubseteq (q_{\mathcal{S}} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},\Sigma})$ **then**
  return $q_{\mathcal{O}}$
else
  return "no UCQ-perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ exists"
end if

Essentially, the algorithm computes the unique (up to equivalence w.r.t. Σ) UCQ-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ using the $\mathsf{MinimallyComplete}$ algorithm (cf. Section 5.2), and then checks whether this latter is also a sound, and therefore a perfect, $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. Notice that this last step is always deterministically feasible in exponential time (cf. Theorem 6.1). Finally, observe that the overall running time of the algorithm is exponential in the size of the input.

Theorem 7.2. *$\mathsf{Perfect}(\Sigma, q_{\mathcal{S}})$ terminates and returns the perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ if it exists and can be expressed as a UCQ, otherwise it reports that no UCQ-perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ exists.*

Furthermore, as a straightforward consequence of Corollary 5.1, we also get the following interesting result.

Corollary 7.2. *The UCQ-perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of a CQ $q_{\mathcal{S}}$ either does not exists, or it can be expressed as a CQ as well.*

Chapter 8

Sound Source-to-Ontology Rewritings in Restricted Scenarios

We now deal with the restricted scenarios mentioned in Section 6.2. Before delving into the technical part, we observe that, despite their limitations, the expressive power of these scenarios is enough for various meaningful applications. Indeed, several popular ontologies are expressible in the DL $\mathit{DL\text{-}Lite}_{\mathsf{RDFS}}$, e.g., `SKOS`[1] [Miles and Pérez-Agüera, 2007; Miles and Bechhofer, 2009] and `Dublin Core`[2] [Weibel *et al.*, 1998], and the form of pure GAV mapping is exactly the one originally defined in the literature of data integration [Lenzerini, 2002].

Furthermore, the class of (U)CQJFEs captures data services expressible in the famous (U)SPJ (Union, Select, Project, Join) fragment of Relational Algebra [Codd, 1970], with the only limitation that joining variables must appear in the final projection of the USPJ Relational Algebra query, i.e., they appear in the target list of the equivalent UCQ. Notice that (*i*) CQJFEs extends the class of *Full Conjunctive Queries* with the possibility of having non-join existential variables occurring in their body. This latter is a well-known class of queries studied for various optimisation in the relational database theory (see, e.g., [Beame *et al.*, 2014; Koutris *et al.*, 2016; Ketsman and Suciu, 2017]); and (*ii*) such fragment is precisely the one needed for all tasks related to source profiling [Abedjan *et al.*, 2017; Abedjan *et al.*, 2018].

In both the restricted scenarios, observe that the DL ontology language adopted in the restricted setting for OBDM specifications is $\mathit{DL\text{-}Lite}_{\mathsf{RDFS}}$, which does not allow for disjointness axioms. It follows that, for any OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ in this restricted setting, since $\mathcal{O}$ is a $\mathit{DL\text{-}Lite}_{\mathsf{RDFS}}$ ontology, we have that $\mathcal{V}_{\mathcal{O}} \equiv \bot$. This has two further implications that are worth mentioning:

- Each $\mathcal{S}$-database D is consistent with Σ, and therefore all the results we will present hold even according to the semantics proposed in [Lutz *et al.*, 2018];
- For each UCQ $q_{\mathcal{O}}$ over $\mathcal{O}$, the UCQ $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ over $\mathcal{S}$ is the perfect $\mathcal{O}$-to-$\mathcal{S}$ Σ-rewriting of $q_{\mathcal{O}}$, that is, $cert^{D}_{q_{\mathcal{O}},\Sigma} = \mathsf{PerfRef}^{D}_{q_{\mathcal{O}},\Sigma}$ for each $\mathcal{S}$-database D.

[1] `Simple Knowledge Organization System: https://www.w3.org/2004/02/skos/`
[2] `http://dublincore.org/`

With the above considerations at hand, we easily get a refinement of Corollary 7.1 and Lemmata 6.1 and 5.1 in this restricted setting for OBDM specifications.

Corollary 8.1. *In the setting for OBDM specifications of the restricted scenarios, $q_{\mathcal{O}}$ is a perfect (respectively, sound, complete) $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ if and only if $q_{\mathcal{S}} \equiv \mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ (respectively, $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \sqsubseteq q_{\mathcal{S}}$, $q_{\mathcal{S}} \sqsubseteq \mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$).*

Let us now introduce some further properties of this restricted setting. Given an OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ of the restricted setting and an atom β over $\mathcal{O}$, we denote by $\rho(\beta, \Sigma)$ the disjunction of conjunctions obtained by first unfolding β with respect to $\mathcal{O}$, and then by unfolding the resulting formula with respect to $\mathcal{M}$.

The unfolding of an atom β with respect to a *DL-Lite*$_{\mathsf{RDFS}}$ ontology $\mathcal{O}$ is the disjunction of atoms $\lambda(\beta, \mathcal{O})$ defined as follows (see also [Cima *et al.*, 2020b]):

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

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

where y denotes a fresh existential variable, A and A' denote atomic concepts, and P and E denote atomic roles. Finally, $\mathcal{O} \models B \sqsubseteq A$ for a basic role B and atomic concept A (respectively, $\mathcal{O} \models R \sqsubseteq P$ for a basic role R and atomic role P) hold if $B^{\mathcal{I}} \subseteq A^{\mathcal{I}}$ (respectively, $R^{\mathcal{I}} \subseteq P^{\mathcal{I}}$) in each interpretation $\mathcal{I}$ for $\mathcal{O}$ such that $\mathcal{I} \models \mathcal{O}$.

The unfolding of a formula $\lambda(\beta, \mathcal{O})$ for an atom β over an ontology $\mathcal{O}$ with respect to a pure GAV mapping $\mathcal{M}$ relating a schema $\mathcal{S}$ to $\mathcal{O}$ is obtained by replacing each atom β' occurring in $\lambda(\beta, \mathcal{O})$ with the logical disjunction of all the conjunctions of atoms over $\mathcal{S}$ corresponding to the left-hand sides of mapping assertions in $\mathcal{M}$ having the predicate name β' in the right-hand side (being careful to use unique variables in place of those variables that appear in the left-hand side of the mapping assertions but not in the right-hand side of those).

Example 8.1. Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the following OBDM specification:

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

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

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

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

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

Example 8.2. Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the the OBDM specification illustrated in Example 8.1. Consider the CQ $q_{\mathcal{O}} = \{(x) \mid \exists y.P_1(y, x) \wedge A(x)\}$ over $\mathcal{O}$, and let $\beta_1 = P_1(y, x)$ and $\beta_2 = A(x)$. Then, $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ can be obtained by turning the logical query $\{(x) \mid \exists y.\rho(\beta_1, \Sigma) \wedge \rho(\beta_2, \Sigma)\} = \{(x) \mid \exists y.((s_1(y, x)) \vee (\exists y_1.s_1(y, y_1) \wedge s_2(y_1, x))) \wedge ((\exists y_3.s_2(c, x) \wedge s_3(x, y_3)) \vee (\exists y_2.s_3(x, y_2)))\}$ into an equivalent UCQ over schema $\mathcal{S}$, thus obtaining $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} = q^1_{\mathcal{S}} \cup q^2_{\mathcal{S}} \cup q^3_{\mathcal{S}} \cup q^4_{\mathcal{S}}$, where:

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

Before studying sound source-to-ontology rewritings in the restricted scenarios, we introduce some preliminary crucial notions and results for the class of queries $\mathcal{L}_{\mathcal{S}}$ = UCQJFEs used for expressing data services. Furthermore, from now on, for ease of exposition, we assume that all the queries over source schemas do not involve atoms $\top$ and $\bot$. We point out, however, that everything can be generalised in a straightforward manner to include also $\top$ and $\bot$ as possible atoms.

Definition 8.1. Let $q = \{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$ be a CQ over a schema $\mathcal{S}$, and let $\alpha_1 = s(t_{1,1}, \ldots, t_{1,n})$ and $\alpha_2 = s(t_{2,1}, \ldots, t_{2,n})$ be two atoms over $\mathcal{S}$, where $\alpha_2 \in \phi(\vec{x}, \vec{y})$. We say that α_1 *instantiates* α_2, if the following holds for each $i \in [1, n]$: if $t_{2,i}$ is a distinguished variable or a constant, then $t_{2,i} = t_{1,i}$.

Example 8.3. Consider the following CQs $q_1 = \{(x) \mid s_1(c, x, x) \wedge s_2(c, x)\}$ and $q_2 = \{(x) \mid \exists y.s_1(c, x, y) \wedge s_2(x, x)\}$ over a schema $\mathcal{S}$. Let $\alpha_1 = s_1(c, x, x)$, $\alpha_2 = s_2(c, x)$, $\alpha_3 = s_1(c, x, y)$, and $\alpha_4 = s_2(x, x)$. We have that α_1 instantiates α_3, whereas α_2 does not instantiate α_4. $\square$

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

Lemma 8.1. *Let q_1 and q_2 be a CQ and a CQJFE, respectively, over a schema $\mathcal{S}$ with the same target list. We have that $q_1 \sqsubseteq q_2$ if and only if for each atom α_2 of q_2 there exists an atom α_1 of q_1 such that α_1 instantiates α_2.*

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

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

Let the containment problem for UCQJFEs be the following decision problem: given a UCQ q' and a UCQJFE q over the same schema $\mathcal{S}$, check whether $q' \sqsubseteq q$.

Lemma 8.2. *The containment problem for UCQJFEs is in* PTIME.

Proof. To begin, observe that for each pair of UCQs q_1, q_2 we have $q_1 \sqsubseteq q_2$ if and only if for each disjunct q' of q_1 there exists a disjunct q of q_2 such that $q' \sqsubseteq q$ [Sagiv and Yannakakis, 1980]. It is therefore sufficient to show that, given a CQ $q' = \{\vec{t'} \mid \exists \vec{y'}.\phi'(\vec{x'}, \vec{y'})\}$ and a CQJFE $q = \{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$ not necessarily with the same target lists, checking whether $q' \sqsubseteq q$ can be done in polynomial time.

If q' and q do not have the same target list, i.e., $\vec{t'} = (t'_1, \ldots, t'_n) \neq \vec{t} = (t_1, \ldots, t_n)$, then consider the function f from the set of terms in the target list of q to the set of terms in the target list of q' with $f(t_i) = t'_i$, for each $i \in [1, n]$. Formally, since repetitions of terms in target lists is allowed, f might give rise to a *multivalued function.*[3] In this case, as well as in the case that $f(a) = b$ with $a \neq b$ for two constants $a \in \vec{t}$ and $b \in \vec{t'}$, it is straightforward to verify that $q' \not\sqsubseteq q$ trivially holds. Indeed, in those cases there can be no homomorphism from q to q' by construction.

Consider the query q'' obtained in polynomial time from q by replacing every occurrence of term t_i in q (even in the target list) with term $f(t_i) = t'_i$, for each $i \in [1, n]$. Observe that now q'' is a CQJFE with target list $\vec{t'}$, i.e., the same target list of q'. By virtue of Lemma 8.1, we can now check in polynomial time whether $q' \sqsubseteq q''$, where, if the answer is yes, then clearly $q' \sqsubseteq q$ as well; otherwise, it can be readily seen that $D_{q'}$ (i.e., the freezing of q') is a database witnessing that $q' \not\sqsubseteq q$.

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

In what follows in this chapter, unless otherwise stated, we assume that OBDM specifications are expressed in the restricted setting for OBDM specifications mentioned in Section 6.2, i.e., the DL ontology language is *DL-Lite*$_{\mathsf{RDFS}}$ and the mapping language follows the pure GAV approach. Furthermore, unless otherwise stated, we assume that the query language $\mathcal{L}_{\mathcal{S}}$ for expressing data services is the one of UCQJFEs and CQJFEs in Section 8.1 and Section 8.2, respectively.

[3] In mathematics, a *multivalued function* (also known as *multiple-valued function* [Knopp, 1996]) $f : A \to B$ is similar to a function, but it may associate more than one possible element $y \in B$ to each element $x \in A$.

8.1 Restricted Scenario for UCQJFEs

In this section, we study both the verification, and the computation problem for sound source-to-ontology rewritings in the restricted scenario for UCQJFEs. We recall that, while the language $\mathcal{L}_{\mathcal{S}}$ of queries over source schemas $\mathcal{S}$ is the one of UCQJFEs, the language $\mathcal{L}_{\mathcal{O}}$ for queries over ontologies $\mathcal{O}$ remains the one of UCQs.

8.1.1 Verification Problem

The following theorem characterises the computational complexity of the verification problem for sound source-to-ontology rewritings in the restricted scenario for UCQJFEs.

Theorem 8.1. *In the restricted scenario for UCQJFEs, the verification problem is coNP-complete.*

Proof. As for the upper bound, by virtue of Corollary 8.1, it is sufficient to show how to check the containment $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \sqsubseteq q_{\mathcal{S}}$ in coNP. In particular, checking whether $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \not\sqsubseteq q_{\mathcal{S}}$ can be done in NP in the following way: (i) we guess a query q' over $\mathcal{O}$ with the same arity of $q_{\mathcal{O}}$ and size at most $\sigma(q_{\mathcal{O}})$, a sequence ρ of ontology assertions, and a query q'' over $\mathcal{S}$ with the same arity of $q_{\mathcal{O}}$ and size at most $\sigma(\mathcal{M}) \times \sigma(q')$, and (ii) likewise to Theorem 5.1, we check in polynomial time first whether we can rewrite $q_{\mathcal{O}}$ into q' through ρ, and then whether q'' is in $\mathsf{MapRef}(q', \mathcal{M})$. Finally, we check whether $q'' \not\sqsubseteq q_{\mathcal{S}}$, which, since $q_{\mathcal{S}}$ is a UCQJFE, by virtue of Lemma 8.2, this last step can be done in polynomial time as well.

As for the lower bound, the proof of coNP-hardness is by a LogSpace reduction from the VALIDITY problem, which is coNP-complete (see, e.g., [Papadimitriou, 1994]). VALIDITY is the problem of deciding, given a 3-DNF formula $F = c_1 \vee \ldots \vee c_m$ on a set of variables $X = \{x_1, \ldots, x_n\}$, whether F is *valid*, i.e., whether F is satisfied by every possible truth assignment to the variables in X. Each clause c_i is a conjunction of three literals, where each literal is either a variable $x_i \in X$ or its negated.

We define an OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ with $\mathcal{O}$ containing no axioms, and $\mathcal{S}$ and $\mathcal{M}$ as follows: for each variable $x_i \in X$, schema $\mathcal{S}$ comprises two unary relations s_{i_T} and s_{i_F}, and a further unary relation s'_i. Finally, for each variable $x_i \in X$, the mapping $\mathcal{M}$ includes the following three mapping assertions:

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

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

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

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

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

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

To illustrate the reduction, we use the following formula: $F = (x_1 \wedge x_2 \wedge \neg x_3) \bigvee (\neg x_1 \wedge x_2 \wedge \neg x_4)$. In this case, the reduction would produce the mapping $\mathcal{M}$ composed of the following mapping assertions:

$$
\begin{array}{rcl}
s_{1_T}(x) & \rightarrow & A_1(x), \\
s_{1_F}(x) & \rightarrow & A_1(x), \\
s_{2_T}(x) & \rightarrow & A_2(x), \\
s_{2_F}(x) & \rightarrow & A_2(x), \\
s_{3_T}(x) & \rightarrow & A_3(x), \\
s_{3_F}(x) & \rightarrow & A_3(x), \\
s_{4_T}(x) & \rightarrow & A_4(x), \\
s_{4_F}(x) & \rightarrow & A_4(x), \\
s'_1(x) & \rightarrow & B_1(x), \\
s'_2(x) & \rightarrow & B_2(x), \\
s'_3(x) & \rightarrow & B_3(x), \\
s'_4(x) & \rightarrow & B_4(x),
\end{array}
$$

the following UCQJFE $q_{\mathcal{S}} = \{(x_1, x_2, x_3, x_4) \mid s'_1(x_1) \wedge s'_2(x_2) \wedge s'_3(x_3) \wedge s'_4(x_4) \wedge s_{1_T}(x_1) \wedge s_{2_T}(x_2) \wedge s_{3_F}(x_3)\} \cup \{(x_1, x_2, x_3, x_4) \mid s'_1(x_1) \wedge s'_2(x_2) \wedge s'_3(x_3) \wedge s'_4(x_4) \wedge s_{1_F}(x_1) \wedge s_{2_T}(x_2) \wedge s_{4_F}(x_4)\}$ over $\mathcal{S}$, and the following CQJFE $q_{\mathcal{O}} = \{(x_1, x_2, x_3, x_4) \mid B_1(x_1) \wedge B_2(x_2) \wedge B_3(x_3) \wedge B_4(x_4) \wedge A_1(x_1) \wedge A_2(x_2) \wedge A_3(x_3) \wedge A_4(x_4)\}$ over $\mathcal{O}$.

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

"**Only-if part:**" Suppose that formula F is valid, that is, F is satisfied by every possible truth assignment to the variables in X. It follows that, for any possible choice for unfolding the atoms $A_i(x_i)$ for $i = 1, \ldots, n$ (which can be equivalently seen as an assignment $V = \{v_1, \ldots, v_n\}$ to the variables in $X = \{x_1, \ldots, x_n\}$), the query over $\mathcal{S}$ obtained is such that all the atoms also appear in a disjunct q_j of $q_{\mathcal{S}}$ for some $j \in [1, m]$ (equivalently, at least one clause c_j for some $j \in [1, m]$ is satisfied under the truth assignment V). It follows that $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \sqsubseteq q_{\mathcal{S}}$ which, due to Corollary 8.1, implies that $q_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

"**If part:**" Suppose that formula F is not valid, that is, there exists a truth assignment $V = \{v_1, \ldots, v_n\}$ to the variables in $X = \{x_1, \ldots, x_n\}$ that does not satisfy F. Consider now the disjunct q of $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ obtained by unfolding atom $A_i(x_i)$ of $q_{\mathcal{O}}$ with atom $s_{i_T}(x_i)$ if $v_i = 1$, and with atom $s_{i_F}(x_i)$ otherwise (i.e., $v_i = 0$), for each $i \in [1, n]$. As a result, for each disjunct q' of $q_{\mathcal{S}}$, there is at least one atom of q' not occurring in q. In proof, if there exists some disjunct q_j of $q_{\mathcal{S}}$ such that every atom of q_j appears also in q, then the clause c_j corresponding to disjunct q_j is satisfied under the truth assignment V, which would contradict the fact that F is not satisfied under such truth assignment.

This implies that, for each disjunct q_j of $q_{\mathcal{S}}$, there is no homomorphism from q_j

to q. It follows that $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \not\sqsubseteq q_{\mathcal{S}}$ which, due to Corollary 8.1, implies that $q_{\mathcal{O}}$ is not a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. □

Note that (i) the coNP upper bound holds even when ontologies $\mathcal{O}$ are expressed in a fragment of *DL-Lite*$_{\mathcal{R}}$ that does not admit disjointness assertions (thus, a more expressive language of *DL-Lite*$_{\mathsf{RDFS}}$) and mappings $\mathcal{M}$ are GLAV mappings (rather than pure GAV mappings), and (ii) the coNP lower bound already holds when $q_{\mathcal{O}}$ is a CQJFE, both $q_{\mathcal{S}}$ and $q_{\mathcal{O}}$ do not have existential variables, and $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ is such that $\mathcal{O}$ contains no axioms, and $\mathcal{M}$ is both a pure GAV mapping and a LAV mapping. In the following section, we will see that the computational complexity of the verification problem further decreases when $q_{\mathcal{S}}$ is restricted to be a CQJFE.

8.1.2 Computation Problem

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

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

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

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

Lemma 8.3. *Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification, and let $q_{\mathcal{S}}$ be a UCQJFE over $\mathcal{S}$. If a CQ $q_{\mathcal{O}}$ over $\mathcal{O}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, then there exists a CQ $q'_{\mathcal{O}}$ over $\mathcal{O}$ with same target list of $q_{\mathcal{O}}$ such that (*i*) the body of $q'_{\mathcal{O}}$ is the conjunction of at most $bound(\mathcal{M}, q_{\mathcal{S}})$ atoms occurring in the body of $q_{\mathcal{O}}$ (and therefore, $cert_{q_{\mathcal{O}},\Sigma} \sqsubseteq cert_{q'_{\mathcal{O}},\Sigma}$), and (*ii*) $q'_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ as well.*

Proof. Since $q_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, due to Corollary 8.1, we have that $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \sqsubseteq q_{\mathcal{S}}$, that is, each disjunct of $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ is contained in some disjunct of $q_{\mathcal{S}}$. In particular, without loss of generality, we can assume that each disjunct q of $q_{\mathcal{S}}$ is such that there is some disjunct r of $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ for which $r \sqsubseteq q$. In fact, the other disjuncts of $q_{\mathcal{S}}$ that do not satisfy the above condition can simply be discarded, and the resulting $q_{\mathcal{S}}$ will remain such that $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \sqsubseteq q_{\mathcal{S}}$.

Let n denote the arity of $q_{\mathcal{O}}$ and $q_{\mathcal{S}}$, and let the target list of $q_{\mathcal{O}}$ be $\vec{t} = (t_1, \ldots, t_n)$. Without loss of generality, we can assume that the target list of each disjunct q of $q_{\mathcal{S}}$ is the same as $q_{\mathcal{O}}$, i.e., $\vec{t} = (t_1, \ldots, t_n)$. Indeed, if this is not the case, then each disjunct q' of $q_{\mathcal{S}}$ with target list $t' = (t'_1, \ldots, t'_n) \neq \vec{t}$ can be replaced with the (equivalent or more specific) disjunct q obtained from q' by replacing everywhere (even in the target list) term t'_i with term t_i, for $i \in [1, n]$. Notice that, since the target list of each disjunct in $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ is the same as $q_{\mathcal{O}}$ (this is because $\mathcal{M}$ is composed of pure GAV mapping assertions), i.e., $\vec{t}$, and since by assumption there exists some disjunct r of $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ for which $r \sqsubseteq q'$, when applying this replacement it is never the case that for some pair of numbers $j, k \in [1, n]$ with $j \neq k$ we have $t'_j = t'_k$ but $t_j \neq t_k$, or that t'_j is a constant whereas t_j is a distinguished variable for some number $j \in [1, n]$ (otherwise, this would be easily a contradiction to the fact that $r \sqsubseteq q'$).

We now show by induction on $\eta(q_{\mathcal{S}})$ (i.e., the sum of the number of atoms occurring in the body of the various disjuncts of $q_{\mathcal{S}}$) the existence of $m \leq \mathit{bound}(\mathcal{M}, q_{\mathcal{S}})$ atoms $\beta_1, \ldots, \beta_m$ occurring in the body of $q_{\mathcal{O}}$ for which the CQ $q'_{\mathcal{O}} = \{\vec{t} \mid \exists \vec{y}.\beta_1 \wedge \ldots \wedge \beta_m\}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, thus proving the claim. We do this by exploiting Lemma 8.1. Specifically, consider each CQ r that is a disjunct of $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$. We know that there is a CQJFE q that is a disjunct of $q_{\mathcal{S}}$ for which $r \sqsubseteq q$, and, moreover, since r and q have the same target list, by Lemma 8.1 we know that for each atom α_2 of q there exists an atom α_1 of r such that α_1 instantiates α_2.

Base step ($\eta(q_{\mathcal{S}}) = 1$): In this case, $q_{\mathcal{S}}$ is a single CQJFE whose body consists of only one atom α. So, there must exists at least an atom β in the body of $q_{\mathcal{O}}$ for which every possible disjunct of $\rho(\beta, \Sigma)$ contains at least an atom that instantiates α. Indeed, if this is not the case, then the disjunct r of $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ obtained by unfolding each atom β' of $q_{\mathcal{O}}$ with a disjunct of $\rho(\beta', \Sigma)$ which contains no atom that instantiates α would be such that $r \not\sqsubseteq q_{\mathcal{S}}$, because, as a result, there would be no atom in r that instantiates α (cf. Lemma 8.1). Clearly, the fact that $r \not\sqsubseteq q_{\mathcal{S}}$ for a disjunct r of $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ would also imply that $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \not\sqsubseteq q_{\mathcal{S}}$ which, by Corollary 8.1, in turn would imply that $q_{\mathcal{O}}$ is not a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, which would be a contradiction on the initial assumption.

Thus, such atom β in the body of $q_{\mathcal{O}}$ must exists. But then, by exploiting Lemma 8.1, it is trivial to see that the CQ $q'_{\mathcal{O}} = \{\vec{t} \mid \exists \vec{y}.\beta\}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, as required.

Inductive step: We start with the following observation. Since each disjunct r of $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ is such that there exists a disjunct q of $q_{\mathcal{S}}$ for which $r \sqsubseteq q$ (or equivalently, by Lemma 8.1, there exists a disjunct q of $q_{\mathcal{S}}$ for which for each atom α of q there is an atom of r that instantiates α), there must exists at least one atom β of $q_{\mathcal{O}}$ such that in every disjunct of $\rho(\beta, \Sigma)$ there is at least an atom that instantiates some atom occurring in the various disjuncts of $q_{\mathcal{S}}$. Indeed, if this is not the case, then the disjunct r of $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ obtained by unfolding each atom β' of $q_{\mathcal{O}}$ with a disjunct of $\rho(\beta', \Sigma)$ which contains no atom that instantiates some atom of $q_{\mathcal{S}}$ would be trivially such that $r \not\sqsubseteq q_{\mathcal{S}}$ due to Lemma 8.1. As explained previously, the fact that $r \not\sqsubseteq q_{\mathcal{S}}$ for a disjunct r of $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ would also imply that $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \not\sqsubseteq q_{\mathcal{S}}$ which, by Corollary 8.1, in turn would imply that $q_{\mathcal{O}}$ is not a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, which would be a contradiction on the initial assumption.

So, there must be (at least) one atom β in the body of $q_{\mathcal{O}}$ such that in every disjunct of $\rho(\beta, \Sigma)$ there is at least one atom that instantiates some atom occurring in the various disjuncts of $q_{\mathcal{S}}$. In particular, consider $\rho(\beta, \Sigma)$ for such atom β. For each disjunct θ_i of $\rho(\beta, \Sigma)$, let $q_{\mathcal{S}}^{\theta_i}$ be the UCQJFE obtained from $q_{\mathcal{S}}$ by removing all the atoms α such that an atom of θ_i instantiates α. Notice that, since each disjunct θ_i of $\rho(\beta, \Sigma)$ instantiates some atom of $q_{\mathcal{S}}$, each UCQJFE $q_{\mathcal{S}}^{\theta_i}$ is such that (i) $\eta(q_{\mathcal{S}}^{\theta_i}) \leq \eta(q_{\mathcal{S}}) - 1$ (i.e., there is at least an atom of $q_{\mathcal{S}}$ not occurring anymore in $q_{\mathcal{S}}^{\theta_i}$), and (ii) $q_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}^{\theta_i}$, due to the facts that $q_{\mathcal{S}} \sqsubseteq q_{\mathcal{S}}^{\theta_i}$ clearly holds and the initial assumption that $q_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

By the inductive hypothesis, for each disjunct θ_i of $\rho(\beta, \Sigma)$, there are atoms $\beta_1^{\theta_i}, \beta_2^{\theta_i}, \ldots, \beta_{p_i}^{\theta_i}$ for which $q_{\mathcal{O}}^{\theta_i} = \{\vec{t} \mid \exists \vec{y}_{\theta_i}. \beta_1^{\theta_i} \wedge \beta_2^{\theta_i} \wedge \ldots \wedge \beta_{p_i}^{\theta_i}\}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}^{\theta_i}$, where $p_i \leq \mathit{bound}(\mathcal{M}, q_{\mathcal{S}}^{\theta_i})$, i.e., $p_i \leq 1 + \lambda(\mathcal{M})^1 + \lambda(\mathcal{M})^2 + \ldots + \lambda(\mathcal{M})^{\eta(q_{\mathcal{S}})-1}$. But then, consider the following CQ:

$$q'_{\mathcal{O}} = \{\vec{t} \mid \exists \vec{y}. \beta \bigwedge \beta_1^{\theta_1} \wedge \beta_2^{\theta_1} \wedge \ldots \wedge \beta_{p_1}^{\theta_1} \bigwedge \beta_1^{\theta_2} \wedge \beta_2^{\theta_2} \wedge \ldots \wedge \beta_{p_2}^{\theta_2} \bigwedge \cdots \bigwedge \beta_1^{\theta_k} \wedge \beta_2^{\theta_k} \wedge \ldots \wedge \beta_{p_k}^{\theta_k}\}$$

It is not hard to ascertain that $q'_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, where k is the number of disjuncts in $\rho(\beta, \Sigma)$, and $p_i \leq 1 + \lambda(\mathcal{M})^1 + \lambda(\mathcal{M})^2 + \ldots + \lambda(\mathcal{M})^{\eta(q_{\mathcal{S}})-1}$ for each $i \in [1, k]$. In proof, consider each disjunct θ_i of $\rho(\beta, \Sigma)$ for $i \in [1, k]$. Since the CQ $q_{\mathcal{O}}^{\theta_i}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}^{\theta_i}$, by Lemma 8.1, we derive that for each possible disjunct r^{θ_i} obtained by turning in disjunctive normal form the formula $\rho(\beta_1^{\theta_i}, \Sigma) \wedge \rho(\beta_2^{\theta_i}, \Sigma) \wedge \ldots \wedge \rho(\beta_{p_i}^{\theta_i}, \Sigma)$ there is a disjunct q^{θ_i} of $q_{\mathcal{S}}^{\theta_i}$ for which for each atom α of q^{θ_i} there is an atom of r^{θ_i} that instantiates α. This, together with the fact that all the atoms α occurring in $q_{\mathcal{S}}$ and not occurring in $q_{\mathcal{S}}^{\theta_i}$ are such that there is an atom of θ_i that instantiates α, allows us to derive that each disjunct $r_{\wedge}^{\theta_i}$ in the formula $\theta_i \wedge \rho(\beta_1^{\theta_i}, \Sigma) \wedge \rho(\beta_2^{\theta_i}, \Sigma) \wedge \ldots \wedge \rho(\beta_{p_i}^{\theta_i}, \Sigma)$ turned in disjunctive normal form is such that there is a disjunct q of $q_{\mathcal{S}}$ for which for each atom α of q there is an atom of $r_{\wedge}^{\theta_i}$ that instantiates α. Since this is true for each disjunct θ_i of $\rho(\beta, \Sigma)$, and since for each $i \in [1, k]$ the conjunction of atoms $\beta_1^{\theta_i} \wedge \beta_2^{\theta_i} \wedge \ldots \wedge \beta_{p_i}^{\theta_i}$ occurs in the body of the CQ $q'_{\mathcal{O}}$, we easily derive that for each possible disjunct r' of $q'_{\mathcal{O}}$ there is a disjunct q of $q_{\mathcal{S}}$ for which for each atom α of q there is an atom of r' that instantiates α. Thus, by Lemma 8.1, it follows that $q'_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, as required.

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

By relying on the above lemma, we immediately derive the following enumerative algorithm MaximallySoundUCQJFEs for computing UCQ-maximally sound source-to-ontology rewritings in the restricted scenario for UCQJFEs.

Informally, the algorithm simply enumerates all the possible CQs over $\mathcal{O}$ with at most $\mathit{bound}(\mathcal{M}, q_{\mathcal{S}})$ atoms occurring in their bodies and possibly involving constants occurring in $q_{\mathcal{S}}$ and $\mathcal{M}$ as terms. Then, it returns the union of all and only the ones that are sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewritings of $q_{\mathcal{S}}$. Notice that checking whether a CQ $q_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ can be always done deterministically in exponential time (cf. Theorem 8.1).

Example 8.4. Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the following OBDM specification:

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

Algorithm 8.1 MaximallySoundUCQJFEs

Input:

OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ where $\mathcal{O}$ is a *DL-Lite*$_{\mathsf{RDFS}}$ ontology and $\mathcal{M}$ is a pure GAV mapping;

UCQJFE $q_{\mathcal{S}}$ over $\mathcal{S}$

Output:

UCQ $q_{\mathcal{O}}$

```
q_O := ⊥
for each CQ q over O having at most bound(M, q_S) atoms in its body and
    possibly involving constants occurring in q_S and M as terms do
    if q is a sound S-to-O Σ-rewriting of q_S then
        q_O := q_O ∪ q
    end if
end for
return q_O
```

- $\mathcal{M} = \{ m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8 \}$, where:

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

For the UCQJFE $q_{\mathcal{S}} = \{(x_1, x_2) \mid \exists y. s_1(x_1, x_2) \wedge s_2(x_2, y)\} \cup \{(x_1, x_2) \mid \exists y. s_3(x_1, x_2) \wedge s_4(x_2, y)\}$, the unique (up to equivalence w.r.t. Σ) UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ is the query $q_{\mathcal{O}} = \{(x_1, x_2) \mid P_1(x_1, x_2) \wedge A_1(x_2) \wedge P_2(x_1, x_2)\} \cup \{(x, x) \mid A_2(x)\} \cup \{(x, c_1) \mid \exists y. P_3(x, y)\}$. Indeed one can verify that, on the one hand, each disjunct of $q_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, and, on the other hand, each possible CQ q' over $\mathcal{O}$ being a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ is such that $cert_{q',\Sigma} \sqsubseteq cert_{q,\Sigma}$ for some disjunct q of $q_{\mathcal{O}}$.

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

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

Theorem 8.2. *In the restricted scenario for UCQJFEs, MaximallySoundUCQJFEs $(\Sigma, q_{\mathcal{S}})$ terminates and returns the unique (up to equivalence w.r.t. Σ) UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.*

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

As for the correctness, we first point out that the computed UCQ $q_{\mathcal{O}}$ is clearly a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. Indeed, by construction, $q_{\mathcal{O}}$ is a UCQ whose disjuncts are sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewritings of $q_{\mathcal{S}}$. We now show that $q_{\mathcal{O}}$ is actually the unique (up to equivalence w.r.t. Σ) UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, that is, each UCQ $q'_{\mathcal{O}}$ that is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ is such that $cert_{q'_{\mathcal{O}},\Sigma} \sqsubseteq cert_{q_{\mathcal{O}},\Sigma}$ (cf. Definition 3.6). We do this by way of contradiction.

Let $q'_{\mathcal{O}}$ be a UCQ such that $cert_{q'_{\mathcal{O}},\Sigma} \not\sqsubseteq cert_{q_{\mathcal{O}},\Sigma}$, that is, there exists an $\mathcal{S}$-database D consistent with Σ such that $cert^D_{q'_{\mathcal{O}},\Sigma} \not\subseteq cert^D_{q_{\mathcal{O}},\Sigma}$. It follows that there is a tuple of constants $\vec{c} = (c_1, \ldots, c_n)$ such that $\vec{c} \in cert^D_{q'_{\mathcal{O}},\Sigma}$ but, at the same time, $\vec{c} \notin cert^D_{q_{\mathcal{O}},\Sigma}$. Consider now $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$, i.e., the canonical structure of $\mathcal{O}$ with respect to $\mathcal{M}$ and D (cf. Subsection 2.6.3). Notice that, since $\mathcal{M}$ is a GAV mapping and $\mathcal{O}$ is a *DL-Lite*$_{\mathsf{RDFS}}$ ontology, we have that: (i) $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ does not introduce variables, and therefore we can see it as a set of facts $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)} = \{\beta_1, \ldots, \beta_m\}$ over $\mathcal{O}$; and (ii) $cert^D_{q_{\mathcal{O}},\Sigma} = q_{\mathcal{O}}^{\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}}$ for each $\mathcal{S}$-database D. We now exhibit an $\mathcal{S}$-database D' for which (i) $\vec{c} \notin q_{\mathcal{S}}^{D'}$, and (ii) $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)} \subseteq \mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D')}$. To this aim, we exploit the boolean query q_β over $\mathcal{O}$ associated to $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$, i.e., the following boolean CQ:

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

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

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

Obviously, by construction, we have $\vec{c} \in q^{\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}}$. Furthermore, due to the fact that $\mathcal{M}$ is a pure GAV mapping and the fact that $\mathcal{O}$ contains no assertions with $\exists R$ in the right-hand side for a basic role R, each possible disjunct r_q of $\mathsf{PerfRef}_{q,\Sigma}$ has a corresponding disjunct r of $\mathsf{PerfRef}_{q_\beta,\Sigma}$ in which the body of r is obtained from the body of r_q by replacing the distinguished variables x_c (respectively, the existential variable y_c) occurring in q with constant c. Notice that, by assumption, each disjunct r of $\mathsf{PerfRef}_{q_\beta,\Sigma}$ is such that $\vec{c} \in q_{\mathcal{S}}^{D_r}$, i.e., for each disjunct r of $\mathsf{PerfRef}_{q_\beta,\Sigma}$ there is a disjunct $q'_{\mathcal{S}}$ of $q_{\mathcal{S}}$ for which $\vec{c} \in {q'_{\mathcal{S}}}^{D_r}$. Since $q_{\mathcal{S}}$ is a UCQJFE, by exploiting Lemma 8.1, it is not hard to ascertain that this implies that for each disjunct r of $\mathsf{PerfRef}_{q_\beta,\Sigma}$ there is a disjunct $q'_{\mathcal{S}}$ of $q_{\mathcal{S}}$ for which for each atom α of $q'_{\mathcal{S}}$ there is an atom of r that instantiates α. By construction of q, however, the above property holds even if we replace q_β with q, i.e., for each disjunct r_q of $\mathsf{PerfRef}_{q,\Sigma}$ there is a disjunct $q'_{\mathcal{S}}$ of $q_{\mathcal{S}}$ for which for each atom α of $q'_{\mathcal{S}}$ there is an atom of r_q that instantiates α. Thus, by Lemma 8.1, we derive that q is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. But then, due to Lemma 8.3, from q it is possible to derive a CQ q' with same target list as q but whose body is the conjunction of at most $bound(\mathcal{M}, q_{\mathcal{S}})$ atoms occurring in q and such that q' is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. By construction of the algorithm, however, it can be readily seen that such a CQ q' is a disjunct of $q_{\mathcal{O}}$. Two considerations are now in order: (i) since q' has the same target list as q, and since its body is constituted only by a subset of the atoms of q, the fact that $\vec{c} \in q^{\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}}$ implies $\vec{c} \in q'^{\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}}$ as well, and (ii) since $\vec{c} \in q'^{\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}}$ and since q' is a disjunct of $q_{\mathcal{O}}$, we have $\vec{c} \in q_{\mathcal{O}}^{\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}}$. Thus, since in this setting for OBDM specifications $cert^{D}_{q_{\mathcal{O}},\Sigma} = q_{\mathcal{O}}^{\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}}$ for each UCQ $q_{\mathcal{O}}$ and $\mathcal{S}$-database D, we derive that $\vec{c} \in cert^{D}_{q_{\mathcal{O}},\Sigma}$, which is a contradiction on the initial assumption that $\vec{c} \notin cert^{D}_{q_{\mathcal{O}},\Sigma}$. It follows that the former case just considered is not possible because it leads to a contradiction. Therefore, we consider only the latter case.

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

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

Regarding the cost of the algorithm, we observe that the overall running time is exponential in the size of the input. Notice, moreover, that CQs over $\mathcal{O}$ may have an exponential number of atoms with respect to $\eta(q_{\mathcal{S}})$. Next we prove that (i) unless PTIME = NP, the computation problem for sound source-to-ontology rewritings can not be solved in polynomial time, even in the restricted scenario for UCQJFEs; and (ii) there exists OBDM specifications Σ and UCQJFEs $q_{\mathcal{S}}$ for which the unique (up to equivalence w.r.t. Σ) UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ is a CQ whose number of atoms is necessarily exponential with respect to $\eta(q_{\mathcal{S}})$.

Proposition 8.1. *Assuming* PTIME $\subsetneq$ *NP, the computation problem for sound source-to-ontology rewritings in the restricted scenario for UCQJFEs can not be solved in polynomial time.*

Proof. Let F be a 3-DNF formula on a set of variables $X = \{x_1, \dots, x_n\}$. Consider the OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, the UCQJFE $q_{\mathcal{S}}$, and the CQ $q_{\mathcal{O}}$ constructed from F as illustrated in the reduction of the lower bound proof of Theorem 8.1.

In this case, it is straightforward to verify that the unique (up to equivalence w.r.t. Σ) UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ is either the CQ $q_{\mathcal{O}}$ if it is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, or the CQ $q'_{\mathcal{O}} = \{(x_1, \dots, x_n) \mid \bot(x_1, \dots, x_n)\}$ otherwise. Specifically, as shown in that coNP-hardness proof of Theorem 8.1, $q_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ if and only formula F is valid, and therefore, due to the above observation, $q_{\mathcal{O}}$ is the unique (up to equivalence w.r.t. Σ) UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ if and only if formula F is valid.

We have therefore reduced the problem of checking the validity of a 3-DNF formula F to the problem of computing the unique (up to equivalence w.r.t. Σ) UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of a UCQJFE $q_{\mathcal{S}}$, where both $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and $q_{\mathcal{S}}$ can be constructed in LogSpace from F.

Thus, even in the restricted scenario for UCQJFEs, a polynomial time algorithm for computing UCQ-maximally sound source-to-ontology rewritings of UCQJFEs would imply a polynomial time algorithm for checking whether a 3-DNF formula is valid. Since this latter problem is known to be in general coNP-hard, it follows that, unless PTime = NP, the computation problem for sound source-to-ontology rewritings can not be solved in polynomial time, even in the restricted scenario for UCQJFEs. □

Proposition 8.2. *In the restricted scenario for UCQJFEs, there are OBDM specifications Σ and UCQJFEs $q_{\mathcal{S}}$ for which the unique (up to equivalence w.r.t. Σ) UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ is a CQ whose number of atoms in its body is necessarily exponential with respect to $\eta(q_{\mathcal{S}})$.*

Proof. We provide here a small example showing the main reason of why the number of atoms in the body of the unique (up to equivalence w.r.t. the OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$) UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of a UCQJFE $q_{\mathcal{S}}$ may be exponential with respect to $\eta(q_{\mathcal{S}})$.

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

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

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

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

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

In the next section, we will see that the bound on the number of atoms of Lemma 8.3 becomes polynomial (rather than exponential) with respect to $\eta(q_{\mathcal{S}})$ when queries $q_{\mathcal{S}}$ are CQJFEs (rather than unions thereof).

8.2 Restricted Scenario for CQJFEs

In this section, we study both the verification, and the computation problem for sound source-to-ontology rewritings in the restricted scenario for CQJFEs. We recall that, while the language $\mathcal{L}_{\mathcal{S}}$ of queries over source schemas $\mathcal{S}$ is the one of CQJFEs, the language $\mathcal{L}_{\mathcal{O}}$ for queries over ontologies $\mathcal{O}$ remains the one of UCQs.

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

Definition 8.2. Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification, β be an atom over $\mathcal{O}$, and α be an atom occurring in the body of a CQ $q_{\mathcal{S}} = \{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$ over $\mathcal{S}$. We say that β Σ-*covers* α, if the following holds: in each disjunct of $\rho(\beta, \Sigma)$ there is at least an atom α' such that α' instantiates α.

Example 8.5. Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the following OBDM specification:

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

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

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

Obviously, for an OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, an atom β over $\mathcal{O}$, and an atom α over $\mathcal{S}$, checking whether β Σ-covers α can be done in polynomial time.

8.2.1 Verification Problem

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

Lemma 8.4. *Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification, and let $q_{\mathcal{S}}$ and $q_{\mathcal{O}}$ be a CQJFE over $\mathcal{S}$ and a CQ over $\mathcal{O}$, respectively, with the same target list. We have that $q_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ if and only if it is the case that for each atom α of $q_{\mathcal{S}}$ there exists an atom β of $q_{\mathcal{O}}$ such that β Σ-covers α.*

Proof. "**Only-if part:**" Suppose, for the sake of contradiction, that there exists an atom α of $q_{\mathcal{S}}$ such that for no atom β of $q_{\mathcal{O}}$ it is the case that β Σ-covers α. Let q' be the disjunct of $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ obtained by unfolding each atom β of $q_{\mathcal{O}}$ with the disjunct of $\rho(\beta, \Sigma)$ in which there is no atom α' that instantiates α. For each atom β of $q_{\mathcal{O}}$, there is at least one disjunct among the ones of $\rho(\beta, \Sigma)$ that satisfies this condition, otherwise, following Definition 8.2, we would trivially derive a contradiction on the fact that β does not Σ-cover α.

Thus, the disjunct q' of $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ contains no atom that instantiates the atom α of $q_{\mathcal{S}}$, and therefore, due to Lemma 8.1, this implies that $q' \not\sqsubseteq q_{\mathcal{S}}$. It follows that $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \not\sqsubseteq q_{\mathcal{S}}$, which, due to Corollary 8.1, in turn implies that $q_{\mathcal{O}}$ is not a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, as required.

"**If part:**" Since for each atom α of $q_{\mathcal{S}}$ there is an atom β such that β Σ-covers α, we trivially derive that each possible disjunct q' of $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$ satisfies the following condition: for each atom α of $q_{\mathcal{S}}$, there is an atom α' of q' such that α' instantiates α. Due to Lemma 8.1, it follows that that $q' \sqsubseteq q$. Since this is true for each possible disjunct q' of $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma}$, we further derive that $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \sqsubseteq q_{\mathcal{S}}$, which, due to Corollary 8.1, implies that $q_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, as required. $\square$

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

Theorem 8.3. *In the restricted scenario for CQJFEs, the verification problem is in* PTIME.

Proof. Due to Corollary 8.1, it is sufficient to show how to check the containment $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \sqsubseteq q_{\mathcal{S}}$ in polynomial time, where $q_{\mathcal{O}}$ and $q_{\mathcal{S}}$ are a UCQ over $\mathcal{O}$ and a CQJFE over $\mathcal{S}$, respectively. To start, note that by construction $q_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ if and only if each disjunct q of $q_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, i.e., $\mathsf{PerfRef}_{q_{\mathcal{O}},\Sigma} \sqsubseteq q_{\mathcal{S}}$ if and only if $\mathsf{PerfRef}_{q,\Sigma} \sqsubseteq q_{\mathcal{S}}$ for each disjunct q of $q_{\mathcal{O}}$. It is therefore enough to show that, given a CQ $q = \{\vec{t'} \mid \exists \vec{y'}.\phi'(\vec{x'}, \vec{y'})\}$ over $\mathcal{O}$ and a CQJFE $q_{\mathcal{S}} = \{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$ over $\mathcal{S}$, checking whether $\mathsf{PerfRef}_{q,\Sigma} \sqsubseteq q_{\mathcal{S}}$ can be done in polynomial time.

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

Consider the query $q'_{\mathcal{S}}$ obtained in polynomial time from $q_{\mathcal{S}}$ by replacing every occurrence of the term t_i in $q_{\mathcal{S}}$ (even in the target list) with the term $f(t_i) = t'_i$, for each $i \in [1, n]$. Observe that now $q'_{\mathcal{S}}$ is a CQJFE having the same target list $\vec{t'}$ of q. By virtue of Lemma 8.4, we can now check whether q is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q'_{\mathcal{S}}$ by checking whether it is the case that for each atom α of q' there exists an atom β of q such that β Σ-covers α. This can be clearly done in polynomial time.

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

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

8.2.2 Computation Problem

As for the computation problem, we now provide an algorithm for computing UCQ-maximally sound source-to-ontology rewritings which avoids the enumeration of all possible CQs over $\mathcal{O}$ of a certain bound as algorithm $\mathsf{MaximallySoundUCQJFEs}$ in the previous section does. The computation of the returned UCQ over $\mathcal{O}$ is rather guided by the atoms occurring in the input query $q_{\mathcal{S}}$, in a very similar fashion to the *bucket algorithm* [Levy *et al.*, 1996] used for rewriting queries using views.

Specifically, by exploiting Lemma 8.4, the idea is as follows: for each atom α_i occurring in the body of $q_{\mathcal{S}}$, we compute a set B_i containing all the atoms β over $\mathcal{O}$ such that β Σ-cover α. Then, disjuncts of the final UCQ $q_{\mathcal{O}}$ over $\mathcal{O}$ are constructed by simply selecting atoms from each set B_i and conjoining them.

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

Example 8.6. Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the following OBDM specification:

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

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

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

Notice that, however, if we consider more specific queries obtained by applying some equalities to the query $q_{\mathcal{S}}$, such as $q^1_{\mathcal{S}} = \{(x_1, x_1) \mid s(x_1, x_1)\}$ (obtained by applying the equality $x_1 = x_2$) and $q^2_{\mathcal{S}} = \{(x_1, c_1) \mid s(x_1, c_1)\}$ (obtained by applying the equality $x_2 = c_1$), then we get that atom $A(x_1)$ Σ-covers $s(x_1, x_1)$ and atom $A'(x_1)$ Σ-covers $s(x_1, c_1)$. As a result, due to Lemma 8.4, queries $q^1_{\mathcal{O}} = \{(x_1, x_1) \mid A(x_1)\}$ and $q^2_{\mathcal{O}} = \{(x_1, c_1) \mid A'(x_1)\}$ are a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q^1_{\mathcal{S}}$ and $q^2_{\mathcal{S}}$, respectively. It follows that, since by construction $q^i_{\mathcal{S}} \sqsubseteq q_{\mathcal{S}}$ for both $i = 1$ and $i = 2$, both $q^1_{\mathcal{O}}$ and $q^2_{\mathcal{O}}$ are sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewritings of $q_{\mathcal{S}}$ as well.

Furthermore, when applying equalities, we have to take into account not only constants occurring in mapping assertions $\mathcal{M}$, but also constants occurring in the body of the input query. Consider indeed the CQJFE $q'_{\mathcal{S}} = \{(x) \mid s(x, c_2)\}$ over $\mathcal{S}$. Observe that there is no atom β over $\mathcal{O}$ such that β Σ-covers $s(x, c_2)$. Notice, however, that if we consider the query $q^3_{\mathcal{S}} = \{(c_2) \mid s(c_2, c_2)\}$ (obtained by applying the equality $x = c_2$ to $q'_{\mathcal{S}}$), then we get that atom $A(c_2)$ Σ-covers $s(c_2, c_2)$. As a result, due to Lemma 8.4, the query $q^3_{\mathcal{O}} = \{(c_2) \mid A(c_2)\}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q^3_{\mathcal{S}}$, and therefore a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q'_{\mathcal{S}}$ since $q^3_{\mathcal{S}} \sqsubseteq q'_{\mathcal{S}}$. □

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

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

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

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

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

The following example illustrates an execution of the $\mathsf{HeadCompletion}$ algorithm.

Example 8.7. Let $q_{\mathcal{S}}$ be the following CQ $q_{\mathcal{S}} = \{(x_1, x_2) \mid \exists y.s_1(x_1, c_2, y) \wedge s_2(x_1, x_2)\}$, and let con be the following set of constants $\mathsf{con} = \{c_1, c_2\}$. One can verify that $\mathsf{HeadCompletion}(q_{\mathcal{S}}, \mathsf{con})$ returns the UCQ $Q = \bigcup_{1 \leq i \leq 10} q^i_{\mathcal{S}}$, where:

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

Algorithm 8.2 HeadCompletion

Input:
CQ q;
set of constants con

Output:
UCQ Q

$Q := q$
repeat
  $Q' := Q$
  for each CQ $q' \in Q'$ **do**
    Let $q' = \{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$
    for each pair of compatible terms t_1, t_2 in $\vec{t} \cup \mathsf{con}$ **do**
      $Q := Q \cup q'[t_1/t_2]$
    end for
  end for
until $Q' = Q$
return Q

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

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

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

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

Example 8.8. Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the following OBDM specification:

- $\mathcal{O} = \emptyset$

Algorithm 8.3 MaximallySoundCQJFEs

Input:

OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ where $\mathcal{O}$ is a *DL-Lite*$_{\mathsf{RDFS}}$ ontology and $\mathcal{M}$ is a pure GAV mapping;
CQJFE $q_{\mathcal{S}}$ over $\mathcal{S}$

Output:

UCQ $q_{\mathcal{O}}$

$q_{\mathcal{O}} := \bot$
$\mathsf{con} := \mathsf{con}_{\mathcal{M}} \cup \mathsf{con}_{q_{\mathcal{S}}}$
for each CQ $q \in \mathsf{HeadCompletion}(q_{\mathcal{S}}, \mathsf{con})$ **do**
    Let $q = \{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$, where $\phi(\vec{x}, \vec{y}) = \alpha_1 \wedge \ldots \wedge \alpha_{\eta(q_{\mathcal{S}})}$
    for $i \leftarrow 1$ to $\eta(q_{\mathcal{S}})$ **do**
        $B_i := \emptyset$
        for each possible atom β over $\mathcal{O}$ having as arguments the terms occurring in α_i and possibly fresh existential variables **do**
            if β Σ-covers α_i **then**
                $B_i := B_i \cup \beta$
            end if
        end for
    end for
    for each combinations of atoms $(\beta_1, \ldots, \beta_{\eta(q_{\mathcal{S}})}) \in B_1 \times \ldots \times B_{\eta(q_{\mathcal{S}})}$ **do**
        $q_{\mathcal{O}} := q_{\mathcal{O}} \cup \{\vec{t} \mid \exists \vec{y'}.\phi'(\vec{x}, \vec{y'})\}$, where $\phi'(\vec{x}, \vec{y'}) = \beta_1 \wedge \ldots \wedge \beta_{\eta(q_{\mathcal{S}})}$
    end for
end for
return $q_{\mathcal{O}}$

- $\mathcal{S} = \{\ s_1, s_2, s_3, s_4, s_5, s_6\ \}$
- $\mathcal{M} = \{\ m_1, m_2, m_3, m_4, m_5, m_6\ \}$

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

Let $q_{\mathcal{S}}$ be the CQJFE illustrated in Example 8.7. As a first step, the algorithm computes $\mathsf{HeadCompletion}(q_{\mathcal{S}}, \mathsf{con})$ which, since $\mathsf{con}_{\mathcal{M}} = \{c_1\}$ and $\mathsf{con}_{q_{\mathcal{S}}} = \{c_2\}$, it turns out to be the UCQ $Q = \bigcup_{1 \leq i \leq 10} q^i_{\mathcal{S}}$ illustrated in Example 8.7, where $\mathsf{con} = \mathsf{con}_{\mathcal{M}} \cup \mathsf{con}_{q_{\mathcal{S}}} = \{c_1, c_2\}$.

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

As for the query $q_{\mathcal{S}}^2 = \{(x_1, x_1) \mid \exists y.s_1(x_1, c_2, y) \wedge s_2(x_1, x_1)\}$, we have $B_1^2 = \{P_1(x_1, c_2)\}$ and $B_2^2 = \{A_1(x_1), A_2(x_1)\}$. Thus, the CQs $q_{\mathcal{O}}^1 = \{(x_1, x_1) \mid P_1(x_1, c_2) \wedge A_1(x_1)\}$ and $q_{\mathcal{O}}^2 = \{(x_1, x_1) \mid P_1(x_1, c_2) \wedge A_1(x_1)\}$ are disjuncts of the final UCQ $q_{\mathcal{O}}$.

As for the query $q_{\mathcal{S}}^3 = \{(x_1, c_1) \mid \exists y.s_1(x_1, c_2, y) \wedge s_2(x_1, c_1)\}$, we have $B_1^3 = \{P_1(x_1, c_2)\}$ and $B_2^3 = \{P_3(x, y')\}$. Thus, the CQ $q_{\mathcal{O}}^3 = \{(x_1, c_1) \mid \exists y'.P_1(x_1, c_1) \wedge P_3(x_1, y')\}$ is a disjunct of the final UCQ $q_{\mathcal{O}}$.

For the queries $q_{\mathcal{S}}^j$ with $j = [8, 9, 10]$, we observe that all disjuncts over $\mathcal{O}$ generated by the algorithm are subsumed (w.r.t. Σ) by $q_{\mathcal{O}}^i$ for some $i = [1, 2, 3]$. As a conclusion, one can verify that $\mathsf{MaximallySoundCQJFEs}(\Sigma, q_{\mathcal{S}})$ returns a UCQ that is equivalent (w.r.t. Σ) to $q_{\mathcal{O}} = q_{\mathcal{O}}^1 \cup q_{\mathcal{O}}^2 \cup q_{\mathcal{O}}^3$, which is the unique (up to equivalence w.r.t. Σ) UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. $\square$

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

Theorem 8.4. *In the restricted scenario for CQJFEs, MaximallySoundCQJFEs $(\Sigma, q_{\mathcal{S}})$ terminates and returns the unique (up to equivalence w.r.t. Σ) UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.*

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

As for the correctness, we first show that the computed $q_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. By construction, each disjunct $\{\vec{t} \mid \exists \vec{y'}.\phi'(\vec{x}, \vec{y'})\}$ of $q_{\mathcal{O}}$ satisfies the following condition: there is a query $q \in \mathsf{HeadCompletion}(q_{\mathcal{S}}, \mathsf{con})$ with target list $\vec{t}$ such that for each atom α_i of q there is an atom β_i of $\phi(\vec{x}, \vec{y'})$ that Σ-covers α_i, where $\mathsf{con} = \mathsf{con}_{\mathcal{M}} \cup \mathsf{con}_{q_{\mathcal{S}}}$. Due to Lemma 8.4, this implies that $\{\vec{t} \mid \exists \vec{y'}.\phi'(\vec{x}, \vec{y'})\}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of a query $q \in \mathsf{HeadCompletion}(q_{\mathcal{S}}, \mathsf{con})$. Since for each query $q \in \mathsf{HeadCompletion}(q_{\mathcal{S}}, \mathsf{con})$ we trivially have that $q \sqsubseteq q_{\mathcal{S}}$, we derive that $\{\vec{t} \mid \exists \vec{y'}.\phi'(\vec{x}, \vec{y'})\}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ as well. Furthermore, since the above condition is true for each disjunct $\{\vec{t} \mid \exists \vec{y'}.\phi'(\vec{x}, \vec{y'})\}$ of $q_{\mathcal{O}}$, it follows that the computed $q_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. We now show that $q_{\mathcal{O}}$ is actually the unique (up to equivalence w.r.t. Σ) UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, that is, each UCQ $q'_{\mathcal{O}}$ that is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ is such that $cert_{q'_{\mathcal{O}},\Sigma} \sqsubseteq cert_{q_{\mathcal{O}},\Sigma}$ (cf. Definition 3.6). We do this by way of contradiction.

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

$$\Gamma = \bigcup_{h \in H} h(q_{\mathcal{S}})$$

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

- Let $q = \{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\} = \{\vec{t} \mid \exists \vec{y}.\alpha_1 \wedge \ldots \wedge \alpha_{\eta(q_{\mathcal{S}})}\} \in \mathsf{HeadCompletion}(q_{\mathcal{S}}, \mathsf{con})$ be the most specific query over $\mathcal{S}$ for which $\vec{c} = (c_1, \ldots, c_n) \in q^D$ still holds, i.e., the CQ whose target list $\vec{t} = (t_1, \ldots, t_n)$ is such that (i) for each $i \in [1, n]$, if c_i is a constant occurring either in $\mathsf{con}_{q_{\mathcal{S}}}$ or in $\mathsf{con}_{\mathcal{M}}$, then $t_i = c_i$ (otherwise term t_i is a distinguished variable), and (ii) for each pair of numbers $i, j \in [1, n]$, $c_i = c_j$ if and only if $t_i = t_j$.
- Consider the target list $\vec{t} = (t_1, \ldots, t_n)$ of q and the tuple of constants $\vec{c} = (c_1, \ldots, c_n)$. Let $\Lambda' = \{\beta'_1, \ldots, \beta'_k\}$ be the set of atoms over $\mathcal{O}$ obtained from the set of facts $\Lambda = \{\beta_1, \ldots, \beta_k\}$ by (i) replacing everywhere the constant $c_i \in \vec{c}$ with the term $t_i \in \vec{t}$ (either a distinguished variable, or the constant c_i itself), for each $i \in [1, n]$, and (ii) replacing everywhere each constant c occurring neither in $q_{\mathcal{S}}$ nor in $\mathcal{M}$ with a fresh existential variable y_c.

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

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

Consider the latter case, that is, there exists at least an atom α_i of q for which no atom $\beta' \in \Lambda'$ is such that β' Σ-covers α_i. It is not hard to ascertain that this implies that there is at least an atom α'_i of $q_{\mathcal{S}}(\vec{c})$ for which no atom $\beta \in \Lambda$ is such that β Σ-covers α'_i, where we recall that $q_{\mathcal{S}}(\vec{c}) = \{() \mid \exists \vec{y}.\alpha'_1 \wedge \ldots \wedge \alpha'_{\eta(q_{\mathcal{S}})}\}$ is the boolean CQ obtained from $q_{\mathcal{S}}$ by replacing each occurrence of term t'_i in the body of $q_{\mathcal{S}}$ with constant c_i, for each $i \in [1, n]$ (where $\vec{t'} = (t_1, \ldots, t_n)$ is the target list of $q_{\mathcal{S}}$). But then, consider the set of facts Ω obtained by unfolding each fact $\beta \in \Lambda$ with a disjunct of $\rho(\beta, \Sigma)$ such that there is no atom over $\mathcal{S}$ that instantiates α' (clearly, since by assumption β does not Σ-cover α', following Definition 8.2, at least one of such disjunct must exists). As a result, we trivially have that $\Omega \not\models q_{\mathcal{S}}(\vec{c})$, which implies that $\vec{c} \notin q_{\mathcal{S}}^{\Omega}$.

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

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

As a specialisation of Lemma 8.3 in the restricted scenario for CQJFEs, we have the following result which straightforward follows from the above theorem.

Corollary 8.2. *Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification, and let $q_{\mathcal{S}}$ be a CQJFE over $\mathcal{S}$. If a CQ $q_{\mathcal{O}}$ over $\mathcal{O}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, then there exists a CQ $q'_{\mathcal{O}}$ over $\mathcal{O}$ with same target list of $q_{\mathcal{O}}$ such that (*i*) the body of $q'_{\mathcal{O}}$ is the conjunction of at most $\eta(q_{\mathcal{S}})$ atoms occurring in the body of $q_{\mathcal{O}}$ (and therefore, $cert_{q_{\mathcal{O}},\Sigma} \sqsubseteq cert_{q'_{\mathcal{O}},\Sigma}$), and (*ii*) $q'_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ as well.*

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

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

Proposition 8.3. *In the restricted scenario for CQJFEs, there are OBDM specifications Σ and CQJFEs $q_{\mathcal{S}}$ for which the unique (up to equivalence w.r.t. Σ) UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ is a UCQ having necessarily an exponential number of disjuncts with respect to $\eta(q_{\mathcal{S}})$, for exponentially many source queries with respect to the size of the target list of $q_{\mathcal{S}}$.*

Proof. We provide here a small example showing the main reason of why the unique (up to equivalence w.r.t. the OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$) UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of a CQJFE $q_{\mathcal{S}}$ may contain an exponential number of disjuncts with respect to $\eta(q_{\mathcal{S}})$, for exponentially many source queries with respect to the size of the target list of $q_{\mathcal{S}}$.

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

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

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

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

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

One can verify that for each of the CQs illustrated above there are at least eight CQs over $\mathcal{O}$ that must necessarily appear in the unique (up to equivalence w.r.t. Σ) UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. For instance, consider the query $q^6_{\mathcal{S}}$ in case 6. We have that $B_1 = \{P_{1,1}(x_1,x_1), P_{1,2}(x_1,x_1), A_{1,1}(x_1), A_{1,2}(x_1)\}$, $B_2 = \{P_{2,1}(x_3,x_4), P_{2,2}(x_3,x_4)\}$, and $B_3 = \{P_{3,1}(x_5,x_5), P_{3,2}(x_5,x_5), A_{3,1}(x_5), A_{3,2}(x_5)\}$ are the set of all the atoms that Σ-cover $s_1(x_1,x_1)$, $s_2(x_1,x_2)$, and $s_3(x_5,x_5)$, respectively. In particular, if we consider the subsets $B'_1 = \{A_{1,1}(x_1), A_{1,2}(x_1)\}$ and $B'_3 = \{A_{3,1}(x_1), A_{3,2}(x_5)\}$ of B_1 and B_3, respectively, then it is easy to verify that for each possible combination of atoms $(\beta_1, \beta_2, \beta_3)$ occurring in the Cartesian Product

$B'_1 \times B_2 \times B'_3$, we have that the CQ $\{(x_1, x_1, x_3, x_4, x_5, x_5) \mid \exists \vec{y'}. \beta_1 \wedge \beta_2 \wedge \beta_3\}$ is necessarily a disjunct of the unique (up to equivalence w.r.t. Σ) UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, and, moreover, each of these CQs will not be produced when considering any other query in $\mathsf{HeadCompletion}(q_{\mathcal{S}}, \{\})$.

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

8.3 View-based Query Processing in the presence of Disjunctive Views

In Subsection 3.3.3, we have figured out the relationship between the notion of source-to-ontology rewriting and the view-based query processing approach. Specifically, for OBDM specifications $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ with $\mathcal{M}$ being a pure GAV mapping and $\mathcal{O} = \emptyset$, the problem of computing a perfect (respectively, UCQ-maximally sound) $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of a UCQ $q_{\mathcal{S}}$ is equivalent to the problem of computing an exact (respectively, UCQ-maximally sound) rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}_{\mathcal{M}}$, where $\mathcal{V}_{\mathcal{M}}$ denotes the set of UCQ view definitions associated to mapping $\mathcal{M}$ (cf. Theorem 3.2 and Corollary 3.1).

We are now ready to focus on the problem of computing UCQ-maximally sound rewritings of UCQs $q_{\mathcal{S}}$ with respect to sets of UCQ view definitions $\mathcal{V}$. Using results proven so far, in this section we delineate the precise dividing line between the existence and the non-existence cases along the dimension of join existential variables occurring in the bodies of the various disjuncts of $q_{\mathcal{S}}$.

As already sketched in Example 3.10 and formally proven in [Duschka and Genesereth, 1998; Afrati and Chirkova, 2019], there are pairs of sets of UCQ view definitions $\mathcal{V}$ and CQs $q_{\mathcal{S}}$ for which no UCQ-maximally sound rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$ exists. However, each CQ $q_{\mathcal{S}}$ occurring in these pairs used to prove such a negative result has more than one join existential variable occurring in its body. We can strengthen this negative result. Indeed, by combining the proof of point 1 of Theorem 6.2 and Corollary 3.1, we immediately obtain the following corollary.

Corollary 8.3. *There exists a set of UCQ view definitions $\mathcal{V}$ over a schema $\mathcal{S}$ and a boolean CQ $q_{\mathcal{S}}$ over $\mathcal{S}$ with only one variable occurring in its body for which no UCQ-maximally sound rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$ exists. Furthermore, both $\mathcal{V}$ and $\mathcal{S}$ use only predicates with arity at most two, and all view definitions in $\mathcal{V}$ are single CQs except one view definition which is the union of only two CQs.*

On the contrary, using results presented in this chapter, we are able to show that having no join existential variables occurring in the body of queries $q_{\mathcal{S}}$ is a sufficient condition that guarantees the existence of UCQ-maximally sound rewritings of

UCQs $q_{\mathcal{S}}$ with respect to set of UCQ view definitions $\mathcal{V}$, i.e., UCQ-maximally sound rewritings of UCQJFEs $q_{\mathcal{S}}$ with respect to sets UCQ view definitions $\mathcal{V}$ always exist.

Towards this goal, we consider OBDM specifications $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ where $\mathcal{M}$ is a pure GAV mapping and $\mathcal{O} = \emptyset$ is not a usual ontology, but it is simply a schema (without assertions) whose predicates have arity possibly greater than 2. It is straightforward to argue that Algorithm MaximallySoundUCQJFEs (respectively, Algorithm MaximallySoundCQJFEs) can be used for computing UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewritings of UCQJFEs (respectively, CQJFEs) $q_{\mathcal{S}}$ also for this kind of OBDM specifications $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$.

Let $\mathcal{V} = \{V_1, \ldots, V_n\}$ be a set of UCQ view definitions over a schema $\mathcal{S}$. We denote by $\Sigma_{\mathcal{V}} = \langle \mathcal{O}_{\mathcal{V}}, \mathcal{S}, \mathcal{M}_{\mathcal{V}} \rangle$ the OBDM specification where: (i) for each $i \in [1, n]$, the schema $\mathcal{O}_{\mathcal{V}}$ comprises a predicate V_i of the same arity of the UCQ associated to symbol V_i in the view definitions $\mathcal{V}$; (ii) $\mathcal{M}_{\mathcal{V}}$ is the pure GAV mapping relating schema $\mathcal{S}$ to schema $\mathcal{O}_{\mathcal{V}}$ obtained by including, for each $i \in [1, n]$ and for each disjunct $\{\vec{x} \mid \exists \vec{y}. \phi(\vec{x}, \vec{y})\}$ occurring in the UCQ associated to the symbol V_i, the mapping assertion $\exists \vec{y}. \phi(\vec{x}, \vec{y}) \rightarrow V_i(\vec{x})$.

Example 8.9. Let $\mathcal{V} = \{V_1, V_2\}$ be the set of UCQ view definitions over schema $\mathcal{S} = \{s_1, s_2, s_3, s_4\}$, where:

- $V_1 = \{(x_1, x_2, x_3) \mid s_3(x_1, x_2, x_3)\} \cup \{(x_1, x_2, x_3) \mid \exists y. s_1(x_1, y) \wedge s_2(y, x_2, x_3)\} \cup \{(x_1, x_2, x_3) \mid \exists y_1, y_2. s_1(x_1, y_1) \wedge s_4(x_1, x_2, x_3, y_2)\}$
- $V_2 = \{(x_1, x_2, x_3, x_4) \mid s_1(x_1, x_2) \wedge s_3(x_2, x_3, x_4)\} \cup \{(x_1, x_2, x_3, x_4) \mid \exists y. s_2(x_1, x_2, y) \wedge s_4(y, x_3, x_3, x_4)\}$

Then, the OBDM specification $\Sigma_{\mathcal{V}} = \langle \mathcal{O}_{\mathcal{V}}, \mathcal{S}, \mathcal{M}_{\mathcal{V}} \rangle$ is such that $\mathcal{O}_{\mathcal{V}}$ is a schema with a ternary predicate V_1 and a quaternary predicate V_2, and $\mathcal{M}_{\mathcal{V}} = \{m_1, m_2, m_3, m_4, m_5\}$ is a pure GAV mapping relating $\mathcal{S}$ to $\mathcal{O}_{\mathcal{V}}$, where:

$$
\begin{array}{lrcl}
m_1: & s_3(x_1, x_2, x_3) & \rightarrow & V_1(x_1, x_2, x_3), \\
m_2: & \exists y. s_1(x_1, y) \wedge s_2(y, x_2, x_3) & \rightarrow & V_1(x_1, x_2, x_3), \\
m_3: & \exists y_1, y_2. s_1(x_1, y_1) \wedge s_4(x_1, x_2, x_3, y_2) & \rightarrow & V_1(x_1, x_2, x_3), \\
m_4: & s_1(x_1, x_2) \wedge s_3(x_2, x_3, x_4) & \rightarrow & V_2(x_1, x_2, x_3, x_4), \\
m_5: & \exists y. s_2(x_1, x_2, y) \wedge s_4(y, x_3, x_3, x_4) & \rightarrow & V_2(x_1, x_2, x_3, x_4).
\end{array}
$$

□

We now provide the following theorem, which can be seen as the dual of Theorem 3.2 and can be shown using exactly the same arguments used in that proof.

Theorem 8.5. *Let $\mathcal{V}$ be a set of UCQ view definitions over a schema $\mathcal{S}$, and let $q_{\mathcal{S}}$ and $q_{\mathcal{V}}$ be two UCQs over $\mathcal{S}$ and over the view alphabet $\mathcal{V}$ (equivalently, $\mathcal{O}_{\mathcal{V}}$), respectively. We have that $q_{\mathcal{V}}$ is an exact (respectively, a sound) rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$ if and only if $q_{\mathcal{V}}$ is a perfect (respectively, sound) $\mathcal{S}$-to-$\mathcal{O}_{\mathcal{V}}$ $\Sigma_{\mathcal{V}}$-rewriting of $q_{\mathcal{S}}$.*

Proof. As already observed in the proof of Theorem 3.2:

1. By [Levy *et al.*, 1995], a UCQ $q_{\mathcal{V}}$ is an exact (respectively, a sound) rewriting of a UCQ $q_{\mathcal{S}}$ with respect to a set of UCQ view definitions $\mathcal{V}$ if and only if

$exp_{\mathcal{V}}(q_{\mathcal{V}}) \equiv q_{\mathcal{S}}$ (respectively, $exp_{\mathcal{V}}(q_{\mathcal{V}}) \sqsubseteq q_{\mathcal{S}}$), where $exp_{\mathcal{V}}(\cdot)$ is the function that, given a UCQ $q_{\mathcal{V}}$ over the view alphabet, replace each atom occurring in $q_{\mathcal{V}}$ by the definition of the views (being careful to use unique variables in place of those variables that appear in the bodies of the view but not in the heads of those), and then turning the resulting formula into an equivalent UCQ.

2. For OBDM specifications $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ with $\mathcal{O} = \emptyset$ and $\mathcal{M}$ a pure GAV mapping, a UCQ q is a perfect (respectively, sound) $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of a UCQ $q_{\mathcal{S}}$ if and only if $\mathsf{MapRef}(q, \mathcal{M}) \equiv q_{\mathcal{S}}$ (respectively, $\mathsf{MapRef}(q, \mathcal{M}) \sqsubseteq q_{\mathcal{S}}$), where $\mathsf{MapRef}(q, \mathcal{M})$ in this case is equivalent to *unfolding* the query q with respect to $\mathcal{M}$ [Poggi *et al.*, 2008], i.e., replacing each atom α occurring in q by the logical disjunction of all the left-hand sides of mapping assertions in $\mathcal{M}$ having the predicate name α in the right-hand side (being careful to use unique variables in place of those variables that appear in the left-hand side of the mapping assertions but not in the right-hand side of those), and then turning the resulting formula into an equivalent UCQ. Technically speaking, in [Poggi *et al.*, 2008] the unfolding is specified for ontologies $\mathcal{O}$, and thus whose schema comprises only unary and binary predicates (i.e., atomic concepts and atomic roles, respectively). We point out that, however, it can be straightforwardly generalised in the data integration context when $\mathcal{O}$ is any schema whose predicates have also arity greater than 2, see, e.g., [Lenzerini, 2002].

3. By construction, $exp_{\mathcal{V}}(q_{\mathcal{V}}) = \mathsf{MapRef}(q_{\mathcal{V}}, \mathcal{M}_{\mathcal{V}})$ for any set of UCQ view definitions $\mathcal{V}$ and for any UCQ $q_{\mathcal{V}}$ over the view alphabet $\mathcal{V}$ (equivalently, $\mathcal{O}_{\mathcal{V}}$).

Thus, $q_{\mathcal{V}}$ is an exact (respectively, a sound) rewriting of $q_{\mathcal{S}}$ if and only if $exp_{\mathcal{V}}(q_{\mathcal{V}}) \equiv q_{\mathcal{S}}$ (respectively, $exp_{\mathcal{V}}(q_{\mathcal{V}}) \sqsubseteq q_{\mathcal{S}}$), which, since $exp_{\mathcal{V}}(q_{\mathcal{V}}) = \mathsf{MapRef}(q_{\mathcal{V}}, \mathcal{M}_{\mathcal{V}})$, it is so if and only if $\mathsf{MapRef}(q_{\mathcal{V}}, \mathcal{M}_{\mathcal{V}}) \equiv q_{\mathcal{S}}$ (respectively, $\mathsf{MapRef}(q_{\mathcal{V}}, \mathcal{M}_{\mathcal{V}}) \sqsubseteq q_{\mathcal{S}}$), and therefore if and only if $q_{\mathcal{V}}$ is a perfect (respectively, sound) $\mathcal{S}$-to-$\mathcal{O}_{\mathcal{V}}$ $\Sigma_{\mathcal{V}}$-rewriting of $q_{\mathcal{S}}$, as required. □

From the sound part of the above theorem, we derive the following corollary, which can be seen as the dual of Corollary 3.1.

Corollary 8.4. *Let $\mathcal{V}$ be a set of UCQ view definitions over a schema $\mathcal{S}$, and let $q_{\mathcal{S}}$ and $q_{\mathcal{V}}$ be two UCQs over $\mathcal{S}$ and over the view alphabet $\mathcal{V}$ (equivalently, $\mathcal{O}_{\mathcal{V}}$), respectively. We have that $q_{\mathcal{V}}$ is a UCQ-maximally sound rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$ if and only if $q_{\mathcal{V}}$ is the unique (up to equivalence w.r.t. $\Sigma_{\mathcal{V}}$) UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}_{\mathcal{V}}$ $\Sigma_{\mathcal{V}}$-rewriting of $q_{\mathcal{S}}$.*

This, together with what illustrated in the previous sections, allow us to derive a technique to compute UCQ-maximally sound rewritings of UCQJFEs $q_{\mathcal{S}}$ with respect to sets of UCQ view definitions $\mathcal{V}$, thus proving that, for each pair composed by a set of UCQ view definitions $\mathcal{V}$ over a schema $\mathcal{S}$ and a UCQJFE $q_{\mathcal{S}}$ over $\mathcal{S}$, a UCQ-maximally sound rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$ is guaranteed to exists.

In particular, given a set of UCQ view definitions $\mathcal{V}$ over a schema $\mathcal{S}$ and a UCQJFE (respectively, CQJFE) $q_{\mathcal{S}}$ over $\mathcal{S}$, we can first construct $\Sigma_{\mathcal{V}}$ and then run the $\mathsf{MaximallySoundUCQJFEs}$ (respectively, $\mathsf{MaximallySoundCQJFEs}$) algorithm on $\Sigma_{\mathcal{V}}$ and $q_{\mathcal{S}}$. Due to Theorem 8.2 (respectively, Theorem 8.4), the UCQ returned by $\mathsf{MaximallySoundUCQJFEs}(\Sigma_{\mathcal{V}}, q_{\mathcal{S}})$ (respectively, $\mathsf{MaximallySoundCQJFEs}(\Sigma_{\mathcal{V}}, q_{\mathcal{S}})$) is the unique (up to equivalence w.r.t. $\Sigma_{\mathcal{V}}$) UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}_{\mathcal{V}}$ $\Sigma_{\mathcal{V}}$-rewriting of $q_{\mathcal{S}}$, and then, by Corollary 8.4, it is also a UCQ-maximally sound rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$.

By combining the above discussion with Theorem 3.3, we can easily obtain the main result of this section.

Theorem 8.6. *Let $\mathcal{V}$ be a set of UCQ view definitions over a schema $\mathcal{S}$, and let $q_{\mathcal{S}}$ be a UCQJFE (respectively, CQJFE) over $\mathcal{S}$. Let denote by $q_{\mathcal{V}}$ the UCQ over the schema $\mathcal{O}_{\mathcal{V}}$ (equivalently, view alphabet $\mathcal{V}$) returned by $\mathsf{MaximallySoundUCQJFEs}(\Sigma_{\mathcal{V}}, q_{\mathcal{S}})$ (respectively, $\mathsf{MaximallySoundCQJFEs}(\Sigma_{\mathcal{V}}, q_{\mathcal{S}})$). We have that:*

- *$q_{\mathcal{V}}$ is a UCQ-maximally sound rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$;*
- *$q_{\mathcal{V}}$ is a perfect rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$;*
- *$q_{\mathcal{V}}$ is a UCQ-exact rewriting of $q_{\mathcal{S}}$ wih respect to $\mathcal{V}$, if this latter exists.*

As an interesting implication observe that, given any set of UCQ view definitions $\mathcal{V}$ over a schema $\mathcal{S}$ and any UCQJFE $q_{\mathcal{S}}$ over $\mathcal{S}$, a perfect rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$ can be always expressed as a UCQ in which the body of each disjunct is the conjunction of at most $bound(\mathcal{M}_{\mathcal{V}}, q_{\mathcal{S}})$ atoms (cf. Lemma 8.3). Furthermore, when $q_{\mathcal{S}}$ is a CQJFE, the body of each disjunct is the conjunction of at most $\eta(q_{\mathcal{S}})$ atoms (cf. Corollary 8.2), exactly as in the case of CQ view definitions $\mathcal{V}$ and UCQs $q_{\mathcal{S}}$ [Levy *et al.*, 1995].

We conclude with the following curious observations: the assumption that the target list of each disjunct of the various UCQ view definitions in $\mathcal{V}$ does not have repeated variables or constants is essential for the above theorem to hold. In fact, when removing this assumption, by the proof of point 5 of Theorem 6.2 we can easily construct a set of UCQ view definitions $\mathcal{V}$ and a CQJFE $q_{\mathcal{S}}$ for which no UCQ-maximally sound rewriting of $q_{\mathcal{S}}$ exists. Furthermore, when removing this assumption, there are cases where a UCQ-maximally sound rewriting of a CQJFE $q_{\mathcal{S}}$ with respect to a set of CQ view definitions $\mathcal{V}$ exists but it does not correspond to a perfect rewriting of $q_{\mathcal{S}}$ with respect to $\mathcal{V}$ (cf. Example 3.9).

Chapter 9

Non-Monotonic Source-to-Ontology Rewritings

In this chapter, we investigate the notion of abstraction in the case where source-to-ontology rewritings can be expressed in a *non-monotonic query language*. One basic issue to address in this endeavour is selecting the non-monotonic query language.

Our choice in this chapter is to use *EQL-Lite*(UCQ) [Calvanese *et al.*, 2007a],[1] which is a language equipped with a single modal operator $\mathbf{K}$. The modal operator is used to formalise the epistemic state of the current OBDM system according to the minimal knowledge semantics (see later). Informally, the formula $\mathbf{K}\varrho$ is read as "ϱ is known to hold in the OBDM system". Queries in *EQL-Lite*(UCQ) can use conjunction, negation, and existential quantification, and have atoms that are expressed exactly as $\mathbf{K}\varrho$, where ϱ is a UCQ. With this combination of operators, it is possible to ask for those x such that a given $\phi(x)$ is not known to hold, and this is crucial for characterising a set of tuples that are not certain answers to a given source query $q_{\mathcal{S}}$. The epistemic operator enables also other interesting features. For instance, we can distinguish between asking for those x such that it is known that there is y for which $P(x, y)$ holds (where y can be unknown), and asking for those x such that there is y for which $P(x, y)$ is known to hold (and therefore y is known).

EQL-Lite(UCQ) is a particularly well-behaved fragment of *EQL*, a variant of the well-known *First-Order Modal Logic* of knowledge/belief [Levesque, 1984; Reiter, 1992; Levesque and Lakemeyer, 2000] (see also [Chellas, 1980]). Before exploring *EQL-Lite*(UCQ) source-to-ontology rewritings, we first recall the basis of *EQL*.

An *epistemic interpretation* for an ontology $\mathcal{O}$ is a pair $\langle E, \mathcal{I}\rangle$, where E is a possibly infinite set of FOL interpretations for $\mathcal{O}$, and $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}\rangle$ is an interpretation in E. We inductively define when an *EQL* sentence ψ is true in an epistemic interpretation $\langle E, \mathcal{I}\rangle$, written $\langle E, \mathcal{I}\rangle \models \psi$, as follows:

$$
\begin{array}{lll}
\langle E, \mathcal{I}\rangle \models A(c) & \text{iff} & \mathcal{I} \models A(c), \\
\langle E, \mathcal{I}\rangle \models P(c_1, c_2) & \text{iff} & \mathcal{I} \models P(c_1, c_2), \\
\langle E, \mathcal{I}\rangle \models \psi_1 \wedge \psi_2 & \text{iff} & \langle E, \mathcal{I}\rangle \models \psi_1 \text{ and } \langle E, \mathcal{I}\rangle \models \psi_2, \\
\langle E, \mathcal{I}\rangle \models \neg\psi & \text{iff} & \langle E, \mathcal{I}\rangle \not\models \psi, \\
\langle E, \mathcal{I}\rangle \models \exists x.\psi & \text{iff} & \langle E, \mathcal{I}\rangle \models \psi^x_c \text{ for some constant } c \in \Delta^{\mathcal{I}}, \\
\langle E, \mathcal{I}\rangle \models \mathbf{K}\psi & \text{iff} & \langle E, \mathcal{I}'\rangle \models \psi \text{ for every } \mathcal{I}' \in E,
\end{array}
$$

[1] In fact, we consider a restricted version of *EQL-Lite*(UCQ), in which (in)equalities are disallowed.

where ψ, x, and ψ_c^x denote an arbitrary *EQL* sentence, a variable, and the *EQL* sentence obtained by replacing the variable x with the constant c, respectively.

As in knowledge base scenarios, in OBDM, among the various epistemic interpretations, the interest is in the specific ones representing the *minimal epistemic state* of the OBDM system, i.e., the state in which the OBDM system has minimal knowledge. Namely: let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification and D be an $\mathcal{S}$-database. Then, a $\langle \Sigma, D \rangle$*-EQL-interpretation* is an epistemic interpretation $\langle E, \mathcal{I} \rangle$ for which $E = Mod_D(\Sigma)$. Finally, we say that an *EQL* sentence ψ is *EQL-logically implied* by $\langle \Sigma, D \rangle$, denoted by $\langle \Sigma, D \rangle \models_{EQL} \psi$, if for every $\langle \Sigma, D \rangle$-*EQL*-interpretation $\langle E, \mathcal{I} \rangle$ we have $\langle E, \mathcal{I} \rangle \models \psi_{\mathcal{I}}$, where $\psi_{\mathcal{I}}$ is obtained from ψ by replacing each constant c occurring in ψ with the domain object $c^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ (if defined, i.e., if $c \in \mathsf{dom}(D)$).

This led us to naturally define the certain answers of *EQL* queries in OBDM systems. First, we need to define *EQL* queries: an *EQL query* is a query of the form $q = \{\vec{t} \mid \psi(\vec{x})\}$, where the target list $\vec{t}$ is an n-tuple of terms, and the body $\psi(\vec{x})$ is an *EQL* formula in which the free variables are exactly the variables occurring in $\vec{t}$. For an *EQL* query $q = \{(t_1, \ldots, t_n) \mid \psi(\vec{x})\}$ of arity n and an n-tuple of constants $\vec{c} = (c_1, \ldots, c_n)$, we denote by $q(\vec{c}) = \{() \mid \psi(\vec{x}/\vec{c})\}$ the boolean *EQL* query in which the *EQL* sentence $\psi(\vec{x}/\vec{c})$ corresponds to $\perp$ in the case that there is some $i \in [1, n]$ for which $t_i \neq c_i$ and t_i is a constant, otherwise $\psi(\vec{x}/\vec{c})$ is obtained from $\psi(\vec{x})$ by replacing all the occurrences of the term t_i with the constant c_i, for each $i \in [1, n]$. Given an OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, an $\mathcal{S}$-database D, and an *EQL* query $q_{\mathcal{O}}$ over $\mathcal{O}$ of arity n, the certain answers of $q_{\mathcal{O}}$ with respect to Σ and D, denoted as always by $cert^D_{q_{\mathcal{O}},\Sigma}$, is the set $cert^D_{q_{\mathcal{O}},\Sigma} = \{\vec{c} \in \mathsf{dom}(D)^n \mid \langle \Sigma, D \rangle \models_{EQL} q_{\mathcal{O}}(\vec{c})\}$.

9.1 Towards *EQL-Lite*(UCQ) Abstractions

We start by recalling the *EQL-Lite*(UCQ) query language, and show how queries in such a language can be rewritten as FOL queries over the source schema to compute certain answers with respect to OBDM systems. We then show how the *EQL-Lite*(UCQ) query language allows to obtain better abstractions of data services, compared to the usual language of UCQs.

9.1.1 The *EQL-Lite*(UCQ) Query Language

The *EQL-Lite*(UCQ) query language, introduced in [Calvanese *et al.*, 2007a],[2] is an epistemic query language whose *epistemic atoms* are formulas of the form $\mathbf{K}\varrho$, where ϱ is a UCQ. Such a query language is a particularly well-behaved fragment of *EQL* queries. Formally, an *EQL-Lite*(UCQ) query over a DL ontology $\mathcal{O}$ is a query of the form $q_{\mathcal{O}} = \{\vec{t} \mid \psi(\vec{x})\}$, where the target list $\vec{t}$ is an n-tuple of terms, and the body $\psi(\vec{x})$ is a formula built according to the following syntax (we recall that the free variables occurring in formula $\psi(\vec{x})$ are exactly the variables occurring in $\vec{t}$):

$$\psi \ ::= \ \mathbf{K}\varrho \mid \exists x.\psi \mid \psi_1 \wedge \psi_2 \mid \psi_1 \vee \psi_2 \mid \neg\psi,$$

where ϱ is a disjunction of existentially quantified conjunction of atoms over $\mathcal{O}$ sharing the same free variables.

[2] In fact, we consider a restricted version of *EQL-Lite*(UCQ), in which (in)equalities are disallowed.

Example 9.1. Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the following OBDM specification:

- $\mathcal{O} = \{ \ \mathsf{Professor} \sqsubseteq \exists \mathsf{Teaches}, \mathsf{Teaches} \sqsubseteq \mathsf{Likes}, \exists \mathsf{WorksFor} \sqsubseteq \exists \mathsf{Likes} \ \}$
- $\mathcal{S} = \{ \ s_1, s_2, s_3, s_4 \ \}$
- $\mathcal{M} = \{ \ m_1, m_2, m_3, m_4 \ \}$, where:

$$\begin{array}{lrcl} m_1: & \exists y. s_1(x_1, x_2, y) & \to & \mathsf{Teaches}(x_1, x_2), \\ m_2: & s_2(x) & \to & \mathsf{Professor}(x), \\ m_3: & \exists y_1, y_2. s_3(x_1, x_2, y_1, y_2) & \to & \mathsf{Likes}(x_1, x_2), \\ m_4: & \exists y_1, y_2. s_4(x, y_1, y_2) & \to & \exists z. \mathsf{WorksFor}(x, z). \end{array}$$

Consider the following *EQL-Lite*(UCQ) queries over the ontology $\mathcal{O}$:

- $q^1_{\mathcal{O}} = \{(x) \mid \exists y. \mathbf{K}(\mathsf{Teaches}(x, y))\}$;
- $q^2_{\mathcal{O}} = \{(x) \mid \exists y. \mathbf{K}(\mathsf{Likes}(x, y))\}$;
- $q^3_{\mathcal{O}} = \{(x) \mid \exists y. \mathbf{K}(\mathsf{Likes}(x, y)) \land \neg \mathbf{K}(\mathsf{Teaches}(x, y))\}$.

Intuitively, $q^1_{\mathcal{O}}$ retrieves all the people for whom at least one subject is *known* they teach, $q^2_{\mathcal{O}}$ retrieves all the people for whom at least one subject is *known* they like, and finally $q^3_{\mathcal{O}}$ retrieves all the people for whom at least one subject is *known* they like and is *not known* they teach this subject. □

Each *EQL-Lite*(UCQ) query $q_{\mathcal{O}}$ over $\mathcal{O}$ can be associated with a FOL query over a new schema. Formally, let $q_{\mathcal{O}}$ be an *EQL-Lite*(UCQ) query over $\mathcal{O}$ whose epistemic atoms are $\mathbf{K}\varrho_1, \ldots, \mathbf{K}\varrho_m$. For each $i \in [1, m]$, we denote by $q^{\varrho_i}_{\mathcal{O}}$ the UCQ over $\mathcal{O}$ associated to ϱ_i, i.e., the UCQ $q^{\varrho_i}_{\mathcal{O}} = \{\vec{z} \mid \exists \vec{y_1}. \phi_1(\vec{z}, \vec{y_1})\} \ \cup \ldots \cup \ \{\vec{z} \mid \exists \vec{y_l}. \phi_l(\vec{z}, \vec{y_l})\}$, where $\vec{z} = (z_1, \ldots, z_{ar(\varrho_i)})$ is the tuple of all free variables occurring ϱ_i (we recall that each free variable $z_i \in \vec{z}$ of ϱ_i occurs also in the target list of $q_{\mathcal{O}}$), and $\varrho_i = \exists \vec{y_1}. \phi_1(\vec{z}, \vec{y_1}) \lor \ldots \lor \exists \vec{y_l}. \phi_l(\vec{z}, \vec{y_l})$. Finally, we denote by $q^{FOL}_{\mathcal{O}}$ the FOL query over the schema $\mathcal{R}_{q_{\mathcal{O}}} = \{R_{\mathbf{K}\varrho_1}, \ldots, R_{\mathbf{K}\varrho_m}\}$ obtained from $q_{\mathcal{O}}$ in the following way: for each $i \in [1, m]$, the epistemic atom $\mathbf{K}\varrho_i$ is replaced with the atom $R_{\mathbf{K}\varrho_i}(z_1, \ldots, z_{ar(\varrho_i)})$.

In order to reduce query answering of *EQL-Lite*(UCQ) queries to the standard evaluation of Relational Algebra queries, and therefore of SQL queries (thus taking advantage of optimisation strategies provided by DBMSs), we have to introduce the notion of "domain independence", which is the semantical restriction on FOL queries needed to get equivalence with Relational Algebra queries [Codd, 1972].

In the DL context, an FOL query $q_{\mathcal{O}}$ of arity n over an ontology $\mathcal{O}$ is *domain independent* if $\{\vec{c} \in \mathsf{Const}^n \mid \mathcal{I}_1 \models q_{\mathcal{O}}(\vec{c})\} = \{\vec{c} \in \mathsf{Const}^n \mid \mathcal{I}_2 \models q_{\mathcal{O}}(\vec{c})\}$ for each pair of FOL interpretations $\mathcal{I}_1 = \langle \Delta^{\mathcal{I}_1}, \cdot^{\mathcal{I}_1} \rangle$ and $\mathcal{I}_2 = \langle \Delta^{\mathcal{I}_2}, \cdot^{\mathcal{I}_2} \rangle$ for $\mathcal{O}$ such that $\cdot^{\mathcal{I}_1} \equiv \cdot^{\mathcal{I}_2}$, where $\cdot^{\mathcal{I}_1} \equiv \cdot^{\mathcal{I}_2}$ if $A^{\mathcal{I}_1} = A^{\mathcal{I}_2}$ (respectively, $P^{\mathcal{I}_1} = P^{\mathcal{I}_2}$) for each atomic concept A (respectively, atomic role P) in the alphabet of $\mathcal{O}$. We say that an *EQL-Lite*(UCQ) query $q_{\mathcal{O}}$ over an ontology $\mathcal{O}$ is *domain independent* if its associated $q^{FOL}_{\mathcal{O}}$ is so.

Several syntactic sufficient conditions have been devised to guarantee domain independence of FOL queries, see, e.g., [Abiteboul *et al.*, 1995]. Such syntactic conditions can be directly translated into syntactic conditions on *EQL-Lite*(UCQ) queries. As in relational database scenarios, where one allows only for FOL queries that are domain independent, in the sequel, whenever we speak about *EQL-Lite*(UCQ) queries, we will always mean domain independent *EQL-Lite*(UCQ) queries.

EQL-Lite(UCQ) queries enjoy a very interesting computational property: one can decouple the reasoning needed for answering the epistemic atoms, which can be delegated to the underlying OBDM service for answering UCQs, from the reasoning needed for dealing with the other operators of the whole query. Formally, let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification, D be an $\mathcal{S}$-database, and $q_{\mathcal{O}}$ be an *EQL-Lite*(UCQ) query over $\mathcal{O}$ whose epistemic atoms are $\mathbf{K}\varrho_1, \ldots, \mathbf{K}\varrho_m$. We denote by $\mathcal{I}^D_{q_{\mathcal{O}},\Sigma}$ the set of facts over the schema $\mathcal{R}_{q_{\mathcal{O}}} = \{R_{\mathbf{K}\varrho_1}, \ldots, R_{\mathbf{K}\varrho_m}\}$ obtained by including, for each $i \in [1, m]$ and for each $\vec{c} \in cert^D_{q^{\varrho_i}_{\mathcal{O}},\Sigma}$, the fact $R_{\mathbf{K}\varrho_i}(\vec{c})$. By using [Calvanese *et al.*, 2007a, Theorem 6], we easily get the following property.

Proposition 9.1. *Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification and $q_{\mathcal{O}}$ be an EQL-Lite(UCQ) query over $\mathcal{O}$ of arity n. Then, for each $\mathcal{S}$-database D consistent with Σ, we have that $cert^D_{q_{\mathcal{O}},\Sigma} = \{\vec{c} \in \mathsf{dom}(D) \mid \mathcal{I}^D_{q_{\mathcal{O}},\Sigma} \models q^{FOL}_{\mathcal{O}}(\vec{c})\}$.*

The above proposition tells us that, in order to compute the certain answers of an *EQL-Lite*(UCQ) query $q_{\mathcal{O}}$, we can compute the certain answers of the UCQs $q^{\varrho_i}_{\mathcal{O}}$ associated to the epistemic atoms $\mathbf{K}\varrho_i$ of $q_{\mathcal{O}}$, and then consider $q_{\mathcal{O}}$ as an FOL query, where such certain answers are regarded as the extensions of the epistemic atoms.

In the setting for OBDM specifications $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ considered in this thesis (i.e., *DL-Lite*$_{\mathcal{R}}$ as ontology language and the GLAV approach for the mapping language), the certain answers of the UCQs $q^{\varrho_i}_{\mathcal{O}}$ over $\mathcal{O}$ associated to the epistemic atoms $\mathbf{K}\varrho_i$ of an *EQL-Lite*(UCQ) $q_{\mathcal{O}}$ over $\mathcal{O}$ can be computed by means of a suitable query over the schema $\mathcal{S}$, which is $\mathsf{PerfRef}_{q^{\varrho_i}_{\mathcal{O}},\Sigma}$. Specifically, let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification and $q_{\mathcal{O}}$ be an *EQL-Lite*(UCQ) query over $\mathcal{O}$. We denote by $\mathsf{EQLPerfRef}_{q_{\mathcal{O}},\Sigma}$ the FOL query over the schema $\mathcal{S}$ obtained from $q_{\mathcal{O}}$ by replacing each of its epistemic atoms $\mathbf{K}\varrho_i$ with the logical *body* of the UCQ $\mathsf{PerfRef}_{q^{\varrho_i}_{\mathcal{O}},\Sigma}$.

Example 9.2. Refer to Example 9.1, and let $\varrho_1 = \mathsf{Teaches}(x, y)$ and $\varrho_2 = \mathsf{Likes}(x, y)$. Then, $q^{\varrho_1}_{\mathcal{O}} = \{(x, y) \mid \mathsf{Teaches}(x, y)\}$ and $q^{\varrho_2}_{\mathcal{O}} = \{(x, y) \mid \mathsf{Likes}(x, y)\}$. Therefore:

- $\mathsf{EQLPerfRef}_{q^1_{\mathcal{O}},\Sigma} = \{(x) \mid \exists y.\mathsf{PerfRef}_{q^{\varrho_1}_{\mathcal{O}},\Sigma}\} = \{(x) \mid \exists y.(\exists y'.s_1(x, y, y'))\}$;
- $\mathsf{EQLPerfRef}_{q^2_{\mathcal{O}},\Sigma} = \{(x) \mid \exists y.\mathsf{PerfRef}_{q^{\varrho_2}_{\mathcal{O}},\Sigma}\} = \{(x) \mid \exists y.(\exists y'_1, y'_2.s_3(x, y, y'_1, y'_2) \vee \exists y'.s_1(x, y, y'))\}$;
- $\mathsf{EQLPerfRef}_{q^3_{\mathcal{O}},\Sigma} = \{(x) \mid \exists y.(\mathsf{PerfRef}_{q^{\varrho_2}_{\mathcal{O}},\Sigma} \wedge \neg\mathsf{PerfRef}_{q^{\varrho_1}_{\mathcal{O}},\Sigma})\} = \{(x) \mid \exists y.((\exists y'_1, y'_2.s_3(x, y, y'_1, y'_2) \vee \exists y'.s_1(x, y, y')) \wedge \neg\exists y'.s_1(x, y, y'))\}$. $\square$

As a consequence of Proposition 9.1 and the above discussion, we can easily obtain the following theorem, which can be seen as the analogous of [Calvanese *et al.*, 2007a, Theorem 12] in the OBDM context.

Theorem 9.1. *Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification and $q_{\mathcal{O}}$ be an EQL-Lite(UCQ) query over $\mathcal{O}$. Then, for each $\mathcal{S}$-database D consistent with Σ, we have that $cert^D_{q_{\mathcal{O}},\Sigma} = \mathsf{EQLPerfRef}^D_{q_{\mathcal{O}},\Sigma}$.*

From the above theorem, we can derive the following observation for OBDM specifications $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ where $\mathcal{O}$ is a *DL-Lite*$_{\mathcal{R}}$ ontology and $\mathcal{M}$ is a GLAV mapping, which generalises the one given at the end of Chapter 2 for the sublanguage of UCQs: if $q_{\mathcal{O}}$ is an *EQL-Lite*(UCQ) query $q_{\mathcal{O}}$ over $\mathcal{O}$ of arity n, then the FOL query $\mathsf{EQLPerfRef}_{q_{\mathcal{O}},\Sigma} \vee \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},\Sigma}$ is the perfect $\mathcal{O}$-to-$\mathcal{S}$ Σ-rewriting of $q_{\mathcal{O}}$, i.e., $(\mathsf{EQLPerfRef}_{q_{\mathcal{O}},\Sigma} \vee \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},\Sigma})^D = cert^D_{q_{\mathcal{O}},\Sigma}$ for every $\mathcal{S}$-database D.

This allows us to generalise also Lemmata 5.1 and 6.1 and Corollary 7.1 when the class $\mathcal{L}_{\mathcal{O}}$ of queries over ontologies $\mathcal{O}$ is the one of *EQL-Lite*(UCQ) queries, and thus is more general than the one of UCQs considered in the mentioned results.

Corollary 9.1. *Let $n = ar(q_{\mathcal{O}}) = ar(q_{\mathcal{S}})$. We have that:*

- *$q_{\mathcal{O}}$ is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ if and only if the following subsumption between queries hold: $q_{\mathcal{S}} \sqsubseteq (\mathsf{EQLPerfRef}_{q_{\mathcal{O}},\Sigma} \vee \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},\Sigma})$;*
- *$q_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ if and only if the following subsumption between queries hold: $\mathsf{EQLPerfRef}_{q_{\mathcal{O}},\Sigma} \sqsubseteq (q_{\mathcal{S}} \vee \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},\Sigma})$;*
- *$q_{\mathcal{O}}$ is a perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ if and only if both the following subsumptions between queries hold: $q_{\mathcal{S}} \sqsubseteq (\mathsf{EQLPerfRef}_{q_{\mathcal{O}},\Sigma} \vee \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},\Sigma})$ and $\mathsf{EQLPerfRef}_{q_{\mathcal{O}},\Sigma} \sqsubseteq (q_{\mathcal{S}} \vee \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},\Sigma})$.*

9.1.2 *EQL-Lite*(UCQ) Source-to-Ontology Rewritings

As anticipated in the beginning of this section, we next show that considering *EQL-Lite*(UCQ) queries as target query language provides more expressivity in finding source-to-ontology rewritings, compared to UCQs. In particular, the next example shows that there are cases where no perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting exists in the class of UCQ$^{\neq}$s, whereas it exists in the class of *EQL-Lite*(UCQ) queries.

Example 9.3. Consider the OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ illustrated in Example 9.1, and let the data service be expressed as the CQJFE $q_{\mathcal{S}} = \{(x) \mid \exists y, y'. s_1(x, y, y')\}$ over $\mathcal{S}$. From results presented in Chapter 5, one can see that the query $q^c_{\mathcal{O}} = \{(x) \mid \exists y. \mathsf{Teaches}(x, y)\}$ is the UCQ$^{\neq}$-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. Notice, however, that due to the presence of the ontology assertion $\mathsf{Professor} \sqsubseteq \exists \mathsf{Teaches}$ and of the mapping assertion m_2, the certain answers of $q^c_{\mathcal{O}}$ with respect to Σ and a given $\mathcal{S}$-database D include the values stored both in the first component of s_1 and in s_2. Thus, $q^c_{\mathcal{O}}$ is not a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, which allows us to conclude that no UCQ$^{\neq}$-perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ exists.

On the other hand, consider the *EQL-Lite*(UCQ) query $q^1_{\mathcal{O}}$ illustrated in Example 9.1. One can verify that $q^1_{\mathcal{O}}$ is a perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ (indeed, observe that $\mathsf{EQLPerfRef}_{q^1_{\mathcal{O}},\Sigma} \equiv q_{\mathcal{S}}$, where $\mathsf{EQLPerfRef}_{q^1_{\mathcal{O}},\Sigma}$ is the FOL query illustrated in Example 9.2 corresponding to the perfect $\mathcal{O}$-to-$\mathcal{S}$ Σ-rewriting of $q^1_{\mathcal{O}}$). $\square$

Unfortunately, exactly as in the case of UCQ$^{\neq}$s as target query language, the next example shows that there are pairs of OBDM specifications $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and CQJFEs $q_{\mathcal{S}}$ over $\mathcal{S}$ for which no perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ exists in the class of *EQL-Lite*(UCQ) queries.

Example 9.4. Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be again the OBDM specification illustrated in Example 9.1, and let the data service be expressed as the CQJFE $q_{\mathcal{S}} = \{(x) \mid \exists y, y', y''.s_3(x, y, y', y'')\}$ over $\mathcal{S}$. Consider the *EQL-Lite*(UCQ) queries $q^2_{\mathcal{O}}$ and $q^3_{\mathcal{O}}$ illustrated in Example 9.1. One can see that, because of the presence of the ontology assertion $\mathsf{Teaches} \sqsubseteq \mathsf{Likes}$ and of the mapping assertion m_1, the query $q^2_{\mathcal{O}}$ is too general to be a perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting. Indeed, its certain answers with respect to Σ and a given $\mathcal{S}$-database D include the values stored in the first component of both s_1 and s_3. On the other hand, the query $q^3_{\mathcal{O}}$ is too specific to be a perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. Indeed, there are $\mathcal{S}$-databases D for which its certain answers with respect to Σ and D do not include the values stored in the first component of s_3. The above observations allow us to conclude that no perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ exists in the class of *EQL-Lite*(UCQ) queries. $\square$

Clearly, as always, when perfect source-to-ontology rewritings in the class of *EQL-Lite*(UCQ) queries do not exist, the goal is to find queries in such a class that provide the best approximations of $q_{\mathcal{S}}$. By developing on the above example, we show that the class of *EQL-Lite*(UCQ) queries allows to find better approximations of source-to-ontology rewritings compared to the UCQ query language.

Example 9.5. Refer to Example 9.4, where $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ is the OBDM specification illustrated in Example 9.1 and the data service is expressed as the CQJFE $q_{\mathcal{S}} = \{(x) \mid \exists y, y', y''.s_3(x, y, y', y'')\}$ over $\mathcal{S}$. From results of previous chapters, one can verify that the queries $q^c_{\mathcal{O}} = \{(x) \mid \exists y.\mathsf{Likes}(x, y)\}$ and $q^s_{\mathcal{O}} = \{(x) \mid \bot(x)\}$ are, respectively, the (unique up to equivalence w.r.t. Σ) UCQ$^{\neq}$-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ and the (unique up to equivalence w.r.t. Σ) UCQ$^{\neq}$-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.

As for the best approximations of $q_{\mathcal{S}}$ (w.r.t. Σ) in the class of *EQL-Lite*(UCQ) queries, one can verify that the *EQL-Lite*(UCQ) queries $q^2_{\mathcal{O}}$ and $q^3_{\mathcal{O}}$ illustrated in Example 9.1 are, respectively, the (unique up to equivalence w.r.t. Σ) minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of *EQL-Lite*(UCQ) queries and the unique (up to equivalence w.r.t. Σ) maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of *EQL-Lite*(UCQ) queries. It is also clear that $q^2_{\mathcal{O}}$ and $q^3_{\mathcal{O}}$ are better approximations of $q_{\mathcal{S}}$ (w.r.t. Σ) compared to $q^c_{\mathcal{O}}$ and $q^s_{\mathcal{O}}$, respectively. $\square$

In the next sections, we carry out a study on the problem of computing the best abstractions of data services, expressed as queries over the source schema, in the class of *EQL-Lite*(UCQ) queries and also in a fragment of it.

Interestingly, for the class of *EQL-Lite*(UCQ) queries we have the following result, which is the analogous of Proposition 3.2 for the *EQL-Lite*(UCQ) query language, and can be easily proven by following exactly the same line of arguments used in the proof of that proposition.

Proposition 9.2. *If q_1 and q_2 are minimally complete (respectively, maximally sound) $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewritings of $q_{\mathcal{S}}$ in the class of EQL-Lite(UCQ) queries, then they are equivalent w.r.t. Σ.*

9.2 On the non-existence of *EQL-Lite*(UCQ) Source-to-Ontology Rewritings

Surprisingly, we now prove that neither minimally complete, nor maximally sound source-to-ontology rewritings in the class of *EQL-Lite*(UCQ) queries are guaranteed to exist, even in the case of OBDM specifications $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and data services $q_{\mathcal{S}}$ where $\mathcal{O} = \emptyset$, $\mathcal{M}$ is a pure GAV mapping, and $q_{\mathcal{S}}$ is a CQJFE over $\mathcal{S}$.

We start by looking at the minimally complete case.

Theorem 9.2. *There exists an OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ with $\mathcal{O} = \emptyset$ and $\mathcal{M}$ being a pure GAV mapping and a CQJFE $q_{\mathcal{S}}$ for which no minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of* EQL-Lite(*UCQ*) *queries exists.*

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

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

$$
\begin{array}{lrcl}
m_1: & s(x) & \rightarrow & B(x),\\
m_2: & s_1(x) \wedge s_2(x) & \rightarrow & B(x),\\
m_3: & s_1(x) & \rightarrow & A_1(x),\\
m_4: & s_2(x_1) \wedge s_3(x_1, x_2) & \rightarrow & P(x_1, x_2),\\
m_5: & s_1(x_2) \wedge s_5(x_1, x_2) & \rightarrow & P(x_1, x_2),\\
m_6: & s_2(x) \wedge s_4(x) & \rightarrow & A_2(x),\\
m_7: & s(x_1) \wedge s_1(x_2) \wedge s_2(x_2) & \rightarrow & R(x_1, x_2).
\end{array}
$$

Let $q_{\mathcal{S}}$ be the following boolean CQJFE over $\mathcal{S}$: $q_{\mathcal{S}} = \{() \mid \exists y.s(y)\}$.

From results of Chapter 5, we know that the UCQ-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ is the query $q^c_{\mathcal{O}} = \{() \mid \exists y.B(y)\}$. Observe that, however, $q^c_{\mathcal{O}}$ is not a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. This is because there are $\mathcal{S}$-databases D for which (i) there are no facts of the form $s(c)$ (i.e., the extension of source predicate s in D is empty) and (ii) D contains both the facts $s_1(c)$ and $s_2(c)$ for a certain constant $c \in \mathsf{Const}$, thus resulting in $q^D_{\mathcal{S}} = \emptyset$ whereas $cert^D_{q_{\mathcal{O}},\Sigma} = \{\}$.

With the ability of the *EQL-Lite*(UCQ) query language of expressing epistemic forms of negations, we can detect some of those cases by adding $\neg\mathbf{K}(\varrho)$ in conjunction to the epistemic atom $\mathbf{K}(\exists y.B(y))$, where ϱ is the body of a CQ being a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $\{() \mid \exists y.s_1(y) \wedge s_2(y)\}$. At the same time, however, we need to ensure the fact that such new *EQL-Lite*(UCQ) query that we are building is still a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. In other words, we want that whenever an $\mathcal{S}$-database D contains a fact $s(c)$ for a constant $c \in \mathsf{Const}$, the certain answers of the new query are not empty, even if $\mathbf{K}(\varrho)$ is true in the current OBDM system. By looking at the mapping $\mathcal{M}$, we can ensure this by simply adding the epistemic atom $\mathbf{K}(\exists y, y'.R(y, y'))$ in disjunction to the new *EQL-Lite*(UCQ) query.

With the above discussion at hand, we now introduce a pattern for an infinite number of *EQL-Lite*(UCQ) queries over $\mathcal{O}$ and related technical lemmata. Specifically, for every $i \geq 0$, let $q^i_{\mathcal{O}}$ be the following *EQL-Lite*(UCQ) query over $\mathcal{O}$:

$$q^i_{\mathcal{O}} = \{() \mid \big(\mathbf{K}(\exists y.B(y)) \wedge (\bigwedge_{k=0}^{k=i} \neg\mathbf{K}(\varrho_k))\big) \vee \mathbf{K}(\exists y, y'.R(y, y'))\}, \text{where}$$

- $\varrho_0 = \exists y_0.A_1(y_0) \wedge A_2(y_0)$.
- $\varrho_k = \exists y_0, \ldots, y_k.A_1(y_0) \wedge (\bigwedge_{j=0}^{j=k-1} P(y_j, y_{j+1})) \wedge A_2(y_k)$, for each $k \in [1, i]$.

For instance, with $i = 2$, we have $q^2_{\mathcal{O}} = \{() \mid \big(\mathbf{K}(\exists y.B(y)) \wedge \neg\mathbf{K}(\exists y_0.A_1(y_0) \wedge A_2(y_0)) \wedge \neg\mathbf{K}(\exists y_0, y_1.A_1(y_0) \wedge P(y_0, y_1) \wedge A_2(y_1)) \wedge \neg\mathbf{K}(\exists y_0, y_1, y_2.A_1(y_0) \wedge P(y_0, y_1) \wedge P(y_1, y_2) \wedge A_2(y_2))\big) \vee \mathbf{K}(\exists y, y'.R(y, y'))\}$.

Observe that, for each $k \in [0, i]$, the body ϱ_k in the epistemic atom $\mathbf{K}(\varrho_k)$ is exactly the body of the CQ $q^k_{\mathcal{O}} = \{() \mid \varrho_k\}$ illustrated in the proof of point 1 of Theorem 6.2, which we recall that is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of the query $\{() \mid \exists y.s_1(y) \wedge s_2(y)\}$ (note that the mapping assertions m_3, m_4, m_5, and m_6 are identical to the mapping assertions m_1, m_2, m_3, and m_4, respectively, of the mapping illustrated in the proof of point 1 of Theorem 6.2).

Lemma 9.1. *For every natural number $i \geq 0$, we have that $q^i_{\mathcal{O}}$ is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.*

Proof. Consider any $i \geq 0$ and the associated *EQL-Lite*(UCQ) query $q^i_{\mathcal{O}}$. Let D be any $\mathcal{S}$-database D for which $q^D_{\mathcal{S}} = \{\}$, which is equivalent to say that $s(c) \in D$ for some constant $c \in \mathsf{Const}$. There are two possible cases: either for all $k \in [0, i]$ the epistemic atom $\mathbf{K}(\varrho_k)$ is false in the OBDM system $\langle \Sigma, D \rangle$, or not.

In the former case, due to the mapping assertion m_1 and the fact that $s(c) \in D$, we trivially have that the epistemic atom $\mathbf{K}(\exists y.B(y))$ is true in $\langle \Sigma, D \rangle$. It follows that the formula $\big(\mathbf{K}(\exists y.B(y)) \wedge \neg\mathbf{K}(\varrho_0) \wedge \ldots \wedge \neg\mathbf{K}(\varrho_i)\big)$ is as well true in $\langle \Sigma, D \rangle$, thus implying that $cert^D_{q^i_{\mathcal{O}}, \Sigma} = \{\}$.

In the latter case, there is some $k \in [0, i]$ for which $\mathbf{K}(\varrho_k)$ is true in $\langle \Sigma, D \rangle$. Notice that, since ϱ_k is the body of a CQ that is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $\{() \mid \exists y.s_1(y) \wedge s_2(y)\}$ (cf. Lemma 6.2), we derive that D contains both the facts $s_1(c')$ and $s_2(c')$ for some constant $c' \in \mathsf{Const}$. This, together with the fact that $s(c) \in D$ for some constant $c \in \mathsf{Const}$, and by looking at the m_7 mapping assertion of $\mathcal{M}$, allows us to easily derive that the epistemic atom $\mathbf{K}(\exists y, y'.R(y, y'))$ is true in $\langle \Sigma, D \rangle$, thus implying that $cert^D_{q^i_{\mathcal{O}}, \Sigma} = \{\}$ also in this case.

Thus, for any $\mathcal{S}$-database D, if $q^D_{\mathcal{S}} = \{\}$, then $cert^D_{q^i_{\mathcal{O}}, \Sigma} = \{\}$. This clearly implies that $q^i_{\mathcal{O}}$ is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, as required. $\square$

Lemma 9.2. *For every natural number $i \geq 0$, we have that $cert_{q^{i+1}_{\mathcal{O}}, \Sigma} \sqsubset cert_{q^i_{\mathcal{O}}, \Sigma}$.*

Proof. The proof immediately follows from the proof of Lemma 6.3, which implies that, for any pair of natural numbers $l, m \geq 0$ with $l \neq m$, the CQs $q^l_{\mathcal{O}} = \{() \mid \varrho_l\}$ and $q^m_{\mathcal{O}} = \{() \mid \varrho_m\}$ illustrated in the proof of point 1 of Theorem 6.2 are such that both $cert_{q^l_{\mathcal{O}}, \Sigma} \not\sqsubseteq cert_{q^m_{\mathcal{O}}, \Sigma}$ and $cert_{q^m_{\mathcal{O}}, \Sigma} \not\sqsubseteq cert_{q^l_{\mathcal{O}}, \Sigma}$ hold. Consider any natural number $i \geq 0$ and the *EQL-Lite*(UCQ) queries $q^i_{\mathcal{O}}$ and $q^{i+1}_{\mathcal{O}}$. Since the body of $q^{i+1}_{\mathcal{O}}$ contains also $\neg\mathbf{K}(\varrho_{i+1})$ whereas $q^i_{\mathcal{O}}$ does not, from the above observation, we trivially derive that $cert_{q_{\mathcal{O}}^{i+1},\Sigma} \sqsubset cert_{q_{\mathcal{O}}^{i},\Sigma}$. Indeed, there is at least an $\mathcal{S}$-database D (which contains both the facts $s_1(c)$ and $s_2(c)$ for some constant $c \in \mathsf{Const}$) in which the epistemic atom $\mathbf{K}(\varrho_{i+1})$ is true in $\langle\Sigma, D\rangle$ whereas $\mathbf{K}(\varrho_j)$ is false in $\langle\Sigma, D\rangle$ for any $j \in [0, i]$, thus implying that $cert^{D}_{q_{\mathcal{O}}^{i+1},\Sigma} = \emptyset$ whereas $cert^{D}_{q_{\mathcal{O}}^{i},\Sigma} = \{\}$, as required. □

Furthermore, due to Lemma 6.4, we know that each possible sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $\{() \mid \exists y. s_1(y) \wedge s_2(y)\}$ is equivalent to the query $\{() \mid \varrho_i\}$, for some $i \geq 0$. Thus, using similar arguments as the ones given in the proof of Lemma 6.4, it is not difficult to see that each possible *EQL-Lite*(UCQ) query $q'_{\mathcal{O}}$ that is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ is such that there exists an $i \geq 0$ for which $cert_{q_{\mathcal{O}}^{i},\Sigma} \sqsubset cert_{q'_{\mathcal{O}},\Sigma}$ (including $q^{c}_{\mathcal{O}} = \{() \mid \exists y. B(y)\}$, i.e., the UCQ-minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, which is clearly such that $cert_{q_{\mathcal{O}}^{i},\Sigma} \sqsubset cert_{q'_{\mathcal{O}},\Sigma}$ for any $i \geq 0$).

With this observation and the above lemmata at hand, we are now able to prove that no minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of *EQL-Lite*(UCQ) queries exists. Indeed, since each possible *EQL-Lite*(UCQ) query $q'_{\mathcal{O}}$ that is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ is such that there exists an $i \geq 0$ for which $cert_{q_{\mathcal{O}}^{i},\Sigma} \sqsubset cert_{q'_{\mathcal{O}},\Sigma}$, and since by Lemma 9.1 $q^{i}_{\mathcal{O}}$ is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ for any $i \geq 0$, when seeking for a minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of *EQL-Lite*(UCQ) queries one can limit the attention to only the illustrated queries $q^{i}_{\mathcal{O}}$ for $i \geq 0$. Notice that, however, by Lemma 9.2, for any natural number $i \geq 0$, the query $q^{i+1}_{\mathcal{O}}$ is a better complete approximation of the $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ compared to the query $q^{i}_{\mathcal{O}}$ (i.e., $cert_{q_{\mathcal{O}}^{i+1},\Sigma} \sqsubset cert_{q_{\mathcal{O}}^{i},\Sigma}$). Thus, since in any *EQL-Lite*(UCQ) query there are only a *finite* set of epistemic atoms, we can conclude that no minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of *EQL-Lite*(UCQ) queries exists, as required. □

We now turn to the maximally sound case.

Theorem 9.3. *There exists an OBDM specification $\Sigma = \langle\mathcal{O}, \mathcal{S}, \mathcal{M}\rangle$ with $\mathcal{O} = \emptyset$ and $\mathcal{M}$ being a pure GAV mapping and a CQJFE $q_{\mathcal{S}}$ for which no maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of* EQL-Lite(*UCQ*) *queries exists.*

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

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

$$
\begin{array}{lrcl}
m_1: & s(x) & \rightarrow & B(x), \\
m_2: & s'(x) & \rightarrow & B(x), \\
m_3: & s_1(x) & \rightarrow & A_1(x), \\
m_4: & s_2(x_1) \wedge s_3(x_1, x_2) & \rightarrow & P(x_1, x_2), \\
m_5: & s_1(x_2) \wedge s_5(x_1, x_2) & \rightarrow & P(x_1, x_2), \\
m_6: & s_2(x) \wedge s_4(x) & \rightarrow & A_2(x), \\
m_7: & s'(x_1) \wedge s_1(x_2) \wedge s_2(x_2) & \rightarrow & R(x_1, x_2).
\end{array}
$$

Let $q_{\mathcal{S}}$ be the following boolean CQJFE over $\mathcal{S}$: $q_{\mathcal{S}} = \{() \mid \exists y. s(y)\}$.

From results of Chapter 8, we know that the UCQ-maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ is the CQ $q^s_{\mathcal{O}} = \{() \mid \bot\}$. Informally, this is because the CQ $q^c_{\mathcal{O}} = \{() \mid \exists y. B(y)\}$ is not a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, since the query $q'_{\mathcal{S}} = \{() \mid \exists y. s'(y)\}$ is a disjunct of $\mathsf{PerfRef}_{q^c_{\mathcal{O}},\Sigma}$ such that $q'_{\mathcal{S}} \not\sqsubseteq q_{\mathcal{S}}$ (cf. Lemma 6.1).

With the ability of the *EQL-Lite*(UCQ) query language of expressing epistemic forms of negations, we can detect some of the cases where the epistemic atom $\mathbf{K}(\exists y. B(y))$ is true in $\langle \Sigma, D\rangle$ for a certain $\mathcal{S}$-database D and, nevertheless, it is sure that there are no facts of the form $s'(c)$ in D (i.e., the extension of source predicate s' in D is empty). Observe that, by looking at the mapping assertions in $\mathcal{M}$ (in particular, to the m_7 mapping assertion), this latter requirement can be achieved by the following *EQL-Lite*(UCQ) formula: $\neg\mathbf{K}(\exists y, y'. R(y, y')) \wedge \mathbf{K}(\varrho)$, where ϱ is the body of a UCQ being a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of the CQ $\{() \mid \exists y. s_1(y) \wedge s_2(y)\}$.

With the above discussion at hand, we now introduce a pattern for an infinite number of *EQL-Lite*(UCQ) queries over $\mathcal{O}$ and related technical lemmata. Specifically, for every $i \geq 0$, let $q^i_{\mathcal{O}}$ be the following *EQL-Lite*(UCQ) query over $\mathcal{O}$:

$$q^i_{\mathcal{O}} = \{() \mid \mathbf{K}(\exists y. B(y)) \wedge \neg\mathbf{K}(\exists y, y'. R(y, y')) \wedge \mathbf{K}(\bigcup_{k=0}^{k=i} \varrho_k)\}, \text{where}$$

- $\varrho_0 = \exists y_0. A_1(y_0) \wedge A_2(y_0)$.
- $\varrho_k = \exists y_0, \ldots, y_k. A_1(y_0) \wedge (\bigwedge_{j=0}^{j=k-1} P(y_j, y_{j+1})) \wedge A_2(y_k)$, for each $k \in [1, i]$.

For instance, with $i = 2$, we have $q^2_{\mathcal{O}} = \{() \mid \mathbf{K}(\exists y. B(y)) \wedge \neg\mathbf{K}(\exists y, y'. R(y, y')) \wedge \mathbf{K}((\exists y_0. A_1(y_0) \wedge A_2(y_0)) \cup (\exists y_0, y_1. A_1(y_0) \wedge P(y_0, y_1) \wedge A_2(y_1)) \cup (\exists y_0, y_1, y_2. A_1(y_0) \wedge P(y_0, y_1) \wedge P(y_1, y_2) \wedge A_2(y_2)))\}$.

Observe that, for each $k \in [0, i]$, the disjunct ϱ_k in the last epistemic atom is exactly the body of the CQ $q^k_{\mathcal{O}} = \{() \mid \varrho_k\}$ illustrated in the proof of point 1 of Theorem 6.2, which we recall that is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of the query $\{() \mid \exists y. s_1(y) \wedge s_2(y)\}$ (note that the mapping assertions m_3, m_4, m_5, and m_6 are identical to the mapping assertions m_1, m_2, m_3, and m_4, respectively, of the mapping illustrated in the proof of point 1 of Theorem 6.2).

Lemma 9.3. *For every natural number $i \geq 0$, we have that $q^i_{\mathcal{O}}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$.*

Proof. Consider any $i \geq 0$ and the associated *EQL-Lite*(UCQ) query $q^i_{\mathcal{O}}$. Let D be any $\mathcal{S}$-database for which $cert^D_{q^i_{\mathcal{O}},\Sigma} = \{\}$. Since $cert^D_{q^i_{\mathcal{O}},\Sigma} = \{\}$, we derive the following implications: (i) the epistemic atom $\mathbf{K}(\bigcup_{k=0}^{k=i} \varrho_k)$ of $q^i_{\mathcal{O}}$ is true in $\langle \Sigma, D\rangle$; (ii) the epistemic atom $\mathbf{K}(\exists y, y'. R(y, y'))$ of $q^i_{\mathcal{O}}$ is false in $\langle \Sigma, D\rangle$; and (iii) the epistemic atom $\mathbf{K}(\exists y. B(y))$ of $q^i_{\mathcal{O}}$ is true in $\langle \Sigma, D\rangle$.

From (i), since $\bigcup_{k=0}^{k=i} \varrho_k$ is the body of a UCQ that is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $\{() \mid \exists y. s_1(y) \wedge s_2(y)\}$ (indeed, by Lemma 6.2, for each $k \in [0, i]$, ϱ_k is the body of a CQ that is a $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $\{() \mid \exists y. s_1(y) \wedge s_2(y)\}$), we derive that the $\mathcal{S}$-database D contains both the facts $s_1(c)$ and $s_2(c)$ for some constant $c \in \mathsf{Const}$. This, together with (ii), and by looking at the m_7 mapping assertion of $\mathcal{M}$, allows us to easily derive that there are no facts of the form $s'(c')$ in D (i.e., the extension of the source predicate s' in D is empty). But then, since (iii) holds, by looking at the mapping $\mathcal{M}$, we derive that the $\mathcal{S}$-database D necessarily contains a fact of the form $s(c'')$ for some constant $c'' \in \mathsf{Const}$, thus implying that $q_{\mathcal{S}}^{D} = \{\}$.

Thus, for any $\mathcal{S}$-database D, if $cert_{q_{\mathcal{O}}^{i},\Sigma}^{D} = \{\}$, then $q_{\mathcal{S}}^{D} = \{\}$. This clearly implies that $q_{\mathcal{O}}^{i}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, as required. $\square$

Lemma 9.4. *For every natural number $i \geq 0$, we have that $cert_{q_{\mathcal{O}}^{i},\Sigma} \sqsubset cert_{q_{\mathcal{O}}^{i+1},\Sigma}$.*

Proof. The proof immediately follows from the proof of Lemma 6.3, which implies that, for any pair of natural numbers $l, m \geq 0$ with $l \neq m$, the CQs $q_{\mathcal{O}}^{l} = \{() \mid \varrho_l\}$ and $q_{\mathcal{O}}^{m} = \{() \mid \varrho_m\}$ illustrated in the proof of point 1 of Theorem 6.2 are such that both $cert_{q_{\mathcal{O}}^{l},\Sigma} \not\sqsubseteq cert_{q_{\mathcal{O}}^{m},\Sigma}$ and $cert_{q_{\mathcal{O}}^{m},\Sigma} \not\sqsubseteq cert_{q_{\mathcal{O}}^{l},\Sigma}$ hold. Consider any natural number $i \geq 0$ and the *EQL-Lite*(UCQ) queries $q_{\mathcal{O}}^{i}$ and $q_{\mathcal{O}}^{i+1}$. Since the disjunct ϱ_{i+1} occurs in the epistemic atom $\mathbf{K}\big(\bigcup_{k=0}^{k=i+1} \varrho_k\big)$ whereas it does not occur in the epistemic atom $\mathbf{K}\big(\bigcup_{k=0}^{k=i} \varrho_k\big)$, from the above observation, we trivially derive that $cert_{q_{\mathcal{O}}^{i},\Sigma} \sqsubset cert_{q_{\mathcal{O}}^{i+1},\Sigma}$. Indeed, there is at least an $\mathcal{S}$-database D (which contains both the facts $s_1(c)$ and $s_2(c)$ for some constant $c \in \mathsf{Const}$) in which the formula ϱ_{i+1} is true in $\langle \Sigma, D\rangle$ (and therefore, the epistemic atom $\mathbf{K}\big(\bigcup_{k=0}^{k=i+1} \varrho_k\big)$ is true in $\langle \Sigma, D\rangle$) whereas formula ϱ_j is false in $\langle \Sigma, D\rangle$ for any $j \in [0, i]$ (and therefore, the epistemic atom $\mathbf{K}\big(\bigcup_{k=0}^{k=i} \varrho_k\big)$ is false in $\langle \Sigma, D\rangle$), thus implying that $cert_{q_{\mathcal{O}}^{i+1},\Sigma}^{D} = \{\}$ whereas $cert_{q_{\mathcal{O}}^{i},\Sigma}^{D} = \emptyset$, as required. $\square$

Furthermore, due to Lemma 6.4, we know that each possible sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $\{() \mid \exists y . s_1(y) \land s_2(y)\}$ is equivalent to the query $\{() \mid \varrho_i\}$, for some $i \geq 0$. Thus, using similar arguments as the ones given in the proof of Lemma 6.4, it is not difficult to see that each possible *EQL-Lite*(UCQ) query $q'_{\mathcal{O}}$ that is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ is such that there exists an $i \geq 0$ for which $cert_{q'_{\mathcal{O}},\Sigma} \sqsubset cert_{q_{\mathcal{O}}^{i},\Sigma}$.

With this observation and the above lemmata at hand, we are now able to prove that no maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of *EQL-Lite*(UCQ) queries exists. Indeed, since each possible *EQL-Lite*(UCQ) query $q'_{\mathcal{O}}$ that is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ is such that there exists an $i \geq 0$ for which $cert_{q'_{\mathcal{O}},\Sigma} \sqsubset cert_{q_{\mathcal{O}}^{i},\Sigma}$, and since by Lemma 9.3 $q_{\mathcal{O}}^{i}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ for any $i \geq 0$, when seeking for a maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of *EQL-Lite*(UCQ) queries one can limit the attention to only the illustrated queries $q_{\mathcal{O}}^{i}$ for $i \geq 0$. Notice that, however, by Lemma 9.4, for any natural number $i \geq 0$, the query $q_{\mathcal{O}}^{i+1}$ is a better sound approximation of the $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ compared to the query $q_{\mathcal{O}}^{i}$ (i.e., $cert_{q_{\mathcal{O}}^{i},\Sigma} \sqsubset cert_{q_{\mathcal{O}}^{i+1},\Sigma}$). Thus, since in any *EQL-Lite*(UCQ) query there are only a *finite* set of epistemic atoms, and since each of its epistemic atoms $\mathbf{K}(\varrho)$ must be such that ϱ is the body of a UCQ (and therefore, in ϱ there are only a *finite* set of bodies of CQs), we can conclude that no maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of *EQL-Lite*(UCQ) queries exists, as required. $\square$

In the light of the above inexpressibility results, we explore two alternative special scenarios. In what follows, we will limit our attention to the DL ontology language *DL-Lite*$_{\mathcal{R}}^{-}$, which is the fragment of *DL-Lite*$_{\mathcal{R}}$ where disjointness assertions are disallowed. Observe that, for each OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M}\rangle$ where $\mathcal{O}$ is a *DL-Lite*$_{\mathcal{R}}^{-}$ ontology and $\mathcal{M}$ is a GLAV mapping, since inconsistencies can not arise, every $\mathcal{S}$-database D is consistent with Σ. Thus, all the results we will present in the following hold even according to the semantics proposed in [Lutz *et al.*, 2018].

More specifically, in Section 9.3 we weaken the target query language by considering a fragment of the *EQL-Lite*(UCQ) query language that, notwithstanding, still enjoys a non-monotonic feature. In Section 9.4, instead, we weaken the mapping language by considering a special case of the GLAV approach that is incomparable with both the GAV, and the LAV approach.

9.3 Source-to-Ontology Rewritings in a fragment of *EQL-Lite*(UCQ)

In this section, we explore the possibility of expressing source-to-ontology rewritings where the target query language is a fragment, still non-monotonic, of the *EQL-Lite*(UCQ) query language considered so far.

In particular, while the proof of Theorem 9.3 shows that epistemic negation (even when not nested) already suffices to prevent the existence of maximally sound source-to-ontology rewritings in the class of *EQL-Lite*(UCQ) queries (even with empty ontologies and pure GAV mappings), the proof of Theorem 9.2 suggests to remove the union (i.e., the rule $\psi ::= \psi_1 \vee \psi_2$) from the syntax of the *EQL-Lite*(UCQ) query language, in order to get a target query language $\mathcal{L}_\mathcal{O}$ ensuring the existence of minimally complete source-to-ontology rewritings in $\mathcal{L}_\mathcal{O}$. Thus, based on the observation that union can be expressed by means of conjunction and nested negation, we next consider the fragment of the *EQL-Lite*(UCQ) query language where both nested negation and union operator are disallowed.

Formally, an *EQL-Lite*$^-$(UCQ) query over a DL ontology $\mathcal{O}$ is a query of the form $q_\mathcal{O} = \{\vec{t} \mid \psi(\vec{x})\}$, where the target list $\vec{t}$ is an n-tuple of terms, and the body $\psi(\vec{x})$ is a formula built according to the following syntax (we recall that the free variables occurring in formula $\psi(\vec{x})$ are exactly the variables occurring in $\vec{t}$):

$$\begin{array}{lcl} \psi & ::= & \mathbf{K}\varrho \;\mid\; \exists y.\psi \;\mid\; \psi_1 \wedge \psi_2 \;\mid\; \neg\delta \\ \delta & ::= & \mathbf{K}\varrho \;\mid\; \exists y.\delta \end{array}$$

The following example illustrates the *EQL-Lite*$^-$(UCQ) query language.

Example 9.6. The queries $q^i_\mathcal{O}$, for $i \geq 0$, used in the proof of Theorem 9.3, as well as the queries $q^1_\mathcal{O}$, $q^2_\mathcal{O}$, and $q^3_\mathcal{O}$ illustrated in Example 9.1, are *EQL-Lite*$^-$(UCQ) queries. On the contrary, for any $i \geq 0$, the query $q^i_\mathcal{O} = \{() \mid \mathbf{K}(\varrho) \wedge \neg\mathbf{K}(\varrho_0) \wedge \neg\mathbf{K}(\varrho_1) \wedge \ldots \wedge \neg\mathbf{K}(\varrho_i) \vee \mathbf{K}(\varrho')\}$ introduced in the proof of Theorem 9.2 is an *EQL-Lite*(UCQ) query but not an *EQL-Lite*$^-$(UCQ) query. $\square$

Observe that, since all the queries over the ontology involved in the proof of Theorem 9.3 are *EQL-Lite*$^-$(UCQ) queries, such proof in fact shows a stronger result than the one stated in the theorem: maximally sound source-to-ontology rewritings are not guaranteed to exist even in the class of *EQL-Lite*$^-$(UCQ) queries (and even for OBDM specifications $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and data services $q_\mathcal{S}$ where $\mathcal{O} = \emptyset$, $\mathcal{M}$ is a pure GAV mapping, and $q_\mathcal{S}$ is a CQJFE over $\mathcal{S}$).

It thus remains to study the minimally complete case. Specifically, we now provide the algorithm MinimallyCompleteEpistemic for computing minimally complete source-to-ontology rewritings of CQs over the source schema when the target query language is the class of *EQL-Lite*$^-$(UCQ) queries. Thus proving that, for each pair composed by an OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ where $\mathcal{O}$ is a *DL-Lite*$^-_{\mathcal{R}}$ ontology and $\mathcal{M}$ is a GLAV mapping and a CQ $q_{\mathcal{S}}$ over $\mathcal{S}$, the unique (up to equivalence w.r.t. Σ) minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of *EQL-Lite*$^-$(UCQ) queries is guaranteed to exists.

Algorithm 9.1 MinimallyCompleteEpistemic

Input:

OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ where $\mathcal{O}$ is a *DL-Lite*$^-_{\mathcal{R}}$ ontology and $\mathcal{M}$ is a GLAV mapping;

CQ $q_{\mathcal{S}} = \{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$ over $\mathcal{S}$

Output:

EQL-Lite$^-$(UCQ) query $q_{\mathcal{O}}$ over $\mathcal{O}$

1: $q_{\mathcal{O}} := \{\vec{t} \mid \exists \vec{y}'.\mathbf{K}(\exists \vec{z}.\mathcal{M}(q_{\mathcal{S}}) \wedge \top(\vec{x}))\}$, where $\vec{y}' \subseteq \vec{y}$ denotes the subset of existential variables of $q_{\mathcal{S}}$ occurring in $\mathcal{M}(q_{\mathcal{S}})$, whereas $\vec{z}$ denotes the set of fresh existential variables introduced by $\mathcal{M}(q_{\mathcal{S}})$

2: **return** $q_{\mathcal{O}}$

Roughly speaking, the algorithm computes an *EQL-Lite*$^-$(UCQ) query $q_{\mathcal{O}}$ by first chasing (the incomplete $\mathcal{S}$-database associated to) $q_{\mathcal{S}}$ with respect to $\mathcal{M}$, using $\top$ to bind possible distinguished variables of $q_{\mathcal{S}}$ that are not involved in $\mathcal{M}(q_{\mathcal{S}})$, and then using the epistemic operator to bind existential variables coming from $q_{\mathcal{S}}$. Note, in particular, that the latter is achieved by pushing the subset $\vec{y}'$ of the existential variables $\vec{y}$ of $q_{\mathcal{S}}$ occurring in $\mathcal{M}(q_{\mathcal{S}})$ inside the $\mathbf{K}$ operator. Finally, $\vec{z}$ denotes the set of fresh existential variables introduced by the chase.

Example 9.7. Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the following OBDM specification:

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

$$\begin{array}{lrcl} m_1: & \exists y.s_1(x_1,x_2) \wedge s_2(x_2,y) & \rightarrow & \exists z.P(x_1,z) \wedge P(z,x_2), \\ m_2: & \exists y_1,y_2,y_3.s_2(x,y_1) \wedge s_3(y_1,y_2,y_3) & \rightarrow & \exists z.P'(x,z), \\ m_3: & \exists y.s_3(y,x,c_1) & \rightarrow & A(x). \end{array}$$

Let the data service be the CQ $q_{\mathcal{S}} = \{(x_1,x_2) \mid \exists y_1,y_2.s_1(x_1,y_1) \wedge s_2(y_1,y_2) \wedge s_3(y_2,x_2,c_2)\}$ over $\mathcal{S}$. One can verify that MinimallyCompleteEpistemic$(\Sigma, q_{\mathcal{S}})$ returns the *EQL-Lite*$^-$(UCQ) query $q_{\mathcal{O}} = \{(x_1,x_2) \mid \exists y_1.\mathbf{K}(\exists z_1,z_2.P(x_1,z_1) \wedge P(z_1,y_1) \wedge P'(y_1,z_2) \wedge \top(x_2))\}$, which corresponds to the unique (up to equivalence w.r.t. Σ) minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting in the class of *EQL-Lite*$^-$(UCQ) queries. $\square$

It is worth noting that, even though the "$\neg$" operator is available in the *EQL-Lite*$^-$(UCQ) query language, the queries returned by the MinimallyCompleteEpistemic algorithm are free from this operator. The *EQL-Lite*(UCQ) queries without occurrences of the "$\neg$" operator enjoy the following interesting and useful property:

Proposition 9.3. *Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification, and let $q_{\mathcal{O}}$ be an EQL-Lite(UCQ) query over $\mathcal{O}$ without occurrences of the "$\neg$" operator. Then, the query $\mathsf{EQLPerfRef}_{q_{\mathcal{O}},\Sigma}$ over $\mathcal{S}$ can be always expressed as an equivalent UCQ.*

Proof. As illustrated in Subsection 9.1.1, the query $\mathsf{EQLPerfRef}_{q_{\mathcal{O}},\Sigma}$ is obtained from $q_{\mathcal{O}}$ by replacing each of its epistemic atoms $\mathbf{K}\varrho_i$ with the logical body of the UCQ $\mathsf{PerfRef}_{q_{\mathcal{O}}^{\varrho_i},\Sigma}$ over $\mathcal{S}$, where ϱ_i and $q_{\mathcal{O}}^{\varrho_i}$ are the body of a UCQ over $\mathcal{O}$ and the UCQ over $\mathcal{O}$ associated to ϱ_i, respectively. Thus, since by assumption in $q_{\mathcal{O}}$ there are no occurrences of the "$\neg$" operator, and since it is so even in $\mathsf{PerfRef}_{q_{\mathcal{O}}^{\varrho_i},\Sigma}$ for each epistemic atom $\mathbf{K}\varrho_i$ occurring in $q_{\mathcal{O}}$, we have that $\mathsf{EQLPerfRef}_{q_{\mathcal{O}},\Sigma}$ is a FOL query whose operators occurring in its body are only $\wedge$, $\vee$, and $\exists$. But then, it is trivial to see that a FOL query with only such operators occurring in its body can be always expressed as an equivalent UCQ, as required. $\square$

Moreover, the *EQL-Lite*$^-$(UCQ) queries of the same shape as those returned by the MinimallyCompleteEpistemic algorithm further enjoy the following property:

Proposition 9.4. *Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification, D be an $\mathcal{S}$-database consistent with Σ, and $q_{\mathcal{O}}$ be an EQL-Lite$^-$(UCQ) query $q_{\mathcal{O}}$ over $\mathcal{O}$ of the form $q_{\mathcal{O}} = \{\vec{t} \mid \exists \vec{y}.\mathbf{K}(\phi)\}$, where ϕ is the body of a CQ. We have that $\vec{c} \in cert^{D}_{q_{\mathcal{O}},\Sigma}$ if and only if there is a function h from the set of terms occurring in ϕ to the set of terms occurring in $\mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}}$ for which (i) $h(y)$ is a constant, for each $y \in \vec{y}$; (ii) $h(c) = c$, for each constant c; (iii) $h(\phi) \subseteq \mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}}$, where $h(\phi)$ is the image of ϕ under h; and finally (vi) $h(\vec{t}) = \vec{c}$ (h is also called a homomorphism from $q_{\mathcal{O}}$ to $\mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}}$ for which $h(y)$ is a constant, for each $y \in \vec{y}$, and $h(\vec{t}) = \vec{c}$).*

Proof. We observe that (*ii*), (*iii*), and (*vi*) are the necessary and sufficient conditions that a CQ q over $\mathcal{O}$ of the form $q = \{\vec{t} \mid \phi\}$ (where ϕ is the body of a CQ) must satisfy to be such that $\vec{c} \in cert^{D}_{q,\Sigma}$ for a tuple of constants $\vec{c}$ (cf. Chapter 2).

We further observe that, by combining results of [Fagin *et al.*, 2005a, Proposition 4.2] with [Calvanese *et al.*, 2007b, Theorem 29], it is well-known that there is a homomorphism from $\mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}}$ to $\mathcal{I}$ (where this latter seen as a set of facts over $\mathcal{O}$), for each model $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ of $\langle \Sigma, D \rangle$ (i.e., for each interpretation $\mathcal{I} \in Mod_D(\Sigma)$).

From the above observations, and due to the semantic meaning of the existential variables occurring outside the epistemic operator $\mathbf{K}$ of an *EQL* formula, the claim of the proposition can be easily verified. $\square$

By exploiting the above results, we are now ready to establish termination and correctness of the MinimallyCompleteEpistemic algorithm.

Theorem 9.4. *Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification where $\mathcal{O}$ is a DL-Lite$^-_{\mathcal{R}}$ ontology and $\mathcal{M}$ is a GLAV mapping, and let $q_{\mathcal{S}}$ be a CQ over $\mathcal{S}$. We have that MinimallyCompleteEpistemic$(\Sigma, q_{\mathcal{S}})$ terminates and returns the unique (up to equivalence w.r.t. Σ) minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of EQL-Lite$^-$(UCQ) queries.*

Proof. Analogously to the MinimallyComplete algorithm illustrated in Chapter 5, the termination of the MinimallyCompleteEpistemic algorithm easily follows from the termination of the chase of a source instance (possibly containing variables) with respect to a GLAV mapping, or, equivalently, with respect to a set of source-to-target tgds [Fagin *et al.*, 2005a].

As for the correctness, we first show that the query $q_{\mathcal{O}} = \{\vec{t} \mid \exists\vec{y}.\mathbf{K}(\exists\vec{z}.\mathcal{M}(q_{\mathcal{S}}) \wedge \top(\vec{x}))\}$ returned by the algorithm is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. Observe that the possibly introduced fresh existential variables in $\vec{z}$ do not appear outside the epistemic operator $\mathbf{K}$, and therefore the CQ $q_{\mathcal{S}}$ corresponds to, or it is contained in, a disjunct of $\mathsf{EQLPerfRef}_{q_{\mathcal{O}},\Sigma}$ (observe that, due to Proposition 9.3, the query $\mathsf{EQLPerfRef}_{q_{\mathcal{O}},\Sigma}$ can be expressed as an equivalent UCQ). Thus, due to Corollary 9.1, the fact that $q_{\mathcal{S}} \sqsubseteq \mathsf{EQLPerfRef}_{q_{\mathcal{O}},\Sigma}$ implies that $q_{\mathcal{O}}$ is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. We now show that $q_{\mathcal{O}}$ is actually the unique (up to equivalence w.r.t. Σ) minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of *EQL-Lite*$^{-}$(UCQ) queries, that is, each *EQL-Lite*$^{-}$(UCQ) query $q'_{\mathcal{O}}$ that is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ is such that $cert_{q_{\mathcal{O}},\Sigma} \sqsubseteq cert_{q'_{\mathcal{O}},\Sigma}$ (cf. Definition 3.7). We do this by way of contradiction.

Let $q'_{\mathcal{O}}$ be an *EQL-Lite*$^{-}$(UCQ) query such that $cert_{q_{\mathcal{O}},\Sigma} \not\sqsubseteq cert_{q'_{\mathcal{O}},\Sigma}$, that is, there exists an $\mathcal{S}$-database D consistent with Σ such that $cert^{D}_{q_{\mathcal{O}},\Sigma} \not\subseteq cert^{D}_{q'_{\mathcal{O}},\Sigma}$. It follows that there is a tuple of constant $\vec{c} = (c_1, \ldots, c_n)$ such that $\vec{c} \notin cert^{D}_{q'_{\mathcal{O}},\Sigma}$, but $\vec{c} \in cert^{D}_{q_{\mathcal{O}},\Sigma}$. Since $q'_{\mathcal{O}}$ is an *EQL-Lite*$^{-}$(UCQ), there are two possible cases: either $q'_{\mathcal{O}}$ contains a part of the form $\neg\delta$ (with $\delta \not\equiv \bot$) in its body, or not.

In the former case, consider the $\mathcal{S}$-database $D' \supseteq D$ in which each source predicate $s \in \mathcal{S}$ contains all possible tuples of constants occurring in D whose arity is the one of s. Obviously, we have that $\vec{c} \in q_{\mathcal{S}}^{D'}$, and, since $\mathcal{O}$ is a *DL-Lite*$^{-}_{\mathcal{R}}$ ontology, D' is consistent with Σ (i.e., $Mod_D(\Sigma) \neq \emptyset$). Furthermore, by construction of D' and the fact that in the *EQL-Lite*$^{-}$(UCQ) query language nested negation is disallowed (i.e., the *EQL* formula δ occurring in the body of $q'_{\mathcal{O}}$ must be of the form $\exists\vec{y}.\mathbf{K}(\varrho)$, where ϱ is the body of a UCQ), we have that the formula δ occurring in the body of $q'_{\mathcal{O}}$ is *EQL*-logically implied by $\langle\Sigma, D'\rangle$ (and thus, formula $\neg\delta$ is *not EQL*-logically implied by $\langle\Sigma, D'\rangle$) when replacing the free variables of $q'_{\mathcal{O}}$ with any tuple of constants. Since by construction of the *EQL-Lite*$^{-}$(UCQ) query language formula $\neg\delta$ in $q'_{\mathcal{O}}$ occurs either alone or in conjunction to a subformula of $q'_{\mathcal{O}}$, this clearly implies that $q'_{\mathcal{O}}$ is not *EQL*-logically implied by $\langle\Sigma, D'\rangle$ when replacing the free variables of $q'_{\mathcal{O}}$ with any tuple of constants, and therefore also for the tuple of constants $\vec{c}$. As a result, we have that $\vec{c} \notin cert^{D'}_{q'_{\mathcal{O}},\Sigma}$. But then, the facts that $\vec{c} \in q_{\mathcal{S}}^{D'}$ and $\vec{c} \notin cert^{D'}_{q'_{\mathcal{O}},\Sigma}$ imply that $q'_{\mathcal{O}}$ is not a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, as required.

Consider the latter case, that is, the *EQL-Lite*$^{-}$(UCQ) query $q'_{\mathcal{O}}$ is of the form $q'_{\mathcal{O}} = \{\vec{t'} \mid \exists\vec{y'}.\mathbf{K}(\varrho_1) \wedge \ldots \wedge \mathbf{K}(\varrho_m)\}$ with ϱ_i being the body of a UCQ, for each $i \in [1, m]$. In this case, it is possible to proceed in a similar way to the proof of Theorem 5.2. Consider the freezing of $q_{\mathcal{S}} = \{\vec{t} \mid \exists\vec{y}.\phi(\vec{x}, \vec{y})\}$, i.e., the set D' of all facts over $\mathcal{S}$ obtained from $\phi(\vec{x}, \vec{y})$ by replacing each variable $v \in \vec{x} \cup \vec{y}$ with a different fresh constant denoted by c_v. Let, moreover, $\vec{c'} = (c'_1, \ldots, c'_n)$ be the freezed tuple of constants where, for each $i \in [1, n]$, $c'_i = t_i$ if t_i is a constant, and $c'_i = c_x$ if $t_i = x$ (i.e., $t_i \in \vec{x}$ is a distinguished variable). Obviously, we have that $\vec{c'} \in q_{\mathcal{S}}^{D'}$

holds by construction, and, since $\mathcal{O}$ is a $\mathit{DL\text{-}Lite}_{\mathcal{R}}^{-}$ ontology, D' is consistent with Σ (i.e., $\mathit{Mod}_{D'}(\Sigma) \neq \emptyset$). We now prove that $\vec{c'} \notin \mathit{cert}_{q'_{\mathcal{O}},\Sigma}^{D'}$.

Consider $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$, i.e., the canonical structure of $\mathcal{O}$ with respect to $\mathcal{M}$ and D. Since $\vec{c} \in \mathit{cert}_{q_{\mathcal{O}},\Sigma}^{D}$, due to Proposition 9.4, we derive that there exists a homomorphism h from $q_{\mathcal{O}}$ to $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ for which (i) $h(\mathcal{y})$ is a constant, for each $\mathcal{y} \in \vec{\mathcal{y}}$, and (ii) $h(\vec{t}) = \vec{c}$. Due to the facts that $\mathcal{M}$ is a GLAV mapping and $\mathcal{O}$ is a $\mathit{DL\text{-}Lite}_{\mathcal{R}}^{-}$ ontology, and due to the existence of such homomorphism h, by construction of $q_{\mathcal{O}}$ and D' it is easy to see the existence of a function f from $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D')}$ to $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ for which (i) $f(c_{\mathcal{y}}) = h(\mathcal{y})$, for each $\mathcal{y} \in \vec{\mathcal{y}}$ (recall that $\vec{\mathcal{y}} \subseteq \vec{y}$ is the subset of existential variables of $q_{\mathcal{S}}$ occurring in $\mathcal{M}(q_{\mathcal{S}})$), (ii) $f(c) = h(c) = c$, for each constant c occurring in $q_{\mathcal{O}}$, (iii) $f(c_x) = h(x)$, for each distinguished variable $x \in \vec{x}$ of $q_{\mathcal{S}}$ occurring in $\mathcal{M}(q_{\mathcal{S}})$, and (vi) $f(\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D')}) \subseteq \mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$, where $f(\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D')})$ is the image of $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D')}$ under f. Observe that $f(\vec{c'}) = \vec{c}$ by construction.

Furthermore, by exploiting again Proposition 9.4, it is easy to verify that, since $\vec{c} \notin \mathit{cert}_{q'_{\mathcal{O}},\Sigma}^{D}$ by assumption, then there is some epistemic atom $\mathbf{K}(\varrho_i)$ in the body of $q'_{\mathcal{O}}$ such that for no disjunct ϕ in ϱ_i there is a homomorphism h from ϕ to $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ for which (i) $h(y')$ is a constant, for each $y' \in \vec{y'}$ occurring in ϕ, and (ii) $h(\vec{t'}) = \vec{c}$. But then, due to the existence of the function f, and due to the above consideration, we derive that there there is no disjunct ϕ in ϱ_i such that that there is a homomorphism h' from ϕ to $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D')}$ for which (i) $h'(y')$ is a constant, for each $y' \in \vec{y'}$ occurring in ϕ, and (ii) $h'(\vec{t'}) = \vec{c'}$ (otherwise, the composition function $f \circ h'$ would result in a homomorphism from ϕ to $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ for which (i) $f(h(y'))$ is a constant, for each $y' \in \vec{y'}$ occurring in ϕ, and (ii) $f(h'(\vec{t'})) = \vec{c}$, and therefore this would contradict the assumption that for no disjunct ϕ in ϱ_i there is such a homomorphism). Using Proposition 9.4, it can be easily proven that this implies that $\vec{c'} \notin \mathit{cert}_{q'_{\mathcal{O}},\Sigma}^{D'}$. To conclude the proof, observe that D' is an $\mathcal{S}$-database consistent with Σ for which $\vec{c'} \in q_{\mathcal{S}}^{D'}$ and $\vec{c'} \notin \mathit{cert}_{q'_{\mathcal{O}},\Sigma}^{D'}$, thus implying that $q'_{\mathcal{O}}$ is not a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, as required. $\square$

As for the running time of the MinimallyCompleteEpistemic algorithm, we observe that it is independent of both the size of $\mathcal{O}$ and $\mathcal{S}$, and, due to the application of the chase, it is polynomial in the size of $q_{\mathcal{S}}$ and exponential in the size of $\mathcal{M}$.

It remains to address the case of perfect source-to-ontology rewritings in the class of *EQL-Lite*$^{-}$(UCQ) queries. Consider any pair composed by an OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ where $\mathcal{O}$ is a $\mathit{DL\text{-}Lite}_{\mathcal{R}}^{-}$ ontology and $\mathcal{M}$ is a GLAV mapping and a CQ $q_{\mathcal{S}}$ over $\mathcal{S}$. Clearly, by definition, either the unique (up to equivalence w.r.t. Σ) minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of *EQL-Lite*$^{-}$(UCQ) queries is also a sound, and therefore a perfect, $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, or no perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of *EQL-Lite*$^{-}$(UCQ) queries exists.

With this observation at hand, we can easily derive the algorithm PerfectEpistemic together with its termination and correctness.

Essentially, the algorithm first computes the query $q_{\mathcal{O}}$ which is the unique (up to equivalence w.r.t. Σ) minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of *EQL-Lite*$^{-}$(UCQ) queries, and then, by exploiting Corollary 9.1, checks

Algorithm 9.2 PerfectEpistemic

Input:

OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ where $\mathcal{O}$ is a *DL-Lite*$_{\mathcal{R}}^{-}$ ontology and $\mathcal{M}$ is a GLAV mapping;
CQ $q_{\mathcal{S}}$ over $\mathcal{S}$

Output:

either an *EQL-Lite*$^{-}$(UCQ) $q_{\mathcal{O}}$ over $\mathcal{O}$, or report that "no perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of *EQL-Lite*$^{-}$(UCQ) queries exists"

$q_{\mathcal{O}} :=$ MinimallyCompleteEpistemic$(\Sigma, q_{\mathcal{S}})$
if $\mathsf{EQLPerfRef}_{q_{\mathcal{O}},\Sigma} \sqsubseteq q_{\mathcal{S}}$ **then**
  return $q_{\mathcal{O}}$
else
  return "no perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of *EQL-Lite*$^{-}$(UCQ) queries exists"
end if

whether $q_{\mathcal{O}}$ is also a sound, and therefore a perfect, $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ via the following query containment check: $\mathsf{EQLPerfRef}_{q_{\mathcal{O}},\Sigma} \sqsubseteq q_{\mathcal{S}}$ (note that $\mathcal{V}_{\mathcal{O}} \equiv \bot$ for each *DL-Lite*$_{\mathcal{R}}^{-}$ ontology $\mathcal{O}$). We point out that the above containment is actually a containment of UCQs. Indeed, on the one hand, the query $q_{\mathcal{S}}$ is a CQ, and, on the other hand, for each OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and *EQL-Lite*(UCQ) query $q_{\mathcal{O}}$ over $\mathcal{O}$ without any occurrence of the "$\neg$" operator (as all those returned by the MinimallyCompleteEpistemic algorithm), the FOL query $\mathsf{EQLPerfRef}_{q_{\mathcal{O}},\Sigma}$ can be in fact expressed as a UCQ. This allows us to conclude that the overall running time of the PerfectEpistemic algorithm is exponential in the size of the input.

Theorem 9.5. *Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification where $\mathcal{O}$ is a DL-Lite$_{\mathcal{R}}^{-}$ ontology and $\mathcal{M}$ is a GLAV mapping, and let $q_{\mathcal{S}}$ be a CQ over $\mathcal{S}$. We have that PerfectEpistemic$(\Sigma, q_{\mathcal{S}})$ terminates and returns the perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ if it exists and can be expressed as an EQL-Lite^{-}(UCQ) query, otherwise it reports that no perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of EQL-Lite^{-}(UCQ) queries exists.*

9.4 The case of One-To-One Mappings

In this section, we study the problem of computing source-to-ontology rewritings in the class of *EQL-Lite*(UCQ) queries, for the case of OBDM specifications where the mapping assertions are a special case of GLAV mapping assertions considered so far.

Specifically, we consider OBDM specifications where the mapping language follows the "One-to-One approach". A *One-To-One* mapping assertion is a GLAV mapping assertion $\forall \vec{x}.(\exists \vec{y}.\phi(\vec{x}, \vec{y}) \rightarrow \exists \vec{z}.\varphi(\vec{x}, \vec{z}))$ in which both $\phi(\vec{x}, \vec{y})$ and $\varphi(\vec{x}, \vec{z})$ are simply atoms without constants or repeated variables. We say that a mapping $\mathcal{M}$ is a One-To-One mapping if it consists of a finite set of One-To-One mapping assertions. Observe that GAV (respectively, pure GAV), LAV, and One-To-One are pairwise incomparable approaches for specifying mappings.

Example 9.8. The mapping $\mathcal{M}$ defined in Example 9.1 is a One-To-One mapping. Observe that the mapping assertion $m_4 \in \mathcal{M}$ is neither a GAV, nor a LAV mapping assertion. The pure GAV mapping assertion $s_1(x) \wedge s_2(x) \rightarrow A(x)$ is neither a One-To-One, nor a LAV mapping assertion, whereas the LAV mapping assertion $s_1(x) \rightarrow \exists z. P(x, z) \wedge A(z)$ is neither a One-To-One, nor a GAV mapping assertion. □

We now present a property for One-To-One mappings, which is crucial for the technical treatment of this section.[3]

Proposition 9.5. *Let $\mathcal{M}$ be a One-To-One mapping relating a schema $\mathcal{S}$ to an ontology $\mathcal{O}$. For each $\mathcal{S}$-database D, we have that $\mathcal{M}(D) = \bigcup_{\alpha \in D} \mathcal{M}(\alpha)$.*

Proof. The claim trivially follows by considering that the left-hand side of each One-to-One mapping assertion is constituted simply by a single atom. Thus, in order to compute $\mathcal{M}(D)$ for a One-To-One mapping $\mathcal{M}$ and an $\mathcal{S}$-database D, it is sufficient to consider separately each fact $\alpha \in D$ and take the union of the sets of atoms $\mathcal{M}(\alpha)$, as required. □

In what follows in this section, we implicitly assume that each mapping $\mathcal{M}$ relating a schema $\mathcal{S}$ to an ontology $\mathcal{O}$ is a One-To-One mapping.

9.4.1 Complete Source-to-Ontology Rewritings

Interestingly, we now prove that, when the mapping language follows the One-To-One approach, one can use the MinimallyCompleteEpistemic algorithm to compute also minimally complete source-to-ontology rewritings of CQs over the source schema when the target query language is the class of *EQL-Lite*(UCQ) queries, rather than its fragment of *EQL-Lite*$^-$(UCQ) queries considered in the previous section. Thus proving that, for each pair composed by an OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ where $\mathcal{O}$ is a *DL-Lite*$_{\mathcal{R}}^-$ ontology and $\mathcal{M}$ is a One-To-One mapping and a CQ $q_{\mathcal{S}}$ over $\mathcal{S}$, the unique (up to equivalence w.r.t. Σ) minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of *EQL-Lite*(UCQ) queries is guaranteed to exists and, moreover, it can be expressed as an *EQL-Lite*$^-$(UCQ) query. Before delving into the technical part, we now illustrate its application within a scenario with a One-To-One mapping.

Example 9.9. Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the following OBDM specification:

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

$$
\begin{array}{lrcl}
m_1 : & s_1(x_1, x_2) & \rightarrow & P_1(x_1, x_2), \\
m_2 : & \exists y. s_2(x, y) & \rightarrow & \exists z. P_2(x, z), \\
m_3 : & \exists y_1, y_2. s_3(x, y_1, y_2) & \rightarrow & A(x), \\
m_4 : & \exists y_1, y_2, y_3. s_3(y_1, y_2, y_3) & \rightarrow & \exists z_1, z_2. P_3(z_1, z_2).
\end{array}
$$

[3] Note that Proposition 9.5 is valid not only for One-To-One mapping, but also for LAV mapping.

Let the data service be the CQ $q_{\mathcal{S}} = \{(x) \mid \exists y_1, y_2.s_1(x, y_1) \land s_2(x, y_2) \land s_3(y_1, y_1, y_2)\}$ over $\mathcal{S}$. One can verify that $\mathsf{MinimallyCompleteEpistemic}(\Sigma, q_{\mathcal{S}})$ returns the *EQL-Lite*$^-$(UCQ) query $q_{\mathcal{O}} = \{(x) \mid \exists y_1.\mathbf{K}(\exists z_1, z_2, z_3.P_1(x, y_1) \land P_2(x, z_1) \land A(y_1) \land P_3(z_2, z_3))\}$, which corresponds to the unique (up to equivalence w.r.t. Σ) minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting in the class of *EQL-Lite*(UCQ) queries. □

Theorem 9.6. *Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification where $\mathcal{O}$ is a DL-Lite$_{\mathcal{R}}^-$ ontology and $\mathcal{M}$ is a One-To-One mapping, and let $q_{\mathcal{S}}$ be a CQ over $\mathcal{S}$. We have that $\mathsf{MinimallyCompleteEpistemic}(\Sigma, q_{\mathcal{S}})$ terminates and returns the unique (up to equivalence w.r.t. Σ) minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of EQL-Lite(UCQ) queries.*

Proof. Termination of the algorithm, as well as the fact that it returns a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ (in fact, the unique (up to equivalence w.r.t. Σ) minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of *EQL-Lite*$^-$(UCQ) queries), has already been discussed in the proof of Theorem 9.4 for the more general case of when $\mathcal{M}$ is a GLAV mapping rather than a One-To-One mapping. We now show that, when $\mathcal{M}$ is a One-To-One mapping, the computed *EQL-Lite*$^-$(UCQ) query $q_{\mathcal{O}}$ is actually the unique (up to equivalence w.r.t. Σ) minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ even in the class of *EQL-Lite*(UCQ) queries, that is, each *EQL-Lite*(UCQ) query $q'_{\mathcal{O}}$ that is a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ is such that $cert_{q_{\mathcal{O}},\Sigma} \sqsubseteq cert_{q'_{\mathcal{O}},\Sigma}$ (cf. Definition 3.7). We do this by way of contradiction.

Let $q'_{\mathcal{O}}$ be an *EQL-Lite*(UCQ) query such that $cert_{q_{\mathcal{O}},\Sigma} \not\sqsubseteq cert_{q'_{\mathcal{O}},\Sigma}$, that is, there exists an $\mathcal{S}$-database D consistent with Σ such that $cert^D_{q_{\mathcal{O}},\Sigma} \not\subseteq cert^D_{q'_{\mathcal{O}},\Sigma}$. It follows that there is a tuple of constant $\vec{c}$ such that $\vec{c} \notin cert^D_{q'_{\mathcal{O}},\Sigma}$, but $\vec{c} \in cert^D_{q_{\mathcal{O}},\Sigma}$. We now exhibit an $\mathcal{S}$-database D' consistent with Σ for which (i) $\vec{c} \in q_{\mathcal{S}}^{D'}$, and (ii) $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D')}$ and $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ are *homomorphically equivalent*, i.e., there is a homomorphism from $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D')}$ to $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ and vice versa.

Since $\vec{c} \in cert^D_{q_{\mathcal{O}},\Sigma}$, due to Proposition 9.4, we derive that there is a homomorphism h from $q_{\mathcal{O}}$ to $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ for which (i) $h(\mathcal{y})$ is a constant, for each $\mathcal{y} \in \vec{\mathcal{y}}$, and (ii) $h(\vec{t}) = \vec{c}$. Let h' be the function extending h by assigning a different fresh constant c_y to each existential variable $y \in \vec{y} \setminus \vec{\mathcal{y}}$ (i.e., to each existential variable $y \in \vec{y}$ of $q_{\mathcal{S}}$ not occurring in $\mathcal{M}(q_{\mathcal{S}})$). Consider now $h'(q_{\mathcal{S}})$ and $h(q_{\mathcal{O}}) \subseteq \mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$, i.e., the set of facts corresponding to the image of (the body of) $q_{\mathcal{S}}$ under h' and the set of atoms corresponding to the image of (the body of) $q_{\mathcal{O}}$ under h, respectively (this latter can be a set of atoms because, for some existential variable $z \in \vec{z}$ of $q_{\mathcal{O}}$, $h(z)$ is allowed to be a variable of $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$). Since the left-hand side of each mapping assertion in $\mathcal{M}$ is a single atom without constants or repeated variables, and since $h(q_{\mathcal{O}}) \subseteq \mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ with $h(\mathcal{y})$ being a constant for each $\mathcal{y} \in \vec{\mathcal{y}}$, by construction of $q_{\mathcal{S}}$ and $q_{\mathcal{O}}$, it is not hard to ascertain that, for each fact $\alpha \in h'(q_{\mathcal{S}})$, the chase of fact α with respect to $\mathcal{M}$, i.e., $\mathcal{M}(\alpha)$, is such that there exists a homomorphism from $\mathcal{M}(\alpha)$ to $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$.

But then, the $\mathcal{S}$-database we are seeking is $D' = D \cup h'(q_{\mathcal{S}})$. Indeed, on the one hand, $\vec{c} \in q_{\mathcal{S}}^{D'}$ trivially holds because (i) $\vec{c} \in q_{\mathcal{S}}^{D}$, (ii) $D \subseteq D'$, and (iii) $q_{\mathcal{S}}$ is a CQ. On the other hand, using Proposition 9.5 and the fact that there exists a homomorphism from $\mathcal{M}(\alpha)$ to $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ for each $\alpha \in h'(q_{\mathcal{S}})$, we easily derive that $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D')}$ and $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ are homomorphically equivalent.

To conclude the proof observe that, since $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D')}$ and $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ are homomorphically equivalent, and since $\vec{c} \notin cert^{D}_{q'_{\mathcal{O}},\Sigma}$ by assumption, it is easy to see that $\vec{c} \notin cert^{D'}_{q'_{\mathcal{O}},\Sigma}$ as well. It follows that D' is an $\mathcal{S}$-database consistent with Σ (observe that $\mathcal{O}$ is a *DL-Lite*$_{\mathcal{R}}^{-}$ ontology, and therefore each $\mathcal{S}$-database is consistent with Σ) for which $\vec{c} \in q_{\mathcal{S}}^{D'}$ and $\vec{c} \notin cert^{D'}_{q'_{\mathcal{O}},\Sigma}$, thus implying that $q'_{\mathcal{O}}$ is not a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, as required. $\square$

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

Corollary 9.2. *Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification where $\mathcal{O}$ is a DL-Lite$_{\mathcal{R}}^{-}$ ontology and $\mathcal{M}$ is a One-To-One mapping, and let $q_{\mathcal{S}}$ be a CQ over $\mathcal{S}$. Then, the unique (up to equivalence w.r.t. Σ) minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of EQL-Lite(UCQ) queries can be expressed as an EQL-Lite^{-}(UCQ) query.*

Observe that, when dealing with One-To-One mapping assertions, the application of the chase is feasible in polynomial time even in the size of the mapping $\mathcal{M}$, and therefore the overall running time of the MinimallyCompleteEpistemic algorithm becomes polynomial in the size of the input when $\mathcal{M}$ is a One-To-One mapping.

9.4.2 Sound Source-to-Ontology Rewritings

We now investigate the maximally sound case. Specifically, for OBDM specifications with One-To-One mapping assertions, we now provide the algorithm MaximallySoundEpistemic for computing maximally sound source-to-ontology rewritings of CQJFEs over the source schema when the target query language is the class of *EQL-Lite*(UCQ) queries. Thus proving that, for each pair composed by an OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ where $\mathcal{O}$ is a *DL-Lite*$_{\mathcal{R}}^{-}$ ontology and $\mathcal{M}$ is a One-To-One mapping and a CQJFE $q_{\mathcal{S}}$ over $\mathcal{S}$, the unique (up to equivalence w.r.t. Σ) maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of *EQL-Lite*(UCQ) queries is guaranteed to exists and, moreover, as for the minimally complete case, it can be expressed as an *EQL-Lite*$^{-}$(UCQ) query.

In the algorithm, since $\mathcal{M}$ is a One-To-One mapping, when computing the reformulation $\mathsf{EQLPerfRef}_{q^c_{\mathcal{O}},\Sigma}$ over $\mathcal{S}$ of the *EQL-Lite*$^{-}$(UCQ) query $q^c_{\mathcal{O}}$, we can assume that $\mathsf{PerfRef}_{q^{\varrho}_{\mathcal{O}},\Sigma}$ of the epistemic atom $\mathbf{K}\varrho$ occurring in $q^c_{\mathcal{O}}$ is obtained by first reformulating $q^{\varrho}_{\mathcal{O}}$ with respect to $\mathcal{O}$, and then by unfolding the resulting UCQ with respect to $\mathcal{M}$ (with the proviso that, when unfolding an atom β over $\mathcal{O}$, if a mapping assertion m is such that the k-th argument of the atom in its right-hand side is an existential variable whilst the k-th argument of β is either a distinguished variable or a constant, then m have to be ignored). Furthermore, since the "$\neg$" operator never occurs in the *EQL-Lite*$^{-}$(UCQ) query $q^c_{\mathcal{O}}$ of the algorithm, we can implicitly assume that its reformulation $\mathsf{EQLPerfRef}_{q^c_{\mathcal{O}},\Sigma}$ over $\mathcal{S}$ is first computed adhering to the above procedure, and then turned into an equivalent UCQ (cf. Proposition 9.3).

In a nutshell, the MaximallySoundEpistemic algorithm starts by checking whether there is some distinguished variable of $q_{\mathcal{S}}$ not appearing in $\mathcal{M}(q_{\mathcal{S}})$, and if this is the case, then it returns the query $\{\vec{t} \mid \bot(\vec{x})\}$. Otherwise, the algorithm first computes the query $q^c_{\mathcal{O}} = \{\vec{t} \mid \psi(\vec{x})\}$, where $\psi(\vec{x}) = \exists \vec{y}.\mathbf{K}(\exists \vec{z}.\mathcal{M}(q_{\mathcal{S}}))$, which corresponds to the minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of *EQL-Lite*(UCQ) queries, and then, for each disjunct $q^i_{\mathcal{S}}$ in its reformulation $\mathsf{EQLPerfRef}_{q^c_{\mathcal{O}},\Sigma}$ over $\mathcal{S}$ such that

Algorithm 9.3 MaximallySoundEpistemic

Input:

OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ where $\mathcal{O}$ is a *DL-Lite*$_{\mathcal{R}}^{-}$ ontology and $\mathcal{M}$ is a One-to-One mapping;

CQJFE $q_{\mathcal{S}} = \{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$ over $\mathcal{S}$

Output:

EQL-Lite^{-}(UCQ) query $q_{\mathcal{O}}$ over $\mathcal{O}$

if there is a distinguished variable $x \in \vec{x}$ not occurring in $\mathcal{M}(q_{\mathcal{S}})$ **then**
  return $q_{\mathcal{O}} := \{\vec{t} \mid \bot(\vec{x})\}$
end if
Compute $q_{\mathcal{O}}^{c} :=$ MinimallyCompleteEpistemic$(\Sigma, q_{\mathcal{S}})$
Let $\psi(\vec{x})$ be the body of $q_{\mathcal{O}}^{c}$, i.e., $\psi(\vec{x}) := \exists \vec{y}.\mathbf{K}(\exists \vec{z}.\mathcal{M}(q_{\mathcal{S}}))$
for each CQ $q_{\mathcal{S}}^{i} = \{\vec{t} \mid \exists \vec{y_i}.\phi_i(\vec{x}, \vec{y_i})\} \in \mathsf{EQLPerfRef}_{q_{\mathcal{O}}^{c},\Sigma}$ **do**
  if $q_{\mathcal{S}}^{i} \not\sqsubseteq q_{\mathcal{S}}$ **then**
    $\psi(\vec{x}) := \psi(\vec{x}) \wedge \neg \exists \vec{\mathcal{y}_i}.\mathbf{K}(\exists \vec{z_i}.\mathcal{M}(q_{\mathcal{S}}^{i}))$, where $\vec{\mathcal{y}_i} \subseteq \vec{y_i}$ denotes the subset of existential variables of $q_{\mathcal{S}}^{i}$ occurring in $\mathcal{M}(q_{\mathcal{S}}^{i})$ and not in $q_{\mathcal{S}}$, whereas $\vec{z_i}$ denotes the set of fresh existential variables introduced by $\mathcal{M}(q_{\mathcal{S}}^{i})$
  end if
end for
return $q_{\mathcal{O}} := \{\vec{t} \mid \psi(\vec{x})\}$

$q_{\mathcal{S}}^{i} \not\sqsubseteq q_{\mathcal{S}}$, it adds in conjunction to $\psi(\vec{x})$ the negation of the minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}^{i}$ in the class of *EQL-Lite*(UCQ) queries. Intuitively, by doing so, for each $\mathcal{S}$-database D, the algorithm prevents the returned *EQL-Lite*$^{-}$(UCQ) query $q_{\mathcal{O}}$ to have tuples of constants in its certain answers with respect to Σ and D that are not in the evaluation of $q_{\mathcal{S}}$ over D.

The next example illustrates the algorithm.

Example 9.10. Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the following OBDM specification:

- $\mathcal{O} = \{\ A_1 \sqsubseteq \exists P_1, A_2 \sqsubseteq \exists P_2\ \}$;
- $\mathcal{S} = \{\ s_1, s_2, s_3, s_4, s_5, s_6\ \}$;
- $\mathcal{M} = \{\ m_1, m_2, m_3, m_4, m_5, m_6, m_7\ \}$, where:

$$\begin{array}{lrcl}
m_1: & s_1(x_1, x_2) & \rightarrow & P_1(x_1, x_2), \\
m_2: & \exists y.s_2(x, y) & \rightarrow & \exists z.P_2(x, z), \\
m_3: & \exists y.s_3(x_1, x_2, y) & \rightarrow & P_2(x_1, x_2), \\
m_4: & \exists y_1, y_2.s_4(x, y_1, y_2) & \rightarrow & A_2(x), \\
m_5: & \exists y_1, y_2.s_4(y_1, y_2, x) & \rightarrow & \exists z.P_3(x, z), \\
m_6: & s_5(x) & \rightarrow & \exists z.P_1(x, z), \\
m_7: & s_6(x) & \rightarrow & A_1(x).
\end{array}$$

Let the data service be the CQJFE $q_{\mathcal{S}} = \{(x) \mid \exists y_1, y_2.s_1(x, y_1) \wedge s_2(x, y_2)\}$ over $\mathcal{S}$. The algorithm first sets $q_{\mathcal{O}}^{c}$ to the unique (up to equivalence w.r.t. Σ) minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, i.e., $q_{\mathcal{O}}^{c} = \{(x) \mid \exists y_1.\mathbf{K}(\exists z_1.P_1(x, y_1) \wedge P_2(x, z_1))\}$.

To compute $\mathsf{EQLPerfRef}_{q^c_{\mathcal{O}},\Sigma}$, the query $q^{\varrho}_{\mathcal{O}} = \{(x, y_1) \mid \exists z_1.P_1(x, y_1) \wedge P_2(x, z_1)\}$ associated to the epistemic atom $\mathbf{K}\varrho$ of $q^c_{\mathcal{O}}$ (with $\varrho = \exists z_1.P_1(x, y_1) \wedge P_2(x, z_1)$) is first reformulated with respect to $\mathcal{O}$, thus obtaining the set $\{q^1_{\mathcal{O}}, q^2_{\mathcal{O}}\}$, where $q^1_{\mathcal{O}} = q^{\varrho}_{\mathcal{O}}$, while $q^2_{\mathcal{O}} = \{(x, y_1) \mid P_1(x, y_1) \wedge A_2(x)\}$ is obtained from $q^1_{\mathcal{O}}$ by applying the assertion $A_2 \sqsubseteq \exists P_2$. Observe that the $A_1 \sqsubseteq \exists P_1$ ontology assertion is applied neither in $q^1_{\mathcal{O}}$ nor in $q^2_{\mathcal{O}}$, since in both queries the variables y_1 is a distinguished variable. After that, each query in the set $\{q^1_{\mathcal{O}}, q^2_{\mathcal{O}}\}$ is unfolded with respect to $\mathcal{M}$, and therefore the computed query $\mathsf{EQLPerfRef}_{q^c_{\mathcal{O}},\Sigma}$ over $\mathcal{S}$, which is then assumed to be turned into an equivalent UCQ, corresponds the union of the following CQs over $\mathcal{S}$:

- $q^1_{\mathcal{S}} = \{(x) \mid \exists y_1, y^1_2.s_1(x, y_1) \wedge s_2(x, y^1_2)\}$, obtained from $q^1_{\mathcal{O}}$ by unfolding the atom $P_1(x, y_1)$ through m_1, and the atom $P_2(y_1, z_1)$ through m_2;
- $q^2_{\mathcal{S}} = \{(x) \mid \exists y_1, y^2_2, y^2_3.s_1(x, y_1) \wedge s_3(x, y^2_2, y^2_3)\}$, obtained from $q^1_{\mathcal{O}}$ by unfolding the atom $P_1(x, y_1)$ through m_1, and the atom $P_2(y_1, z_1)$ through m_3 (after renaming variable z_1 with y^2_2);
- $q^3_{\mathcal{S}} = \{(x) \mid \exists y_1, y^3_2, y^3_3.s_1(x, y_1) \wedge s_4(x, y^3_2, y^3_3)\}$, obtained from $q^2_{\mathcal{O}}$ by unfolding the atom $P_1(x, y_1)$ through m_1, and the atom $A_2(y_1)$ through m_4.

Observe that the m_6 mapping assertion is applied neither in $q^1_{\mathcal{O}}$ nor in $q^2_{\mathcal{O}}$ for unfolding the atom $P_1(x, y_1)$, since in both queries the variable y_1 is a distinguished variable, whereas the corresponding argument (with same predicate name P_1) of the atom in the right-hand side of m_6 is only an existential variable.

While $q^1_{\mathcal{S}} \sqsubseteq q_{\mathcal{S}}$, it is easy to see that $q^i_{\mathcal{S}} \not\sqsubseteq q_{\mathcal{S}}$ for both $i = 2$ and $i = 3$. Thus, the *EQL-Lite*$^{-}$(UCQ) query returned by the algorithm is:

$$
\begin{aligned}
q_{\mathcal{O}} = \{(x) \mid \exists y_1.\mathbf{K}(\exists z_1.P_1(x, y_1) \wedge P_2(x, z_1)) \wedge \\
\neg\exists y^2_2.\mathbf{K}(P_1(x, y_1) \wedge P_2(x, y^2_2)) \wedge \\
\neg\exists y^3_3.\mathbf{K}(\exists z_2.P_1(x, y_1) \wedge A_2(x) \wedge P_3(y^3_3, z_2))\}.
\end{aligned}
$$

Note that for the $\mathcal{S}$-database $D = \{s_1(c, c_1), s_2(c_1, c_2)\}$ we have both $(c_1) \in q^D_{\mathcal{S}}$ and $(c_1) \in cert^D_{q_{\mathcal{O}},\Sigma}$. Conversely, consider the $\mathcal{S}$-databases $D_1 = D \cup \{s_3(c_1, c_3, c_4)\}$ and $D_2 = D \cup \{s_4(c_1, c_5, c_6)\}$. Then, we have $(c) \in q^{D_i}_{\mathcal{S}}$ but $(c) \notin cert^{D_i}_{q_{\mathcal{O}},\Sigma}$, for both $i = 1$ and $i = 2$. Thus, $q_{\mathcal{O}}$ is not a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$. □

We are now ready to establish termination and correctness of the MaximallySoundEpistemic algorithm.

Theorem 9.7. *Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification where $\mathcal{O}$ is a DL-Lite$^{-}_{\mathcal{R}}$ ontology and $\mathcal{M}$ is a One-To-One mapping, and let $q_{\mathcal{S}}$ be a CQJFE over $\mathcal{S}$. We have that MaximallySoundEpistemic$(\Sigma, q_{\mathcal{S}})$ terminates and returns the unique (up to equivalence w.r.t. Σ) maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of EQL-Lite(UCQ) queries.*

Proof. Termination of the algorithm follows from the termination of the chase of a source instance (possibly containing variables) with respect to a One-To-One mapping, and the fact that it is always possible to compute $\mathsf{EQLPerfRef}_{q^c_{\mathcal{O}},\Sigma}$ for an OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and an *EQL-Lite*(UCQ) query $q^c_{\mathcal{O}}$ over $\mathcal{O}$.

Let $q_\mathcal{O}$ be the *EQL-Lite*$^-$(UCQ) query returned by the algorithm. We now divide the proof into two parts: we first prove that $q_\mathcal{O}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$, and then we prove that each *EQL-Lite*(UCQ) $q'_\mathcal{O}$ that is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$ is such that $cert_{q'_\mathcal{O},\Sigma} \sqsubseteq cert_{q_\mathcal{O},\Sigma}$.

Lemma 9.5. *$q_\mathcal{O}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$.*

Proof. If the algorithm returns the query $q_\mathcal{O} = \{\vec{t} \mid \bot(\vec{x})\}$, then the claim is trivial. Otherwise, let D be any $\mathcal{S}$-database, and let $\vec{c}$ be any tuple of constants for which $\vec{c} \in cert^D_{q_\mathcal{O},\Sigma}$. We now prove that $\vec{c} \in q_\mathcal{S}^D$ as well. By construction of $q_\mathcal{O}$, it follows that $\vec{c} \in cert^D_{q^c_\mathcal{O},\Sigma}$, where $q^c_\mathcal{O} = \{\vec{t} \mid \exists\vec{y}.\mathbf{K}(\exists\vec{z}.\mathcal{M}(q_\mathcal{S}))\}$ is the query returned by $\mathsf{MinimallyCompleteEpistemic}(\Sigma, q_\mathcal{S})$, i.e., the unique (up to equivalence w.r.t. Σ) minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$ in the class of *EQL-Lite*(UCQ) queries (cf. Theorem 9.6). Since $\vec{c} \in cert^D_{q^c_\mathcal{O},\Sigma}$, by combining Theorem 9.1 with Proposition 9.3, we derive that there is at least a disjunct $q^i_\mathcal{S}$ of the UCQ $\mathsf{EQLPerfRef}_{q^c_\mathcal{O},\Sigma}$ witnessing that $\vec{c} \in cert^D_{q^c_\mathcal{O},\Sigma}$, i.e., a disjunct $q^i_\mathcal{S}$ for which $\vec{c} \in {q^i_\mathcal{S}}^D$. There are two possible cases for the disjunct $q^i_\mathcal{S}$: either $q^i_\mathcal{S} \not\sqsubseteq q_\mathcal{S}$, or $q^i_\mathcal{S} \sqsubseteq q_\mathcal{S}$.

In the former case, by construction of the algorithm, the query $q_\mathcal{O}$ contains in its body the formula $\neg\exists\vec{y}_i.\mathbf{K}(\exists\vec{z}_i.\mathcal{M}(q^i_\mathcal{S}))$ in conjunction to the body of $q^c_\mathcal{O}$. Furthermore, observe that $q^i_\mathcal{S}$ is a disjunct of $\mathsf{EQLPerfRef}_{q^c_\mathcal{O},\Sigma}$, and therefore each distinguished variable occurring in $q^i_\mathcal{S}$ occurs also in $\mathcal{M}(q^i_\mathcal{S})$. So, by Theorem 9.6, we derive that $\exists\vec{y}_i.\mathbf{K}(\exists\vec{z}_i.\mathcal{M}(q^i_\mathcal{S}))$ is the body of a query being a complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q^i_\mathcal{S}$ (in fact, the unique (up to equivalence w.r.t. Σ) minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q^i_\mathcal{S}$ in the class of *EQL-Lite*(UCQ) queries). Thus, since $\vec{c} \in {q^i_\mathcal{S}}^D$, from the above observation we have that formula $\exists\vec{y}_i.\mathbf{K}(\exists\vec{z}_i.\mathcal{M}(q^i_\mathcal{S}))$ is *EQL*-logically implied by $\langle\Sigma, D\rangle$ (and thus, formula $\neg\exists\vec{y}_i.\mathbf{K}(\exists\vec{z}_i.\mathcal{M}(q^i_\mathcal{S}))$ is *not* *EQL*-logically implied by $\langle\Sigma, D\rangle$) when replacing its free variables with the tuple of constants $\vec{c}$. As a consequence, we have that $\vec{c} \notin cert^D_{q_\mathcal{O},\Sigma}$, which is a contradiction to the initial assumption that $\vec{c} \in cert^D_{q_\mathcal{O},\Sigma}$. It follows that the former case just considered is not possible because it leads to a contradiction. Therefore, we consider only the latter case. But then, as for the latter case, observe that $\vec{c} \in {q^i_\mathcal{S}}^D$ and $q^i_\mathcal{S} \sqsubseteq q_\mathcal{S}$ clearly imply that $\vec{c} \in q^D_\mathcal{S}$ as well, as required. □

We now show that $q_\mathcal{O}$ is actually the unique (up to equivalence w.r.t. Σ) maximally sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$ in the class of *EQL-Lite*(UCQ) queries. Since from the above lemma we know that $q_\mathcal{O}$ is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$, it is enough to prove that each *EQL-Lite*(UCQ) $q'_\mathcal{O}$ that is a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$ is such that $cert_{q'_\mathcal{O},\Sigma} \sqsubseteq cert_{q_\mathcal{O},\Sigma}$ (cf. Definition 3.6). We do this by way of contradiction.

Let $q'_\mathcal{O}$ be an *EQL-Lite*(UCQ) query such that $cert_{q'_\mathcal{O},\Sigma} \not\sqsubseteq cert_{q_\mathcal{O},\Sigma}$, that is, there exists an $\mathcal{S}$-database D consistent with Σ such that $cert^D_{q'_\mathcal{O},\Sigma} \not\subseteq cert^D_{q_\mathcal{O},\Sigma}$. It follows that there is a tuple of constants $\vec{c}$ such that $\vec{c} \in cert^D_{q'_\mathcal{O},\Sigma}$, but $\vec{c} \notin cert^D_{q_\mathcal{O},\Sigma}$. If $\vec{c} \notin q^D_\mathcal{S}$, then $q'_\mathcal{O}$ is trivially not a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_\mathcal{S}$, and we are done. Therefore, we assume that $\vec{c} \in q^D_\mathcal{S}$. We now exhibit an $\mathcal{S}$-database D' consistent with Σ for which (*i*) $\vec{c} \notin q^{D'}_\mathcal{S}$, and (*ii*) $\mathcal{C}^{\mathcal{M}(D')}_\mathcal{O}$ and $\mathcal{C}^{\mathcal{M}(D)}_\mathcal{O}$ are *homomorphically equivalent*, i.e., there is a homomorphism from $\mathcal{C}^{\mathcal{M}(D')}_\mathcal{O}$ to $\mathcal{C}^{\mathcal{M}(D)}_\mathcal{O}$ and vice versa.

Consider first the case that the algorithm returns the query $q_{\mathcal{O}} = \{\vec{t} \mid \bot(\vec{x})\}$. In this case, in the body of $q_{\mathcal{S}}$ there is at least a distinguished variable $x \in \vec{x}$ occurring as k-th argument of some source predicate $s \in \mathcal{S}$ such that x does not occur in $\mathcal{M}(q_{\mathcal{S}})$. Let h be any homomorphism from $q_{\mathcal{S}} = \{\vec{t} \mid \exists \vec{y}. \phi(\vec{x}, \vec{y})\}$ to D with $h(\vec{t}) = \vec{c}$, and consider the constant $c \in \vec{c}$ for which $h(x) = c$. The $\mathcal{S}$-database D' we are seeking is obtained from D by replacing each fact of the form $s(\vec{a})$ with $s(\vec{a'})$, where $\vec{a}$ is any tuple of constants in which the k-th argument is $h(x) = c$, and $\vec{a'}$ is obtained from $\vec{a}$ by replacing the k-th argument $h(x) = c$ with a fresh constant c_x. Two considerations follow for the $\mathcal{S}$-database D': (i) Clearly, we have that $\vec{c} \notin q_{\mathcal{S}}^{D'}$ because there can be no homomorphism h from $q_{\mathcal{S}}$ to D' with $h(x) = c$ (and thus, with $h(\vec{t}) = \vec{c}$); (ii) Since x does not occur in $\mathcal{M}(q_{\mathcal{S}})$, and since the left-hand side of One-To-One mapping assertions are simply atoms without constants or repeated variables, it is easy to see that D' is such that $\mathcal{M}(D) = \mathcal{M}(D')$. It follows that D' is such that (i) $\vec{c} \notin q_{\mathcal{S}}^{D'}$, and (ii) $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D')}$ and $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ are homomorphically equivalent.

We now consider the case that the algorithm does not return the query $q_{\mathcal{O}} = \{\vec{t} \mid \bot(\vec{x})\}$, i.e., each distinguished variable of $q_{\mathcal{S}}$ occurs in $\mathcal{M}(q_{\mathcal{S}})$. Consider any homomorphism h from $q_{\mathcal{S}} = \{\vec{t} \mid \exists \vec{y}. \phi(\vec{x}, \vec{y})\}$ to D with $h(\vec{t}) = \vec{c}$, and consider $h(q_{\mathcal{S}})$, i.e., the set of facts corresponding to the image of h under $q_{\mathcal{S}}$. By construction (see also Theorem 9.6), the formula $\exists \vec{y}. \mathbf{K}(\exists \vec{z}. \mathcal{M}(q_{\mathcal{S}}))$ occurring in conjunction in the body of $q_{\mathcal{O}}$ is true in $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ when replacing each free variable $x \in \vec{x}$ and each variable $y \in \vec{y}$ with $h(x)$ and $h(y)$, respectively (observe that the existential variables $\vec{y}$ are those of $q_{\mathcal{S}}$ occurring also in $\mathcal{M}(q_{\mathcal{S}})$). Since, however, $\vec{c} \notin cert_{q_{\mathcal{O}},\Sigma}^{D}$, by construction of $q_{\mathcal{O}}$ we derive that there is at least a formula $\neg \exists \vec{y_i}. \mathbf{K}(\exists \vec{z_i}. \mathcal{M}(q_{\mathcal{S}}^i))$ occurring in conjunction in the body of $q_{\mathcal{O}}$ that is *not* *EQL*-logically implied by $\langle \Sigma, D \rangle$ when replacing each free variable $x \in \vec{x}$ and each variable $y \in \vec{y}$ with $h(x)$ and $h(y)$, respectively, where $q_{\mathcal{S}}^i = \{\vec{t} \mid \exists \vec{y_i}. \phi_i(\vec{x}, \vec{y_i})\}$ is a disjunct of $\mathsf{EQLPerfRef}_{q_{\mathcal{O}}^c,\Sigma}$ for which $q_{\mathcal{S}}^i \not\sqsubseteq q_{\mathcal{S}}$ with $q_{\mathcal{O}}^c = \{\vec{t} \mid \exists \vec{y}. \mathbf{K}(\exists \vec{z}. \mathcal{M}(q_{\mathcal{S}}))\}$. So, formula $\exists \vec{y_i}. \mathbf{K}(\exists \vec{z_i}. \mathcal{M}(q_{\mathcal{S}}^i))$ must be true in $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ when replacing each free variable $x \in \vec{x}$ and each variable $y \in \vec{y}$ with $h(x)$ and $h(y)$, respectively. Therefore, using Proposition 9.4, this implies that it is possible to extend h with a new homomorphism h_i from terms of $\mathcal{M}(q_{\mathcal{S}}^i)$ to $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ for which $h_i(y_i)$ is a constant, for each $y_i \in \vec{y_i}$. Let now h' be the function extending h_i by assigning a different fresh constant c_{y_i} to each existential variable $y_i \notin \vec{y_i} \cup \vec{y}$ of $q_{\mathcal{S}}^i$, i.e., to each existential variable of $q_{\mathcal{S}}^i$ not occurring in $\mathcal{M}(q_{\mathcal{S}}^i)$.

Notice that $q_{\mathcal{S}}^i$ is a disjunct of $\mathsf{EQLPerfRef}_{q_{\mathcal{O}}^c,\Sigma}$ with $q_{\mathcal{O}}^c = \{\vec{t} \mid \exists \vec{y}. \mathbf{K}(\exists \vec{z}. \mathcal{M}(q_{\mathcal{S}}))\}$. Thus, since the left-hand side of each mapping assertion in $\mathcal{M}$ is simply an atom without constants or repeated variables, it is easy to verify that all the possible logical consequences over $\mathcal{O}$ of the set of facts $h(q_{\mathcal{S}})$ is a subset of the logical consequences over $\mathcal{O}$ of the set of facts $h'(q_{\mathcal{S}}^i)$, i.e., there is a homomorphism from $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(h(q_{\mathcal{S}}))}$ to $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(h'(q_{\mathcal{S}}^i))}$. Furthermore, similarly as already observed in the proof of Theorem 9.6, since formula $\exists \vec{y_i}. \mathbf{K}(\exists \vec{z_i}. \mathcal{M}(q_{\mathcal{S}}^i))$ is true in $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ when replacing each free variable $x \in \vec{x}$, each variable $y \in \vec{y}$, and each variable $y_i \in \vec{y_i}$ with $h(x) = h_i(x)$, $h(y) = h_i(y)$, and $h_i(y_i)$, respectively, and since the left-hand side of each mapping assertion in $\mathcal{M}$ is a single atom without constants or repeated variables, we derive that each fact $\alpha \in h'(q_{\mathcal{S}}^i)$ is such that there is a homomorphism from $\mathcal{M}(\alpha)$ to $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$. So, using Proposition 9.5, there is a homomorphism from $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(h'(q_{\mathcal{S}}^i))}$ to $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$.

Due to the fact that there is a homomorphism from $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(h(q_{\mathcal{S}}))}$ to $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(h'(q^i_{\mathcal{S}}))}$, and the fact that there is a homomorphism from $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(h'(q^i_{\mathcal{S}}))}$ to $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$, by making use of Proposition 9.5, it is possible to conclude that the $\mathcal{S}$-database D_h obtained from D by removing all the facts in $h(q_{\mathcal{S}})$ and adding all the facts in $h'(q^i_{\mathcal{S}})$ is such that $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D_h)}$ and $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ are homomorphically equivalent, i.e., $D_h = ((D \setminus h(q_{\mathcal{S}})) \cup h'(q^i_{\mathcal{S}}))$ is such that $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D_h)}$ and $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ are homomorphically equivalent.

Consider now the $\mathcal{S}$-database D' obtained by repeatedly iterating the above process for each possible homomorphism h from $q_{\mathcal{S}}$ to D with $h(\vec{t}) = \vec{c}$, until one obtain a D' such that $\vec{c} \notin q_{\mathcal{S}}^{D'}$. In other words, the $\mathcal{S}$-database D' can be obtained starting from D and then repeatedly removing all the facts in $h(q_{\mathcal{S}})$ and adding all the facts in $h'(q^i_{\mathcal{S}})$, for each homomorphism h from $q_{\mathcal{S}}$ to D with $h(\vec{t}) = \vec{c}$, where $q^i_{\mathcal{S}}$ is a disjunct of $\mathsf{EQLPerfRef}_{q^c_{\mathcal{O}},\Sigma}$ for which (i) $q^i_{\mathcal{S}} \not\sqsubseteq q_{\mathcal{S}}$ and formula $\exists \vec{y}_i.\mathbf{K}(\exists \vec{z}_i.\mathcal{M}(q^i_{\mathcal{S}}))$ is true in $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ when replacing each free variable $x \in \vec{x}$ and each variable $y \in \vec{y}$ with $h(x)$ and $h(y)$, respectively (at least one must exists because $\vec{c} \notin cert^D_{q_{\mathcal{O}},\Sigma}$), and (ii) h' is obtained from h and $q^i_{\mathcal{S}}$ as illustrated above. Using again Proposition 9.5, from the previous observations, we derive that the obtained $\mathcal{S}$-database D' is such that $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D')}$ and $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ are homomorphically equivalent. Furthermore, since $q_{\mathcal{S}}$ is a CQJFE, and therefore it does not have existential variables in join occurring in its body, and since in D' we have removed all the facts $h(q_{\mathcal{S}})$ for each homomorphism h from $q_{\mathcal{S}}$ to D with $h(\vec{t}) = \vec{c}$, one can easily verify that $\vec{c} \notin q_{\mathcal{S}}^{D'}$ by construction.

To conclude the proof note that, both in the case that the algorithm returns the query $\{\vec{t} \mid \bot(\vec{x})\}$ and in the case that it does not return $\{\vec{t} \mid \bot(\vec{x})\}$, it is possible to obtain an $\mathcal{S}$-database D' such that (i) $\vec{c} \notin q_{\mathcal{S}}^{D'}$, and (ii) $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D')}$ and $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ are homomorphically equivalent. Since $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D')}$ and $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ are homomorphically equivalent, and since $\vec{c} \in cert^D_{q'_{\mathcal{O}},\Sigma}$ by assumption, it is easy to see that $\vec{c} \in cert^{D'}_{q'_{\mathcal{O}},\Sigma}$ as well. It follows that D' is an $\mathcal{S}$-database consistent with Σ (observe that $\mathcal{O}$ is a $DL\text{-}Lite^{-}_{\mathcal{R}}$ ontology, and therefore each $\mathcal{S}$-database is consistent with Σ) for which $\vec{c} \notin q_{\mathcal{S}}^{D'}$ and $\vec{c} \in cert^{D'}_{q'_{\mathcal{O}},\Sigma}$, thus implying that $q'_{\mathcal{O}}$ is not a sound $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, as required. □

As for the running time of the $\mathsf{MaximallySoundEpistemic}$ algorithm, we observe that it is independent of the size of $\mathcal{S}$, polynomial in the size of both $\mathcal{O}$ and $\mathcal{M}$, and exponential in the size of $q_{\mathcal{S}}$. This latter is due to the fact that $\mathsf{EQLPerfRef}_{q^c_{\mathcal{O}},\Sigma}$ is in general the union of an exponential number of CQs with respect to the number of atoms occurring in $q_{\mathcal{S}}$, and also due to the various containment check of CQs. Finally, note that the overall running time is exponential in the size of the input.

9.4.3 Perfect Source-to-Ontology Rewritings

We conclude this chapter with a consideration on perfect source-to-ontology rewritings in the class of *EQL-Lite*(UCQ) queries, for the case of OBDM specifications with One-To-One mapping assertions.

Consider any pair composed by an OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ where $\mathcal{O}$ is a $DL\text{-}Lite^{-}_{\mathcal{R}}$ ontology and $\mathcal{M}$ is a One-to-One mapping and a CQ $q_{\mathcal{S}}$ over $\mathcal{S}$. Clearly, by definition, either the unique (up to equivalence w.r.t. Σ) minimally complete $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of *EQL-Lite*(UCQ) queries is also a sound, and therefore a perfect, $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$, or no perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of *EQL-Lite*(UCQ) queries exists.

With this observation at hand, and making use of Theorem 9.6, we can specialise the PerfectEpistemic algorithm in an obvious way for the case of OBDM specifications with One-to-One mapping assertions. For the sake of completeness, we report here the algorithm PerfectEpistemicOneToOne together with its termination and correctness.

Algorithm 9.4 PerfectEpistemicOneToOne

Input:

OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ where $\mathcal{O}$ is a *DL-Lite*$_{\mathcal{R}}^{-}$ ontology and $\mathcal{M}$ is a One-to-One mapping;
CQ $q_{\mathcal{S}}$ over $\mathcal{S}$

Output:

either an *EQL-Lite*$^{-}$(UCQ) $q_{\mathcal{O}}$ over $\mathcal{O}$, or report that "no perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of *EQL-Lite*(UCQ) queries exists"

$q_{\mathcal{O}}$:= MinimallyCompleteEpistemic$(\Sigma, q_{\mathcal{S}})$
if EQLPerfRef$_{q_{\mathcal{O}},\Sigma} \sqsubseteq q_{\mathcal{S}}$ **then**
  return $q_{\mathcal{O}}$
else
  return "no perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of *EQL-Lite*(UCQ) queries exists"
end if

Theorem 9.8. *Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification where $\mathcal{O}$ is a DL-Lite$_{\mathcal{R}}^{-}$ ontology and $\mathcal{M}$ is a One-To-One mapping, and let $q_{\mathcal{S}}$ be a CQ over $\mathcal{S}$. We have that PerfectEpistemicOneToOne$(\Sigma, q_{\mathcal{S}})$ terminates and returns the perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ if it exists and can be expressed as an EQL-Lite(UCQ) query, otherwise it reports that no perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of EQL-Lite(UCQ) queries exists.*

Furthermore, as a straightforward consequence of Corollary 9.2, we also get the following interesting result.

Corollary 9.3. *Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification where $\mathcal{O}$ is a DL-Lite$_{\mathcal{R}}^{-}$ ontology and $\mathcal{M}$ is a One-To-One mapping, and let $q_{\mathcal{S}}$ be a CQ over $\mathcal{S}$. We have that the perfect $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of $q_{\mathcal{S}}$ in the class of EQL-Lite(UCQ) queries either does not exists, or it can be expressed as an EQL-Lite^{-}(UCQ) query.*

Chapter 10

Conclusions

This chapter concludes the thesis with a brief discussion and possible directions for future work.

10.1 Discussion

In this thesis we have introduced a novel reasoning task over the OBDM specification, called abstraction. The main purpose of this task is to automatically produce the semantic characterisation of data services through ontologies, and thus make data services automatically Findable, Accessible, Interoperable, and Reusable (FAIR). We have presented a formal framework for abstraction, via the semantically well-founded notion(s) of source-to-ontology rewriting, which can be seen as the inverse of the well-known and well-studied notion(s) of ontology-to-source rewriting. We have carried out a comprehensive analysis of two important related computational problems within the most common languages used in OBDM, including two restricted scenarios and also the addition of non-monotonicity in the target query language.

We believe that the notions introduced and the technical results presented in this thesis are not only theoretically interesting in themselves, but also have many possible practical applications besides the semantic characterisations of data services, as for example in the fields mentioned in the introduction, namely open data, source profiling, updating, and explanation of classifiers. We point out that the thesis left some interesting and challenging open problems, which are detailed in the below list:

- When the UNA is not adopted, answering $\mathrm{CQ}^{\neq,b}$s over $\mathit{DL\text{-}Lite}_{\mathcal{R}}$ knowledge bases has been shown to be FOL-rewritable. The FOL-rewritability question, however, is still open for unions thereof, although it has been proven that the problem remains in AC^0 in data complexity.

- When the UNA is not adopted, answering $\mathrm{UCQ}^{2,\neq}$s over $\mathit{DL\text{-}Lite}_{\mathsf{RDFS}}$ knowledge bases as well as answering general $\mathrm{CQ}^{\neq}$s over $\mathit{DL\text{-}Lite}^{\neg}_{\mathsf{RDFS}}$ knowledge bases has been shown to be Π^p_2-complete in combined complexity. For the case of $\mathrm{CQ}^{2,\neq}$s over $\mathit{DL\text{-}Lite}^{\neg}_{\mathsf{RDFS}}$ knowledge bases, however, the Π^p_2-hardness part has only been conjectured.

- Strictly related to the above problem is the containment problem for $\mathrm{UCQ}^{\neq}$s. Specifically, while checking whether $q' \sqsubseteq q$ has been shown to be Π^p_2-hard (and therefore Π_2^p-complete) when both q' and q are $\mathrm{CQ}^{\neq}$s as well as when q' is a CQ and q is a $\mathrm{UCQ}^{2,\neq}$, the Π_2^p-hardness of the problem has only been conjectured for the case of q' being a $\mathrm{CQ}^{\neq}$ and q being a $\mathrm{CQ}^{2,\neq}$.

- For the case of mappings that are both pure GAV and LAV, the exact computational complexity of the verification problem for sound source-to-ontology rewritings is still open.

- For the case of LAV mappings, the exact computational complexity of the verification problem for perfect source-to-ontology rewritings is still open.

- Both in the restricted scenario for UCQJFEs and in the restricted scenario for CQJFEs, we point out that the verification problem has been studied for sound source-to-ontology rewritings only. While the proof of Theorem 5.1 already shows that the verification problem for complete source-to-ontology rewritings is NP-complete in both the restricted scenarios, the problem is still open for perfect source-to-ontology rewritings. In particular, it is trivial to establish membership in DP and in NP in the restricted scenario for UCQJFEs and in the restricted scenario for CQJFEs, respectively (for the former, just observe that the set of perfect source-to-ontology rewritings is the intersection between the set of sound source-to-ontology rewritings and the set of complete source-to-ontology rewritings, where, in the restricted scenario for UCQJFEs, soundness and completeness can be verified in coNP and in NP, respectively). However, we only conjecture matching lower bounds for both cases.

- The verification problem for all types of source-to-ontology rewritings when the target query language is the class of *EQL-Lite*(UCQ) queries has not been addressed at all, and thus it is still an open problem to determine the exact computational complexity of the verification problem for each type of source-to-ontology rewriting.

- The proofs of the non-existence cases of minimally complete, and maximally sound source-to-ontology rewritings in the class of *EQL-Lite*(UCQ) queries (Theorems 9.2 and 9.3, respectively) rely on pure GAV mapping assertions. For the case of LAV mappings, however, it is still an open problem to determine whether minimally complete (respectively, maximally sound) source-to-ontology rewritings are guaranteed to exist in the class of *EQL-Lite*(UCQ) queries.

- In addition to the previous cases, we point out that for the case of One-To-One mappings it is still an open problem to determine whether maximally sound source-to-ontology rewritings of CQs in the class of *EQL-Lite*(UCQ) queries are guaranteed to exist.

10.2 Future Work

In addition to tackling the open problems mentioned in the foregoing list, we see many other interesting avenues for future work, including the following:

- Basically, the introduced notions of source-to-ontology rewriting are based on certain answers. In principle, there may be other meaningful properties that a query over the ontology have to satisfy to be considered a perfect (respectively, sound, complete) source-to-ontology rewriting. For instance, one may consider a *model-based semantics* where a query $q_{\mathcal{O}}$ is a perfect (respectively, sound, complete) $\mathcal{S}$-to-$\mathcal{O}$ Σ-rewriting of a query $q_{\mathcal{S}}$ if and only if $q_{\mathcal{S}}^{D} = q_{\mathcal{O}}^{\mathcal{I}}$ (respectively, $q_{\mathcal{O}}^{\mathcal{I}} \subseteq q_{\mathcal{S}}^{D}$, $q_{\mathcal{S}}^{D} \subseteq q_{\mathcal{O}}^{\mathcal{I}}$) for each $\mathcal{S}$-database D and model $\mathcal{I} \in Mod_{\Sigma}(D)$.

- Throughout the thesis, we have implicitly assumed that both the evaluation of queries posed over source databases and the certain answers of queries posed over the ontology of OBDM systems are sets. This is in contrast with the standard semantics of DBMSs, which is based on *bags* (i.e., multisets) and duplicate tuples are retained by default. Considering as starting point the works [Nikolaou *et al.*, 2019; Cima *et al.*, 2019e] that propose a bag semantics for OBDM systems, it would be interesting to investigate the various notions of source-to-ontology rewriting under a bag-based semantics.

- Study the impact of integrity constraints over the source schemas $\mathcal{S}$.

- Extending the analysis to OBDM settings going beyond the one based on *DL-Lite*$_{\mathcal{R}}$, e.g., by considering as ontology languages DLs equipped with role functionality assertions such as *DL-Lite*$_{\mathcal{A}}$, or DLs of the $\mathcal{EL}$ family.

- The class of queries for expressing data services in this thesis has been the one of UCQs (and its fragments). It would be interesting to examine cases where the source query language for expressing data services goes beyond UCQs.

- In Chapter 6, we have seen many cases where UCQ-maximally sound source-to-ontology rewritings are not guaranteed to exists in the general scenario. It would be very useful singling out the minimal class of queries $\mathcal{L}_{\mathcal{O}}$ that guarantees the existence of $\mathcal{L}_{\mathcal{O}}$-maximally sound source-to-ontology rewritings. For instance, one may start to investigate whether such $\mathcal{L}_{\mathcal{O}}$ is the class of (*unions of*) *conjunctive two-way regular path queries*, a class of queries well-studied in the context of lightweight DLs (see, e.g., [Bienvenu *et al.*, 2015]).

- Related to the above problem is to analyse another notable decision problem, namely the *existence problem for a target query language* $\mathcal{L}_{\mathcal{O}}$: check whether an $\mathcal{L}_{\mathcal{O}}$ source-to-ontology rewriting (perfect, or approximated) exists for a given OBDM specification $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and a given source query $q_{\mathcal{S}}$ over $\mathcal{S}$.

- Singling out more interesting scenarios in which source-to-ontology rewritings expressed in the class of *EQL-Lite*(UCQ) queries (or its fragments still non-monotonic) can be actually computed.

- Study the impact of our notions and technical results in other data interoperation architectures, such as peer-to-peer data integration [Calvanese *et al.*, 2004b].

We believe that each of the above issues is an interesting research problem that deserves to be investigated.

References

[Abedjan *et al.*, 2017] Ziawasch Abedjan, Lukasz Golab, and Felix Naumann. Data profiling: A tutorial. In *Proceedings of the 2017 ACM International Conference on Management of Data (SIGMOD 2017)*, pages 1747–1751, 2017.

[Abedjan *et al.*, 2018] Ziawasch Abedjan, Lukasz Golab, Felix Naumann, and Thorsten Papenbrock. *Data Profiling.* Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2018.

[Abiteboul and Duschka, 1998] Serge Abiteboul and Oliver M. Duschka. Complexity of answering queries using materialized views. In *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 1998)*, pages 254–265, 1998.

[Abiteboul *et al.*, 1995] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases.* Addison Wesley Publishing Company, 1995.

[Afrati and Chirkova, 2019] Foto N. Afrati and Rada Chirkova. *Answering Queries Using Views, Second Edition.* Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2019.

[Afrati *et al.*, 1999] Foto N. Afrati, Manolis Gergatsoulis, and Theodoros G. Kavalieros. Answering queries using materialized views with disjunctions. In *Proceedings of the Seventh International Conference on Database Theory (ICDT 1999)*, volume 1540 of *Lecture Notes in Computer Science,* `https://link.springer.com/bookseries/558`, pages 435–452, 1999.

[Aho *et al.*, 1979] Alfred V. Aho, Yehoshua Sagiv, and Jeffrey D. Ullman. Equivalences among relational expressions. *SIAM Journal on Computing*, 8(2):218–246, 1979.

[Antonioli *et al.*, 2014] Natalia Antonioli, Francesco Castanò, Spartaco Coletta, Stefano Grossi, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Emanuela Virardi, and Patrizia Castracane. Ontology-based data management for the italian public debt. In *Proceedings of the Eighth International Conference on Formal Ontology in Information Systems (FOIS 2014)*, volume 267 of *Frontiers in Artificial Intelligence and Applications,* `http://ebooks.iospress.nl/bookseries/frontiers-in-artificial-intelligence-and-applications`, pages 372–385, 2014.

[Arenas *et al.*, 2009] Marcelo Arenas, Jorge Pérez, and Cristian Riveros. The recovery of a schema mapping: Bringing exchanged data back. *ACM Transactions on Database Systems*, 34(4):22:1–22:48, 2009.

[Arenas *et al.*, 2010] Marcelo Arenas, Pablo Barceló, Leonid Libkin, and Filip Murlak. *Relational and XML Data Exchange*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2010.

[Arenas *et al.*, 2014] Marcelo Arenas, Pablo Barceló, Leonid Libkin, and Filip Murlak. *Foundations of Data Exchange*. Cambridge University Press, 2014.

[Arenas *et al.*, 2018] Marcelo Arenas, Georg Gottlob, and Andreas Pieris. Expressive languages for querying the semantic web. *ACM Transactions on Database Systems*, 43(3):13:1–13:45, 2018.

[Arora and Barak, 2009] Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009.

[Artale *et al.*, 2009] Alessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyaschev. The *DL-Lite* family and relations. *Journal of Artificial Intelligence Research*, 36:1–69, 2009.

[Asratian *et al.*, 1998] Armen S. Asratian, Tristan Denley, and Roland Haggkvist. *Bipartite Graphs and their Applications*. Cambridge University Press, 1998.

[Baader *et al.*, 2003] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.

[Baader *et al.*, 2005] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI 2005)*, pages 364–369, 2005.

[Baader *et al.*, 2017] Franz Baader, Ian Horrocks, Carsten Lutz, and Ulrike Sattler. *An Introduction to Description Logic*. Cambridge University Press, 2017.

[Beame *et al.*, 2014] Paul Beame, Paraschos Koutris, and Dan Suciu. Skew in parallel query processing. In *Proceedings of the Thirty-Third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS 2014)*, pages 212–223, 2014.

[Beeri and Vardi, 1984] Catriel Beeri and Moshe Y. Vardi. A proof procedure for data dependencies. *Journal of the ACM*, 31(4):718–741, 1984.

[Berners-Lee *et al.*, 2001] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.

[Bernstein, 1976] Philip A. Bernstein. Synthesizing third normal form relations from functional dependencies. *ACM Transactions on Database Systems*, 1(4):277–298, 1976.

[Bienvenu *et al.*, 2015] Meghyn Bienvenu, Magdalena Ortiz, and Mantas Simkus. Regular path queries in lightweight description logics: Complexity and algorithms. *Journal of Artificial Intelligence Research*, 53:315–374, 2015.

[Bienvenu, 2016] Meghyn Bienvenu. Ontology-mediated query answering: Harnessing knowledge to get more from data. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI 2016)*, pages 4058–4061, 2016.

[Bizer *et al.*, 2009] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data - the story so far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.

[Bondy and Murty, 2008] Adrian Bondy and M. Ram Murty. *Graph Theory*. Graduate Texts in Mathematics. Springer, 2008.

[Börger *et al.*, 1997] Egon Börger, Erich Grädel, and Yuri Gurevich. *The Classical Decision Problem*. Perspectives in Mathematical Logic. Springer, 1997.

[Brickley and Guha, 2014] Dan Brickley and R. V. Guha. RDF Schema 1.1. W3C Recommendation, World Wide Web Consortium, 2014. Available at `https://www.w3.org/TR/2014/REC-rdf-schema-20140225/`.

[Calì *et al.*, 2002] Andrea Calì, Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. On the expressive power of data integration systems. In *Proceedings of the Twenty-First International Conference on Conceptual Modeling (ER 2002)*, volume 2503 of *Lecture Notes in Computer Science,* `https://link.springer.com/bookseries/558`, pages 338–350, 2002.

[Calì *et al.*, 2013] Andrea Calì, Georg Gottlob, and Michael Kifer. Taming the infinite chase: Query answering under expressive relational constraints. *Journal of Artificial Intelligence Research*, 48:115–174, 2013.

[Calvanese and De Giacomo, 2005] Diego Calvanese and Giuseppe De Giacomo. Data integration: A logic-based perspective. *AI Magazine*, 26(1):59–70, 2005.

[Calvanese *et al.*, 2000] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Moshe Y. Vardi. What is view-based query rewriting? In *Proceedings of the Seventh International Workshop on Knowledge Representation meets Databases (KRDB 2000)*, volume 29 of *CEUR Electronic Workshop Proceedings,* `http://ceur-ws.org/`, pages 17–27, 2000.

[Calvanese *et al.*, 2004a] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, Riccardo Rosati, and Guido Vetere. DL-Lite: Practical reasoning for rich Dls. In *Proceedings of the 2004 International Workshop on Description Logics (DL 2004)*, volume 104 of *CEUR Electronic Workshop Proceedings,* `http://ceur-ws.org/`, 2004.

[Calvanese *et al.*, 2004b] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. Logical foundations of peer-to-peer data integration. In *Proceedings of the Twenty-Third ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 2004)*, pages 241–251, 2004.

[Calvanese *et al.*, 2005] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Dl-lite: Tractable description logics for ontologies. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI 2005)*, pages 602–607, 2005.

[Calvanese *et al.*, 2007a] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. EQL-lite: Effective first-order query processing in description logics. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pages 274–279, 2007.

[Calvanese *et al.*, 2007b] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *Journal of Automated Reasoning*, 39(3):385–429, 2007.

[Calvanese *et al.*, 2007c] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Moshe Y. Vardi. View-based query processing: On the relationship between rewriting, answering and losslessness. *Theoretical Computer Science*, 371(3):169–182, 2007.

[Calvanese *et al.*, 2009] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Mariano Rodríguez-Muro, and Riccardo Rosati. Ontologies and databases: The *DL-Lite* approach. In *Reasoning Web. Semantic Technologies for Information Systems, Fifth International Summer School, Tutorial Lectures (RW 2009)*, volume 5689 of *Lecture Notes in Computer Science,* `https://link.springer.com/bookseries/558`, pages 255–356. 2009.

[Calvanese *et al.*, 2011] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Mariano Rodriguez-Muro, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. The Mastro system for ontology-based data access. *Semantic Web Journal*, 2(1):43–53, 2011.

[Calvanese *et al.*, 2012] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Moshe Y. Vardi. Query processing under GLAV mappings for relational and graph databases. *Proceedings of the VLDB Endowment*, 6(2):61–72, 2012.

[Calvanese *et al.*, 2017] Diego Calvanese, Benjamin Cogrel, Sarah Komla-Ebri, Roman Kontchakov, Davide Lanti, Martin Rezk, Mariano Rodriguez-Muro, and Guohui Xiao. Ontop: Answering SPARQL queries over relational databases. *Semantic Web Journal*, 8(3):471–487, 2017.

[Carey *et al.*, 2012] Michael J. Carey, Nicola Onose, and Michalis Petropoulos. Data services. *Communications of the ACM*, 55(6):86–97, 2012.

[Chandra and Merlin, 1977] Ashok K. Chandra and Philip M. Merlin. Optimal implementation of conjunctive queries in relational data bases. In *Proceedings of the Ninth ACM Symposium on Theory of Computing (STOC 1977)*, pages 77–90, 1977.

[Chellas, 1980] Brian F. Chellas. *Modal Logic - An Introduction.* Cambridge University Press, 1980.

[Chen *et al.*, 2014] Min Chen, Shiwen Mao, Yin Zhang, and Victor C. M. Leung. *Big Data - Related Technologies, Challenges and Future Prospects.* Springer Briefs in Computer Science. Springer, 2014.

[Cima *et al.*, 2017a] Gianluca Cima, Giuseppe De Giacomo, Maurizio Lenzerini, and Antonella Poggi. On the SPARQL metamodeling semantics entailment regime for OWL 2 QL ontologies. In *Proceedings of the Seventh International Conference on Web Intelligence, Mining and Semantics (WIMS 2017)*, pages 10:1–10:6, 2017.

[Cima *et al.*, 2017b] Gianluca Cima, Giuseppe De Giacomo, Maurizio Lenzerini, and Antonella Poggi. Querying OWL 2 QL ontologies under the SPARQL metamodeling semantics entailment regime. In *Proceedings of the Twenty-Fifth Italian Symposium on Advanced Database Systems (SEBD 2017)*, volume 2037 of *CEUR Electronic Workshop Proceedings,* `http://ceur-ws.org/`, page 165, 2017.

[Cima *et al.*, 2017c] Gianluca Cima, Maurizio Lenzerini, and Antonella Poggi. Semantic technology for open data publishing. In *Proceedings of the Seventh International Conference on Web Intelligence, Mining and Semantics (WIMS 2017)*, page 1:1, 2017.

[Cima *et al.*, 2019a] Gianluca Cima, Federico Croce, Maurizio Lenzerini, Antonella Poggi, and Elian Toccacieli. On queries with inequalities in *DL-LiteR≠*. In *Proceedings of the Thirty-Second International Workshop on Description Logics (DL 2019)*, volume 2373 of *CEUR Electronic Workshop Proceedings,* `http://ceur-ws.org/`, 2019.

[Cima *et al.*, 2019b] Gianluca Cima, Maurizio Lenzerini, and Antonella Poggi. Exploiting ontologies for explaining data sources semantics. In *Proceedings of Discussion and Doctoral Consortium Papers of the Eighteenth International Conference of the Italian Association for Artificial Intelligence (DDC@AI*IA 2019)*, volume 2495 of *CEUR Electronic Workshop Proceedings,* `http://ceur-ws.org/`, pages 33–35, 2019.

[Cima *et al.*, 2019c] Gianluca Cima, Maurizio Lenzerini, and Antonella Poggi. Reverse engineering of data services. In *Proceedings of the Twenty-Seventh Italian Symposium on Advanced Database Systems (SEBD 2019)*, volume 2400 of *CEUR Electronic Workshop Proceedings,* `http://ceur-ws.org/`, 2019.

[Cima *et al.*, 2019d] Gianluca Cima, Maurizio Lenzerini, and Antonella Poggi. Semantic characterization of data services through ontologies. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI 2019)*, pages 1647–1653, 2019.

[Cima *et al.*, 2019e] Gianluca Cima, Charalampos Nikolaou, Egor V. Kostylev, Mark Kaminski, Bernardo Cuenca Grau, and Ian Horrocks. Bag semantics of dl-lite with functionality axioms. In *Proceedings of the Eighteenth International Semantic Web Conference (ISWC 2019)*, volume 11778 of *Lecture Notes in Computer Science,* `https://link.springer.com/bookseries/558`, pages 128–144, 2019.

[Cima *et al.*, 2019f] Gianluca Cima, Charalampos Nikolaou, Egor V. Kostylev, Mark Kaminski, Bernardo Cuenca Grau, and Ian Horrocks. Bagging the *DL-Lite* family further. In *Proceedings of the Thirty-Second International Workshop on Description Logics (DL 2019)*, volume 2373 of *CEUR Electronic Workshop Proceedings,* `http://ceur-ws.org/`, 2019.

[Cima *et al.*, 2020a] Gianluca Cima, Domenico Lembo, Lorenzo Marconi, Riccardo Rosati, and Domenico Fabio Savo. Controlled query evaluation in ontology-based data access. In *Proceedings of the Nineteenth International Semantic Web Conference (ISWC 2020)*, volume 12506 of *Lecture Notes in Computer Science,* `https://link.springer.com/bookseries/558`, pages 128–146, 2020.

[Cima *et al.*, 2020b] Gianluca Cima, Domenico Lembo, Riccardo Rosati, and Domenico Fabio Savo. Controlled query evaluation in description logics through instance indistinguishability. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI 2020)*, pages 1791–1797, 2020.

[Cima *et al.*, 2020c] Gianluca Cima, Domenico Lembo, Riccardo Rosati, and Domenico Fabio Savo. Controlled query evaluation in description logics through instance indistinguishability (extended abstract). In *Proceedings of the Thirty-Third International Workshop on Description Logics (DL 2020)*, volume 2663 of *CEUR Electronic Workshop Proceedings,* `http://ceur-ws.org/`, 2020.

[Cima *et al.*, 2020d] Gianluca Cima, Maurizio Lenzerini, and Antonella Poggi. Answering conjunctive queries with inequalities in *DL-Lite*$_{\mathscr{R}}$. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2020)*, pages 2782–2789, 2020.

[Cima *et al.*, 2020e] Gianluca Cima, Maurizio Lenzerini, and Antonella 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)*, pages 243–252, 2020.

[Cima, 2017] Gianluca Cima. Preliminary results on ontology-based open data publishing. In *Proceedings of the Thirtieth International Workshop on Description Logics (DL 2017)*, volume 1879 of *CEUR Electronic Workshop Proceedings,* `http://ceur-ws.org/`, 2017.

[Civili *et al.*, 2013] Cristina Civili, Marco Console, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Lorenzo Lepore, Riccardo Mancini, Antonella Poggi, Riccardo Rosati, Marco Ruzzi, Valerio Santarelli, and Domenico Fabio Savo. MASTRO STUDIO: managing ontology-based data access applications. *Proceedings of the VLDB Endowment*, 6(12):1314–1317, 2013.

[Codd, 1970] Edgar F. Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970.

[Codd, 1972] Edgar F. Codd. Relational completeness of data base sublanguages. *Research Report / RJ / IBM / San Jose, California*, RJ987, 1972.

[Cook, 1973] Stephen A. Cook. A hierarchy for nondeterministic time complexity. *Journal of Computer and System Sciences*, 7(4):343–353, 1973.

[Corona *et al.*, 2009] Claudio Corona, Marco Ruzzi, and Domenico Fabio Savo. Filling the gap between OWL 2 QL and quonto: Rowlkit. In *Proceedings of the Twenty-Second International Workshop on Description Logics (DL 2009)*, 2009.

[Croce *et al.*, 2020] Federico Croce, Gianluca Cima, Maurizio Lenzerini, and Tiziana Catarci. Ontology-based explanation of classifiers. In *Proceedings of the Workshops of the EDBT/ICDT 2020 Joint Conference*, volume 2578 of *CEUR Electronic Workshop Proceedings,* `http://ceur-ws.org/`, 2020.

[Cuenca Grau *et al.*, 2008] Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, Bijan Parsia, Peter F. Patel-Schneider, and Ulrike Sattler. OWL 2: The next step for OWL. *Journal of Web Semantics*, 6(4):309–322, 2008.

[Cuenca Grau, 2004] Bernardo Cuenca Grau. A possible simplification of the semantic web architecture. In *Proceedings of the Thirteenth International World Wide Web Conference (WWW 2004)*, pages 704–713, 2004.

[De Giacomo *et al.*, 2018] Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, and Riccardo Rosati. Using ontologies for semantic data integration. In *A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years*, pages 187–202. 2018.

[Doan *et al.*, 2012] AnHai Doan, Alon Y. Halevy, and Zachary G. Ives. *Principles of Data Integration.* Morgan Kaufmann, 2012.

[Duschka and Genesereth, 1997] Oliver M. Duschka and Michael R. Genesereth. Answering recursive queries using views. In *Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 1997)*, pages 109–116, 1997.

[Duschka and Genesereth, 1998] Oliver M. Duschka and Michael R. Genesereth. Query planning with disjunctive sources. In *Proceedings of the AAAI-98 Workshop on AI and Information Integration*, 1998.

[Duval *et al.*, 2002] Erik Duval, Wayne Hodgins, Stuart A. Sutton, and Stuart Weibel. Metadata principles and practicalities. *D-Lib Magazine*, 8(4), 2002.

[Eiter *et al.*, 1997] Thomas Eiter, Georg Gottlob, and Heikki Mannila. Disjunctive datalog. *ACM Transactions on Database Systems*, 22(3):364–418, 1997.

[European Union, Parliament and Council, 2016] European Union, Parliament and Council. General data protection regulation. *Official Journal of the European Union*, L 119/1, 2016.

[Fagin *et al.*, 2005a] Ronald Fagin, Phokion G. Kolaitis, Renée J. Miller, and Lucian Popa. Data exchange: Semantics and query answering. *Theoretical Computer Science*, 336(1):89–124, 2005.

[Fagin *et al.*, 2005b] Ronald Fagin, Phokion G. Kolaitis, and Lucian Popa. Data exchange: Getting to the core. *Journal of the ACM*, 30(1):174–210, 2005.

[Fagin *et al.*, 2005c] Ronald Fagin, Phokion G. Kolaitis, Lucian Popa, and Wang Chiew Tan. Composing schema mappings: Second-order dependencies to the rescue. *ACM Transactions on Database Systems*, 30(4):994–1055, 2005.

[Fagin *et al.*, 2008] Ronald Fagin, Phokion G. Kolaitis, Lucian Popa, and Wang Chiew Tan. Quasi-inverses of schema mappings. *ACM Transactions on Database Systems*, 33(2):11:1–11:52, 2008.

[Fagin, 1977] Ronald Fagin. Multivalued dependencies and a new normal form for relational databases. *ACM Transactions on Database Systems*, 2(3):262–278, 1977.

[Fagin, 2007] Ronald Fagin. Inverting schema mappings. *ACM Transactions on Database Systems*, 32(4):25, 2007.

[Friedman *et al.*, 1999] Marc Friedman, Alon Levy, and Todd Millstein. Navigational plans for data integration. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI 1999)*, pages 67–73, 1999.

[Garey and Johnson, 1979] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company, 1979.

[Garey *et al.*, 1976] Michael R. Garey, David S. Johnson, and Larry J. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237–267, 1976.

[Glimm and Ogbuji, 2013] Birte Glimm and Chimezie Ogbuji. SPARQL 1.1 entailment regimes. W3C Recommendation, World Wide Web Consortium, 2013. Available at `https://www.w3.org/TR/sparql11-entailment/`.

[Glimm, 2011] Birte Glimm. Using SPARQL with RDFS and OWL entailment. In *Reasoning Web. Semantic Technologies for the Web of Data - Seventh International Summer School, Tutorial Lectures (RW 2011)*, volume 6848 of *Lecture Notes in Computer Science,* `https://link.springer.com/bookseries/558`, pages 137–201, 2011.

[Gogacz and Marcinkowski, 2016] Tomasz Gogacz and Jerzy Marcinkowski. Red spider meets a rainworm: Conjunctive query finite determinacy is undecidable. In *Proceedings of the Thirty-Fifth ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS 2016)*, pages 121–134, 2016.

[Goodman and Flaxman, 2017] Bryce Goodman and Seth R. Flaxman. European union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine*, 38(3):50–57, 2017.

[Gottlob and Pieris, 2015] Georg Gottlob and Andreas Pieris. Beyond SPARQL under OWL 2 QL entailment regime: Rules to the rescue. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 2999–3007, 2015.

[Gottlob *et al.*, 2014] Georg Gottlob, Stanislav Kikot, Roman Kontchakov, Vladimir V. Podolskii, Thomas Schwentick, and Michael Zakharyaschev. The price of query rewriting in ontology-based data access. *Artificial Intelligence*, 213:42–59, 2014.

[Gruber, 1993] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[Gruber, 2018] Thomas R. Gruber. Ontology. In *Encyclopedia of Database Systems - Second Edition*. 2018.

[Gutiérrez-Basulto *et al.*, 2012] Víctor Gutiérrez-Basulto, Yazmin Angélica Ibáñez-García, and Roman Kontchakov. An update on query answering with restricted forms of negation. volume 7497 of *Lecture Notes in Computer Science,* `https://link.springer.com/bookseries/558`, pages 75–89, 2012.

[Gutiérrez-Basulto *et al.*, 2015] Víctor Gutiérrez-Basulto, Yazmin Angélica Ibáñez-García, Roman Kontchakov, and Egor V. Kostylev. Queries with negation and inequalities over lightweight ontologies. *Journal of Web Semantics*, 35:184–202, 2015.

[Harris and Seaborne, 2013] Steve Harris and Andy Seaborne. SPARQL 1.1 query language. W3C Recommendation, World Wide Web Consortium, 2013. Available at `http://www.w3.org/TR/sparql11-query`.

[Hartmanis and Stearns, 1965] Juris Hartmanis and Richard E. Stearns. On the computational complexity of algorithms. *Transactions of the American Mathematical Society*, 117:285–306, 1965.

[Imielinski and Lipski Jr., 1984] Tomasz Imielinski and Witold Lipski Jr. Incomplete information in relational databases. *Journal of the ACM*, 31(4):761–791, 1984.

[Karp, 1972] Richard M. Karp. Reducibility among combinatorial problems. In *Proceedings of a Symposium on the Complexity of Computer Computations*, The IBM Research Symposia Series, `https://link.springer.com/bookseries/10380`, pages 85–103, 1972.

[Ketsman and Suciu, 2017] Bas Ketsman and Dan Suciu. A worst-case optimal multi-round algorithm for parallel computation of conjunctive queries. In *Proceedings of the Thirty-Sixth ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS 2017)*, pages 417–428, 2017.

[Kharlamov *et al.*, 2017] Evgeny Kharlamov, Dag Hovland, Martin G. Skjæveland, Dimitris Bilidas, Ernesto Jiménez-Ruiz, Guohui Xiao, Ahmet Soylu, Davide Lanti, Martin Rezk, Dmitriy Zheleznyakov, Martin Giese, Hallstein Lie, Yannis E. Ioannidis, Yannis Kotidis, Manolis Koubarakis, and Arild Waaler. Ontology based data access in statoil. *Journal of Web Semantics*, 44:3–36, 2017.

[Klug, 1988] Anthony C. Klug. On conjunctive queries containing inequalities. *Journal of the ACM*, 35(1):146–160, 1988.

[Knopp, 1996] Konrad Knopp. *Theory of Functions, Parts I and II*. Dover Publications, 1996.

[Kolaitis and Vardi, 2000] Phokion G. Kolaitis and Moshe Y. Vardi. Conjunctive-query containment and constraint satisfaction. *Journal of Computer and System Sciences*, 61(2):302–332, 2000.

[Kolaitis and Vardi, 2008] Phokion G. Kolaitis and Moshe Y. Vardi. A logical approach to constraint satisfaction. In *Complexity of Constraints - An Overview of Current Research Themes*, volume 5250 of *Lecture Notes in Computer Science,* `https://link.springer.com/bookseries/558`, pages 125–155, 2008.

[Kolaitis *et al.*, 1998] Phokion G. Kolaitis, David L. Martin, and Madhukar N. Thakur. On the complexity of the containment problem for conjunctive queries with built-in predicates. In *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 1998)*, pages 197–204, 1998.

[Kontchakov *et al.*, 2014] Roman Kontchakov, Martin Rezk, Mariano Rodriguez-Muro, Guohui Xiao, and Michael Zakharyaschev. Answering SPARQL queries over databases under OWL 2 QL entailment regime. In *Proceedings of the Thirteenth International Semantic Web Conference (ISWC 2014)*, volume 8796 of *Lecture Notes in Computer Science,* `https://link.springer.com/bookseries/558`, pages 552–567, 2014.

[Koutris *et al.*, 2016] Paraschos Koutris, Paul Beame, and Dan Suciu. Worst-case optimal algorithms for parallel query processing. In *Proceedings of the Nineteenth International Conference on Database Theory (ICDT 2016)*, volume 48 of *Leibniz International Proceedings in Informatics,* `https://www.dagstuhl.de/en/publications/lipics`, pages 8:1–8:18, 2016.

[Koutris *et al.*, 2017] Paraschos Koutris, Tova Milo, Sudeepa Roy, and Dan Suciu. Answering conjunctive queries with inequalities. *Theory of Computing Systems*, 61(1):2–30, 2017.

[Lembo *et al.*, 2015] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. Inconsistency-tolerant query answering in ontology-based data access. *Journal of Web Semantics*, 33:3–29, 2015.

[Lembo *et al.*, 2017] Domenico Lembo, Riccardo Rosati, Valerio Santarelli, Domenico Fabio Savo, and Evgenij Thorstensen. Mapping repair in ontology-based data access evolving systems. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI 2017)*, pages 1160–1166, 2017.

[Lenzerini, 2002] Maurizio Lenzerini. Data integration: A theoretical perspective. In *Proceedings of the Twenty-First ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 2002)*, pages 233–246, 2002.

[Lenzerini, 2011] Maurizio Lenzerini. Ontology-based data management. In *Proceedings of the Twentieth International Conference on Information and Knowledge Management (CIKM 2011)*, pages 5–6, 2011.

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

[Levesque and Lakemeyer, 2000] Hector J. Levesque and Gerhard Lakemeyer. *The Logic of Knowledge Bases.* MIT Press, 2000.

[Levesque, 1984] Hector J. Levesque. Foundations of a functional approach to knowledge representation. *Artificial Intelligence*, 23(2):155–212, 1984.

[Levy *et al.*, 1995] Alon Y. Levy, Alberto O. Mendelzon, Yehoshua Sagiv, and Divesh Srivastava. Answering queries using views. In *Proceedings of the Fourteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 1995)*, pages 95–104, 1995.

[Levy *et al.*, 1996] Alon Y. Levy, Anand Rajaraman, and Joann J. Ordille. Querying heterogeneous information sources using source descriptions. In *Proceedings of the Twenty-Second International Conference on Very Large Data Bases (VLDB 1996)*, pages 251–262, 1996.

[Lobo *et al.*, 1992] Jorge Lobo, Jack Minker, and Arcot Rajasekar. *Foundations of Disjunctive Logic Programming.* Logic Programming. MIT Press, 1992.

[Lutz *et al.*, 2018] Carsten Lutz, Johannes Marti, and Leif 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)*, pages 389–398, 2018.

[Machan, 2009] Dyan Machan. DaaS: The new information goldmine. *Wall Street Journal*, pages 1–3, 2009.

[Madry, 2005] Aleksander Madry. Data exchange: On the complexity of answering queries with inequalities. *Information Processing Letters*, 94(6):253–257, 2005.

[Maier *et al.*, 1979] David Maier, Alberto O. Mendelzon, and Yehoshua Sagiv. Testing implications of data dependencies. *ACM Transactions on Database Systems*, 4(4):455–469, 1979.

[Meseguer and Qian, 1993] José Meseguer and Xiaolei Qian. A logical semantics for object-oriented databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data (SIGMOD 1993)*, pages 89–98, 1993.

[Miles and Bechhofer, 2009] Alistair Miles and Sean Bechhofer. SKOS Simple Knowledge Organization System. W3C Recommendation, World Wide Web Consortium, 2009. Available at `http://www.w3.org/TR/skos-reference`.

[Miles and Pérez-Agüera, 2007] Alistair Miles and José R. Pérez-Agüera. SKOS: Simple knowledge organisation for the web. *Cataloging and Classification Quarterly*, 43(3-4):69–83, 2007.

[Millstein *et al.*, 2003] Todd D. Millstein, Alon Y. Halevy, and Marc Friedman. Query containment for data integration systems. *Journal of Computer and System Sciences*, 66(1):20–39, 2003.

[Motik *et al.*, 2012] Boris Motik, Bernardo Cuenca Grau, Ian Horrocks, Zhe Wu, Achille Fokoue, and Carsten 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/`.

[Nash *et al.*, 2010] Alan Nash, Luc Segoufin, and Victor Vianu. Views and queries: Determinacy and rewriting. *ACM Transactions on Database Systems*, 35(3):21:1–21:41, 2010.

[Nikolaou *et al.*, 2019] Charalampos Nikolaou, Egor V. Kostylev, George Konstantinidis, Mark Kaminski, Bernardo Cuenca Grau, and Ian Horrocks. Foundations of ontology-based data access under bag semantics. *Artificial Intelligence*, 274:91–132, 2019.

[OECD, 2015] OECD. Making open science a reality. *OECD Science, Technology and Industry Policy Papers*, (25), 2015.

[Ortiz, 2018] Magdalena Ortiz. Improving data management using domain knowledge. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI 2018)*, pages 5709–5713, 2018.

[Papadimitriou, 1994] Christos H. Papadimitriou. *Computational Complexity*. Addison Wesley Publishing Company, 1994.

[Pentland, 2013] Alex Pentland. The data-driven society. *Scientific American*, 309(4):78–83, 2013.

[Poggi *et al.*, 2008] Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. Linking data to ontologies. *Journal on Data Semantics*, X:133–173, 2008.

[Poggi, 2016] Antonella Poggi. On the SPARQL direct semantics entailment regime for OWL 2 QL. In *Proceedings of the Twenty-Ninth International Workshop on Description Logics (DL 2016)*, volume 1577 of *CEUR Electronic Workshop Proceedings,* `http://ceur-ws.org/`, 2016.

[Rashid *et al.*, 2020] Sabbir M. Rashid, James P. McCusker, Paulo Pinheiro, Marcello P. Bax, Henrique Santos, Jeanette A. Stingone, Amar K. Das, and Deborah L. McGuinness. The semantic data dictionary – an approach for describing and annotating data. *Data Intelligence*, pages 443–486, 2020.

[Reingold, 2008] Omer Reingold. Undirected connectivity in log-space. *Journal of the ACM*, 55(4):17:1–17:24, 2008.

[Reiter, 1992] Raymond Reiter. What should a database know? *Journal of Logic Programming*, 14(1&2):127–153, 1992.

[Rosati and Almatelli, 2010] Riccardo Rosati and Alessandro Almatelli. Improving query answering over *DL-Lite* ontologies. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference (KR 2010)*, pages 290–300, 2010.

[Rosati, 2007] Riccardo Rosati. The limits of querying ontologies. In *Proceedings of the Eleventh International Conference on Database Theory (ICDT 2007)*, volume 4353 of *Lecture Notes in Computer Science,* `https://link.springer.com/bookseries/558`, pages 164–178, 2007.

[Sagiv and Yannakakis, 1980] Yehoshua Sagiv and Mihalis Yannakakis. Equivalences among relational expressions with the union and difference operators. *Journal of the ACM*, 27(4):633–655, 1980.

[Stockmeyer, 1976] Larry J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1–22, 1976.

[ten Cate *et al.*, 2009] Balder ten Cate, Laura Chiticariu, Phokion G. Kolaitis, and Wang-Chiew Tan. Laconic schema mappings: Computing the core with SQL queries. *Proceedings of the VLDB Endowment*, 2(1):1006–1017, 2009.

[Turing, 1937] Alan M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42(1):230–265, 1937.

[van der Meyden, 1993] Ron van der Meyden. Recursively indefinite databases. *Theoretical Computer Science*, 116(1):151–194, 1993.

[van der Meyden, 1997] Ron van der Meyden. The complexity of querying indefinite data about linearly ordered domains. *Journal of Computer and System Sciences*, 54(1):113–135, 1997.

[Vardi, 1982] Moshe Y. Vardi. The complexity of relational query languages. In *Proceedings of the Fourteenth ACM Symposium on Theory of Computing (STOC 1982)*, pages 137–146, 1982.

[Vollmer, 1999] Heribert Vollmer. *Introduction to Circuit Complexity - A Uniform Approach.* Texts in Theoretical Computer Science. An EATCS Series. Springer, 1999.

[Weibel *et al.*, 1998] Stuart Weibel, John A. Kunze, Carl Lagoze, and Misha Wolf. Dublin core metadata for resource discovery. *Request for Comments*, 2413:1–8, 1998.

[Wilkinson *et al.*, 2016] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J. G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A. C. 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J. Lusher, Maryann E. Martone, Albert Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons.

The FAIR guiding principles for scientific data management and stewardship. *Scientific Data*, 3, 2016.

[Xiao *et al.*, 2018] Guohui Xiao, Diego Calvanese, Roman Kontchakov, Domenico Lembo, Antonella Poggi, Riccardo Rosati, and Michael Zakharyaschev. Ontology-based data access: A survey. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI 2018)*, pages 5511–5519, 2018.

[Zaniolo, 1982] Carlo Zaniolo. Database relations with null values. In *Proceedings of the First ACM Symposium on Principles of Database Systems (PODS 1982)*, pages 27–33, 1982.

[Zheng *et al.*, 2013] Zibin Zheng, Jieming Zhu, and Michael R. Lyu. Service-generated big data and big data-as-a-service: An overview. In *Proceedings of the IEEE International Congress on Big Data (BigData Congress 2013)*, pages 403–410, 2013.