Complexity of Answering Counting Aggregate Queries over *DL-Lite*

Egor V. Kostylev[1] and Juan L. Reutter[2]

[1] University of Edinburgh, ekostyle@inf.ed.ac.uk
[2] PUC Chile and University of Edinburgh, jreutter@ing.puc.cl

Abstract. One of the main applications of description logics is the ontology-based data access model, which requires algorithms for query answering over ontologies. In fact, some description logics, like those in the DL-Lite family, are designed so that simple queries, such as conjunctive queries, are efficiently computable. In this paper we study counting aggregate queries over ontologies, i.e. queries which use aggregate functions COUNT and COUNT DISTINCT. We propose an intuitive semantics for certain answers for these queries, which conforms to the open world assumption. We compare our semantics with other approaches that have been proposed in different contexts. We establish data and combined computational complexity for the problems of answering counting aggregate queries over ontologies for several variants of DL-Lite.

1 Introduction

The growing popularity of ontologies as a paradigm for representing knowledge in the Semantic Web is based on the ability to describe incomplete information in the domain of interest.

Several variations of the *Web Ontology Language* (*OWL*) have been formalized to manage ontologies. Most of these languages correspond to various decidable fragments of first order logic, which are called *description logics* (*DLs*). However, applications like *ontology-based data access* (*OBDA*) require algorithms not only to decide standard reasoning problems, such as satisfiability and model checking, but also to answer database-style queries [1, 2]. This motivates the use of description logics of the *DL-Lite* family in, e.g. OWL 2 QL, which have been designed specifically to maximize expressive power while maintaining good query answering properties [3]. In particular, the computational complexity of answering simple queries such as *conjunctive queries* (*CQ*s) and *unions of conjunctive queries* (*UCQs*) over these DLs is the same as for relational databases [4, 5].

Some attention has recently been paid to the problem of answering various extensions of CQs and UCQs over ontologies. For example [6] study path queries over ontologies, while [7], [8] and [**?**] consider adding some form of negation to these simple queries. The general conclusion from these papers is that the complexity of evaluation of such queries is usually higher than for CQs and UCQs and even higher than for similar problems in relational databases. In some cases this difference in complexity is surprisingly high: e.g. while answering UCQs with inequalities is known to be efficiently computable for relational databases, the problem is undecidable when such a query is posed over *DL-Lite* ontologies.

Yet there is another extension of CQs that has received little attention in the context of OBDA – *aggregate queries*. These queries answer questions such as "How many children does Ann have?" or "What is the average salary over each department in the Pandidakterion?" Usually, they combine various aggregate functions, such as `MIN`, `MAX`, `SUM`, `AVERAGE`, `COUNT` and `COUNT DISTINCT` [9], together with a *grouping* functionality, as in the usual `GROUP BY` clause of SQL.

Aggregate queries are an important and heavily used part of almost every relational database query language, including SQL. In the context of the Semantic Web we expect a particular need for aggregates in the OBDA settings, with applications such as SPARQL under entailment regimes [10]. But despite their importance, the study of aggregate queries over ontologies has been lacking, save for a few exceptions [11].

The main reason for the lack of research in this direction is the difficulty of defining a semantics for aggregate queries over ontologies. The complication is that, unlike relational databases, in ontologies one assumes that every knowledge base instance is incomplete and describes a part of the infinite number of models of the knowledge base (i.e. the *open world assumption* is assumed), and a query may have a different answer on each of these models. For standard queries like CQs and UCQs this problem is usually overcome by computing the *certain answers* of queries, i.e. the tuples that are answers in all possible models [4]. This approach, however, is not suitable for aggregate queries, as the following shows.

Consider a knowledge base where Ann is a parent and the ontology asserts that every parent has at least one child. If nothing else is assumed then for every positive integer n there exists a model where Ann has n children. Thus, the answer to a simple query "How many children does Ann have?" in different models of the knowledge base can be any number greater than or equal to 1. The syntactic intersection of these answers (i.e. applying standard certain answers semantics) trivially gives us the empty set, which is clearly not satisfactory. As a different approach, [11] introduced *epistemic* semantics for aggregate queries. In a nutshell, the idea is to apply the aggregation function only to known values. For example, the epistemic answer to the query above is 0, because we do not definitely know anybody who is a child of Ann. But this is clearly not the desired answer: since Ann is a parent we know that she has at least one child. Hence the epistemic semantics does not always give a correct answer to `COUNT` queries.

As the first contribution of this paper, in Sec. 3 we embark on the task of defining a suitable semantics for answering what we call *counting aggregate queries*, which are queries that use `COUNT` or `COUNT DISTINCT` functions. Motivated by the original idea of certain answers, we seek to find the maximal information that is common in the answers to such a query for all the models of a knowledge base. This gives rise to the notion of *aggregate certain answers*, which can be explained as follows: a number is an aggregate certain answer to a counting query over a knowledge base if it does not exceed the result of the query over any model of this knowledge base. For instance, in the above example, even if we do not know precisely how many children Ann has, we know that she has at least one, and thus 1 is an aggregate certain answer to the query.

Of course this semantics is not well suited for aggregation primitives such as `SUM` or `AVERAGE`. But, as we show in this paper, it is a natural and useful semantics for aggregate queries that count.

Having established our semantics, we turn to the study of the algorithmic properties of aggregate certain answers computation for counting queries. We concentrate on ontologies of the *DL-Lite* family, in particular $\mathit{DL\text{-}Lite}_{core}$ and $\mathit{DL\text{-}Lite}_{\mathcal{R}}$ [4]. The choice of these DLs is twofold: first, as mentioned above, these formalisms are important in the OBDA settings; second, they are among the simplest DLs and hence good candidates to begin with.

As usual in the theory of DLs, in Sec. 4 we study these problems assuming that the query and the *terminology* (i.e. the *TBox*) are *fixed*, and the only input is the *assertions* (*ABox*). This corresponds to the *data complexity* of the problem in Vardi's taxonomy [12]. Somewhat surprisingly, our results show that the complexity of aggregate certain answers problem is resilient to the choice of both DL and counting function and is coNP-complete in all cases. In order to get a further understanding of the computational properties of the problems, in Sec. 5 we study their *combined complexity*, i.e. assume that the query, ABox and TBox are the input. Here we do find differences: both count distinct and count aggregate query answering are coNExpTime-complete for $\mathit{DL\text{-}Lite}_{\mathcal{R}}$; yet the former problem is Π_2^p-complete and the latter is in coNExpTime for $\mathit{DL\text{-}Lite}_{core}$. Hereby, the small increase of expressivity from $\mathit{DL\text{-}Lite}_{core}$ to $\mathit{DL\text{-}Lite}_{\mathcal{R}}$ makes at least the count distinct problem exponentially more difficult. As far as we are aware, these are the first tight complexity bounds for answering aggregate queries in the presence of ontologies.

Related Work Although mostly unexplored in the context of ontologies, semantics for aggregate queries have been already defined for other database settings that feature incomplete information. For example, an inconsistent database instance (w.r.t. a set of constraints) describes a set of repairs, each of which satisfies the constraints and can be obtained from the instance by a minimal number of transformations. Aggregate queries over inconsistent databases were explored in [13], where the *range semantics* was defined. Intuitively, this semantics corresponds to the *interval* between the minimal and the maximal possible answers to the query, amongst all the repairs of a given instance. The same semantics was adopted by [14, 15] in the context of data exchange.

However, the techniques from these papers cannot be immediately applied to ontologies, because of several specific properties. In particular, these papers consider variations of the *closed world assumption*, whereas in ontologies the open world assumption is assumed. Furthermore, data exchange settings are based on source-to-target dependencies and weakly acyclic target dependencies. This rules out all types of recursion in ontological knowledge, thus simplifying the study to a great extent.

In the context of ontologies, in [11] the range semantics itself was claimed to be trivially meaningless for aggregate queries over ontologies. For example, for almost any knowledge base we can construct a model such that the aggregate value of an `AVERAGE` query evaluates to any number. Similar examples can be given for all other standard aggregate functions, except for `COUNT` and `COUNT DISTINCT`, which are precisely the aggregates that are the focus of this paper. As we will show the computation of the upper bound of the range is almost trivial in these cases as well. But the lower bound of the range, i.e. the minimal possible value described above, is completely natural, and by no means trivial to compute. In fact, the lower bound of the range semantics is strongly related to our notion of aggregate certain answers as follows: a number is in the aggregate certain answers if and only if it is less than or equal to the lower bound of the range. Thus, this work on aggregate certain answers can be seen as an adaptation of the range semantics of [13] to ontologies.

This paper is an extended version of [16]. We sketch or even omit the proofs of lemmas in the paper, which will be included in the full version.

2 Preliminaries

Syntax of *DL-Lite* Let $A_0, A_1, \dots$ be *atomic concepts* and $P_0, P_1, \dots$ be *atomic roles. Concepts* C and *roles* E of *DL-Lite* languages are formed by the grammar

$$B ::= A_i \mid \exists R, \qquad R ::= P_i \mid P_i^-, \qquad C ::= B \mid \neg B, \qquad E ::= R \mid \neg R.$$

A *TBox* is a finite set of assertions. In the language of *DL-Lite$_{core}$* the assertions are of the form $B \sqsubseteq C$. In *DL-Lite$_\mathcal{R}$* the form $R \sqsubseteq E$ is also allowed. An *ABox* is a set of assertions of the forms $A_i(a)$ and $P_i(a, b)$ where *constants* a, b are from an *active domain* $\mathbb{D}$. A *knowledge base* (or *KB*) $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ of a *DL-Lite* language contains a TBox $\mathcal{T}$ of the language and an ABox $\mathcal{A}$.

Semantics of *DL-Lite* An *interpretation* $\mathcal{I} = (\mathbb{D}^\mathcal{I}, \cdot^\mathcal{I})$ contains a (possibly infinite) *domain* of *elements* $\mathbb{D}^\mathcal{I}$ such that $\mathbb{D} \subseteq \mathbb{D}^\mathcal{I}$, and maps each concept C to a subset $C^\mathcal{I}$ of $\mathbb{D}^\mathcal{I}$ and each role R to a binary relation $R^\mathcal{I}$ over $\mathbb{D}^\mathcal{I}$ such that

$$(P_i^-)^\mathcal{I} = \{(a, b) \mid (b, a) \in P_i^\mathcal{I}\},\ (\neg B)^\mathcal{I} = \mathbb{D}^\mathcal{I} \backslash B^\mathcal{I},$$
$$(\exists R)^\mathcal{I} = \{a \mid \exists b:\ (a, b) \in R^\mathcal{I}\},\ (\neg R)^\mathcal{I} = \mathbb{D}^\mathcal{I} \times \mathbb{D}^\mathcal{I} \backslash R^\mathcal{I}.$$

An interpretation $\mathcal{I}$ is a *model* of a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ (written $\mathcal{I} \models \mathcal{K}$) if for any assertion $B \sqsubseteq C$ in $\mathcal{T}$ it holds that $B^\mathcal{I} \subseteq C^\mathcal{I}$, for any $R \sqsubseteq E$ it holds that $R^\mathcal{I} \subseteq E^\mathcal{I}$, for any $A_i(a)$ in $\mathcal{A}$ it holds that $a \in A_i^\mathcal{I}$, and for any $P_i(a, b)$ it holds that $(a, b) \in P_i^\mathcal{I}$.

The definitions above say that $\mathbb{D} \subseteq \mathbb{D}^\mathcal{I}$ in every interpretation $\mathcal{I}$, which essentially means that for each constant a from the active domain $\mathbb{D}$ we have $a^\mathcal{I} = a$. By this we adopt the *unique name assumption* (*UNA*) on constants, which is conventional for *DL-Lite*. However, dropping this assumption does not affect any result of this paper, and we discuss explicitly how to adopt proofs wherever it is not straightforward.

Conjunctive queries A *conjunctive query* (or *CQ*) is an expression of the form

$$q(\mathbf{x}) \text{ :- } \exists \mathbf{y}\ \phi(\mathbf{x}, \mathbf{y}), \tag{1}$$

where $\mathbf{x}$ is a tuple of *free* variables, $\mathbf{y}$ is a tuple of *existential* variables, and the *body* $\phi(\mathbf{x}, \mathbf{y})$ is a conjunction of *atoms* of the form $A_i(u)$ or $P_i(u_1, u_2)$, where u, u_1, u_2 are variables from $\mathbf{x} \cup \mathbf{y}$.

A CQ (1), holds for an interpretation $\mathcal{I}$ and a tuple $\mathbf{t}$ of elements from $\mathbb{D}^{\mathcal{I}}$ (written $\mathcal{I} \models q(\mathbf{t})$) iff there exists an *evaluation* from q to $\mathbb{D}^{\mathcal{I}}$ for $\mathbf{t}$, i.e. a mapping $h : \mathbf{x} \cup \mathbf{y} \to \mathbb{D}^{\mathcal{I}}$, such that $h(\mathbf{x}) = \mathbf{t}$ and $h(\mathbf{z}) \in S^{\mathcal{I}}$, for every atom $S(\mathbf{z})$ in $\phi(\mathbf{x}, \mathbf{y})$. A tuple $\mathbf{t}$ is in the *certain answer* to a CQ (1) over a KB $\mathcal{K}$ if $\mathcal{I} \models q(\mathbf{t})$ holds for every model $\mathcal{I}$ of $\mathcal{K}$.

3 Counting Queries over Ontologies

The ability to evaluate aggregate queries is a default in every DBMS and is in the standard of SQL. However, as mentioned in the introduction, little attention to this type of queries has been paid in the context of ontologies. Starting to fill this gap, in this section we formally define counting aggregate queries over ontologies of *DL-Lite* family and compare this definition with existing notions in related areas.

3.1 Syntax and Semantics of Counting Queries

Following e.g. [9], an *aggregate conjunctive query* (or *ACQ*) is an expression

$$q(\mathbf{x}, f(\mathbf{z})) \text{ :- } \exists \mathbf{y}\ \phi(\mathbf{x}, \mathbf{y}, \mathbf{z}), \tag{2}$$

where $\mathbf{x}$ is a tuple of *free* variables, $\mathbf{y}$ is a tuple of *existential* variables and $\mathbf{z}$ is a tuple of *aggregation* variables; the *body* $\phi(\mathbf{x}, \mathbf{y}, \mathbf{z})$ is a conjunction of *atoms* of the form $A_i(u)$ or $P_i(u_1, u_2)$, where u, u_1, u_2 are variables from $\mathbf{x} \cup \mathbf{y} \cup \mathbf{z}$; and $f(\mathbf{z})$ is an *aggregation function.* In this paper we consider two such functions: the unary *count distinct* function $\mathit{Cntd}(z)$ and nullary *count* function $\mathit{Count}()$. We call such queries *counting ACQs.*

Example 1. Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a knowledge base where $\mathcal{T}$ consists of the assertion $\mathit{Parent} \sqsubseteq \exists \mathit{HasChild}$, and $\mathcal{A}$ consists of the assertion $\mathit{Parent}(\text{Ann})$. The query

$$q_1(x, \mathit{Count}()) \text{ :- } \exists y\ \mathit{Parent}(x) \land \mathit{HasChild}(x, y)$$

is an ACQ using the count function. Intuitively, it counts the children of each parent. The query

$$q_2(\mathit{Cntd}(y)) \text{ :- } \exists x\ \mathit{Parent}(x) \land \mathit{HasChild}(x, y)$$

is a count distinct ACQ. This query counts all different children having a parent.

To define the semantics of counting queries over a particular model we again follow [9]. We say that the *core* of an ACQ of the form (2) is the CQ $\bar{q}(\mathbf{x} \cup \mathbf{z}) \text{ :- } \exists \mathbf{y}\ \phi(\mathbf{x}, \mathbf{y}, \mathbf{z})$. Also, let $\mathbb{N}^{\infty}$ be the set of natural numbers with 0 and $+\infty$.

A *count* ACQ $q(\mathbf{x}, Count())$ holds for an interpretation $\mathcal{I}$, a tuple $\mathbf{t}$ of elements from $\mathbb{D}^{\mathcal{I}}$ and a number $n \in \mathbb{N}^{\infty}$ (written $\mathcal{I} \models q(\mathbf{t}, n)$) iff n is the number of distinct evaluations from the core $\bar{q}$ to $\mathbb{D}^{\mathcal{I}}$ for $\mathbf{t}$.

A *count distinct* ACQ $q(\mathbf{x}, Cntd(z))$ holds for an interpretation $\mathcal{I}$, a tuple $\mathbf{t}$ of elements from $\mathbb{D}^{\mathcal{I}}$ and a number $n \in \mathbb{N}^{\infty}$ (written $\mathcal{I} \models q(\mathbf{t}, n)$) iff n is the number of distinct elements $a \in \mathbb{D}^{\mathcal{I}}$ such that $\mathcal{I} \models \bar{q}(\mathbf{t}, a)$ for the core $\bar{q}$ of q.

Example 2. Coming back to Ex. 1, consider the interpretation $\mathcal{I}$ where $Parent^{\mathcal{I}} = \{\text{Ann}\}$ and $HasChild^{\mathcal{I}} = \{(\text{Ann}, \text{Joe})\}$, which is clearly a model for $\mathcal{K}$. Then it is not difficult to see that $\mathcal{I} \models q_1(\text{Ann}, 1)$ and $\mathcal{I} \models q_2(1)$. For the model $\mathcal{J}$ such that $Parent^{\mathcal{J}} = \{\text{Ann}, \text{Peter}\}$ and $HasChild^{\mathcal{J}} = \{(\text{Ann,Joe}),(\text{Ann,Rose}),(\text{Peter,Joe})\}$, it holds that $\mathcal{J} \models q_1(\text{Ann}, 2)$, $\mathcal{J} \models q_1(\text{Peter}, 1)$ and $\mathcal{J} \models q_2(2)$.

3.2 Certain Answers of Counting Queries over Ontologies

A knowledge base normally describes not a single model, but an infinite number of them. This is why one is typically interested in computing the *certain answers* of queries over a KB, which are usually defined as the intersection of the answers of the query over all possible models of KB [4, 7].

Unfortunately, a definition based on such a syntactical intersection is of little use for ACQs, since it would almost always be empty. For instance, for the query q_1 from Ex. 1 and 2 we have that $\mathcal{I} \models q_1(\text{Ann}, 1)$, and $\mathcal{I} \not\models q_1(\text{Ann}, 2)$, yet $\mathcal{J} \not\models q_1(\text{Ann}, 1)$ and $\mathcal{J} \models q_1(\text{Ann}, 2)$. This suggests avoiding using such a syntactic intersection when defining the semantics of ACQs over ontologies.

In the context of OBDA this problem has been identified before by [11]. Their solution was to concentrate only on aggregating over *epistemic* knowledge, i.e. over values which are explicitly mentioned in the ABox of a KB. Such epistemic aggregate queries usually have a non-empty certain answer, based on the intersection, for all standard aggregate queries, including Max and $Average$. However, for counting queries this answer may be somehow non-satisfactory. For example, the epistemic answer to the ACQ q_1 over $\mathcal{K}$ from Ex. 1 is $(\text{Ann}, 0)$, because we do not know anybody who is definitely a child of Ann.

That is why we suggest the following definition of certain answers of counting ACQs over DLs, which is essentially the *minimum* over possible values of the counting function over all the models of a KB. In particular, our certain answer to the query q_1 over $\mathcal{K}$ from Ex. 1 contains $(\text{Ann}, 1)$, which reflects the fact that we definitely know that Ann has at least one child in any model. We deem this definition to be in line with the open world assumption, adopted in ontologies.

Definition 1. *A non-negative number* $n \in \mathbb{N}^{\infty}$ *is in the* aggregate certain answers $Cert(q, \mathbf{t}, \mathcal{K})$ *for a counting ACQ* q*, tuple of elements* $\mathbf{t}$*, and a KB* $\mathcal{K}$ *iff* $n \leq \min_{\mathcal{I} \models \mathcal{K}} \{k \mid \mathcal{I} \models q(\mathbf{t}, k)\}$.

Note that a definition like above is non-trivial only for counting standard aggregate queries. Indeed, it relies on their simple property that the minimum above can potentially be any number greater than or equal to 0. For other aggregation functions it is not the case: e.g. such a minimum for $Average$ is trivially almost always $-\infty$.

3.3 Range Semantics of Aggregate Queries

The *range semantics* for aggregate queries was first proposed in [13] to study aggregate queries over inconsistent databases, and it was later adopted in data exchange [14, 15]. In the context of counting ACQs over ontologies it can be defined as follows.

The *range of answers* for a counting ACQ q, a tuple $\mathbf{t}$, and a KB $\mathcal{K}$ is the interval $[m(q, \mathbf{t}, \mathcal{K}), M(q, \mathbf{t}, \mathcal{K})]$, where

$$m(q, \mathbf{t}, \mathcal{K}) = \min_{\mathcal{I} \models \mathcal{K}} \{k \mid \mathcal{I} \models q(\mathbf{t}, k)\}, \qquad M(q, \mathbf{t}, \mathcal{K}) = \max_{\mathcal{I} \models \mathcal{K}} \{k \mid \mathcal{I} \models q(\mathbf{t}, k)\}.$$

It is easy to see that the lower bound of the range interval coincides with the maximal certain answer from Def. 1. Considering the upper bound, let's come back to Ex. 1. We can find a model $\mathcal{I}$ of $\mathcal{K}$ such that $\mathcal{I} \models q_1(\text{Ann}, n)$ for any number $n \geq 1$, i.e. in this case the upper bound is $+\infty$. The following proposition says that this is not unusual.

Proposition 1. *Given a counting ACQ q, a tuple of elements $\mathbf{t}$, and a DL-Lite KB $\mathcal{K}$ the upper endpoint $M(q, \mathbf{t}, \mathcal{K})$ of the range of answers belongs to the set $\{0, 1, +\infty\}$, and can be computed in polynomial time (in the size of q and $\mathcal{K}$).*

Proof. Indeed, $M(q, \mathbf{t}, \mathcal{K}) = 0$ iff $\langle \mathcal{T}, \mathcal{A} \cup \mathcal{A}_q \rangle$ has no model, where $\mathcal{A}_q$ is an ABox over the variables of q as constants, containing the atoms of q as assertions. Otherwise, we have that $M(q, \mathbf{t}, \mathcal{K}) = 1$ only if q uses *count*() and has no existentially quantified variables. In all the remaining cases we have that $M(q, \mathbf{t}, \mathcal{K}) = +\infty$, since nothing prevents a model with an infinite number of witnesses. □

We may thus say that the aggregate certain answers semantics from Def. 1 is in fact an adaptation of the range semantics of [13] to ontologies.

4 Data Complexity of Counting Queries

It has been argued many times that in usual database settings the size of the query and the TBox is much smaller than the size of the ABox (see e.g. [12] as a more general statement and [4] in the context of DL's). This is why in query answering over ontologies one usually explores data complexity of problems, i.e. only database knowledge from ABox is considered as part of the input. In this section we do the same for aggregate certain answers. Formally, let $\mathcal{X} \in \{core, \mathcal{R}\}$, $\mathcal{T}$ be a TBox over *DL-Lite*$_\mathcal{X}$ and $q(\mathbf{x}, f(z))$ be a counting ACQ. We are interested in the following family of problems.

DL-Lite$_\mathcal{X}$ f-Aggregate Certain Answers $(\mathcal{T}, q)$
Input: ABox $\mathcal{A}$, tuple $\mathbf{t}$, and number $n \in \mathbb{N}^\infty$.
Question: Is $n \in Cert(q, \mathbf{t}, \langle \mathcal{T}, \mathcal{A} \rangle)$?

4.1 Count Queries

We start with the lower bound for count ACQs.

Lemma 1. *There exist a DL-Lite$_{core}$ TBox $\mathcal{T}$ and a count ACQ q without free variables such that checking whether $n \in Cert(q, \mathbf{t}_\emptyset, \mathcal{K})$, where $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, for an ABox $\mathcal{A}$, a number n, and the empty tuple $\mathbf{t}_\emptyset$ is* coNP*-hard.*

Proof (sketch). Let A, B and E, P be atomic concepts and roles. Let $\mathcal{T} = \{A \sqsubseteq \exists P, \exists P^- \sqsubseteq B\}$ and $q(Count()) \text{ :- } \exists y_1 \ldots y_4\, B(y_1) \wedge E(y_2, y_3) \wedge P(y_2, y_4) \wedge P(y_3, y_4)$.

The proof is by a reduction from the complement of the NP-complete 3-colouring problem with an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ as input.

Let $\mathbb{D} = \mathcal{V} \cup \{r, g, b, a\}$. Let $\mathcal{A}$ contain $E(u, v)$ and $E(v, u)$ for each $(u, v) \in \mathcal{E}$, $A(v)$ for each $v \in \mathcal{V}$, $B(c)$ for each $c \in \{r, g, b\}$, and $E(a, a)$, $P(a, r)$.

It holds that $4 \in Cert(q, \mathbf{t}_\emptyset, \mathcal{K})$ iff $\mathcal{G}$ has no 3-colouring. □

This lemma continues to hold if one drops the UNA. To adopt the proof, it suffices to state that r, g and b belong to pairwise disjoint concepts, and that a belongs to a concept that is disjoint with a concept containing all v from $\mathcal{V}$.

The proof above make use of the non-connectivity of the query. It is an interesting open problem whether the result holds for connected queries.

Thus, the data complexity of count queries rises from P in the standard database case at least to coNP for *DL-Lite* knowledge bases. The following lemma establishes a matching upper bound for the problem.

Lemma 2. *Let $\mathcal{T}$ be a fixed DL-Lite$_\mathcal{R}$ TBox and $q(\mathbf{x}, Count())$ be a fixed count ACQ. Checking whether $n \in Cert(q, \mathbf{t}, \mathcal{K})$, where $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, for an ABox $\mathcal{A}$, a tuple $\mathbf{t}$, and a number n can be done in* coNP.

Proof (sketch). Given an interpretation $\mathcal{J}$ and a number k, it is well known that checking whether $\mathcal{J} \models \mathcal{K}$ and $\mathcal{J} \models q(\mathbf{t}, k)$ is in polynomial time (since q is fixed). Hence, it is enough to prove that if there exists a model $\mathcal{I}$ of $\mathcal{K}$ such that $\mathcal{I} \models q(\mathbf{t}, n_0)$ for a number n_0 then there exists a model $\bar{\mathcal{I}}$ of $\mathcal{K}$ of polynomial size in the size of $\mathcal{A}$ such that $\bar{\mathcal{I}} \models q(\mathbf{t}, \bar{n})$ for some number $\bar{n} \leq n_0$.

Note that $\mathcal{K}$ always has a model with a domain no bigger than $|\mathbb{D}| + |\mathcal{T}|$, so we may assume that $n_0 \leq (|\mathbb{D}| + |\mathcal{T}|)^{|q|}$ (which is polynomial since q is fixed).

Fix $\mathcal{I}$ as above. There exists a homomorphism $f : Can(\mathcal{K}) \to \mathcal{I}$, where $Can(\mathcal{K})$ is the *canonical model* of $\mathcal{K}$ (see the definition in e.g. [4]). W.l.o.g. we assume that it is surjective, i.e. $f(Can(\mathcal{K})) = \mathcal{I}$; otherwise we can drop elements and assertions of $\mathcal{I}$ which are not in the image of f, without increasing n_0.

Let $\mathbb{D}^*$ be all elements of $\mathbb{D}^\mathcal{I}$ which are either constants from $\mathbb{D}$ or images of variables by evaluations from the core of q to $\mathbb{D}^\mathcal{I}$. We can construct an interpretation $\hat{\mathcal{I}}$ with the domain $\mathbb{D}^{\hat{\mathcal{I}}} = \cup_{d \in \mathbb{D}^\mathcal{I} \setminus \mathbb{D}^*} f^{-1}(d) \cup \mathbb{D}^*$ and with a surjective homomorphism from $Can(\mathcal{K})$ so that $\hat{\mathcal{I}} \models \mathcal{K}$ and $\hat{\mathcal{I}} \models q(\mathbf{t}, \bar{n})$ for some $\bar{n} \leq n_0$.

For every element $d \in \mathbb{D}^{\hat{\mathcal{I}}} \setminus \mathbb{D}^*$ define $\mathcal{N}_q(d)$ as a sub-interpretation of $\mathbb{D}^{\hat{\mathcal{I}}}$ induced by all elements reachable from d by an (undirected) path though roles of length no more than $|q|$ and without intermediate nodes from $\mathbb{D}^*$. Define equivalence $\mathcal{N}_q(d) \sim \mathcal{N}_q(d')$ if there exists an isomorphism between $\mathcal{N}_q(d)$ and $\mathcal{N}_q(d')$ preserving $\mathbb{D}^*$.

Note that every element of the canonical model which is not in $\mathbb{D}$, has at most $|\mathcal{T}| + 1$ immediate neighbours. Hence each $d \in \mathbb{D}^{\hat{\mathcal{I}}} \setminus \mathbb{D}^*$ also has at most $|\mathcal{T}| + 1$ immediate neighbours in $\hat{\mathcal{I}}$. Moreover, it holds that $|\mathbb{D}^*| \leq n_0|q| + |\mathbb{D}|$. So, each $\mathcal{N}_q(d)$ is of polynomial size and there is only a polynomial number of equivalence classes induced by $\sim$. Consider the model $\bar{\mathcal{I}}$ obtained from $\hat{\mathcal{I}}$ by merging all d_1, d_2 such that $\mathcal{N}_q(d_1) \sim \mathcal{N}_q(d_2)$ and the distance from $\mathbb{D}$ to d_1 and d_2 in the canonical model modulo $|q|+1$ is the same. The model $\bar{\mathcal{I}}$ is as required, since such merging does not create new homomorphisms of the body of q. □

Note that the lower bound was shown for *DL-Lite*$_{core}$, while the upper bound holds for any *DL-Lite*$_{\mathcal{R}}$ KB. Since *DL-Lite*$_{\mathcal{R}}$ is more expressive than *DL-Lite*$_{core}$, the lemmas above give us the following complexity result.

Theorem 1. *The problem DL-Lite$_{\mathcal{X}}$ Count-*Aggregate Certain Answers *$(\mathcal{T}, q)$ is* coNP*-complete in data complexity for any $\mathcal{X} \in \{core, \mathcal{R}\}$.*

4.2 Count Distinct Queries

The coNP bounds also apply for count distinct ACQs. The lower bound is again established by reduction from the complement of the 3-colouring problem.

Lemma 3. *There exist a DL-Lite$_{core}$ TBox $\mathcal{T}$ and a count distinct ACQ q without free variables such that checking whether $n \in Cert(q, \mathbf{t}_\emptyset, \mathcal{K})$, where $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, for an ABox $\mathcal{A}$ and a number n is* coNP*-hard.*

Proof (sketch). Consider $\mathcal{T} = \{\exists E \sqsubseteq \exists P\}$ and $q(Cntd(z))$:- $\exists y_1 \ldots y_4\ P(y_1, z) \wedge R(y_1, y_2) \wedge P(y_2, y_3) \wedge P(y_4, y_3) \wedge E(y_4, y_2)$, where E, P, R are atomic roles.

Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be an input graph as in the proof of Lem. 1. Let $\mathbb{D}$ contain the set of elements $\{v, v_1, v_2, v_3, v_4, v_5\}$ for each $v \in \mathcal{V}$, and elements $a, a_1, a_2, a_3, r, g, b$. Let $\mathcal{A}$ contain the assertions $E(u, v)$ and $E(v, u)$ for each $(u, v) \in \mathcal{E}$; the assertions $R(v, v_1), P(v_1, v_2), P(v_3, v_2), E(v_3, v_1), R(v_4, v)$, $P(v_4, v_5)$ for each $v \in \mathcal{V}$; and the assertions $R(a, a_1)$, $P(a_1, a_2)$, $P(a_3, a_2)$, $E(a_3, a_1)$, $P(a, r)$, $P(a, g)$ and $P(a, b)$. It holds that $4 \in Cert(q, \mathbf{t}_\emptyset, \mathcal{K})$ iff $\mathcal{G}$ has no 3-colouring. □

This lemma again holds for the case when UNA is dropped, and the proof can be adopted in the same way as the proof of Lem. 1. The matching algorithm is also similar to the count case.

Lemma 4. *Let $\mathcal{T}$ be a fixed DL-Lite$_{\mathcal{R}}$ TBox and $q(\mathbf{x}, Cntd(z))$ be a fixed count distinct ACQ. Checking whether $n \in Cert(q, \mathbf{t}, \langle \mathcal{T}, \mathcal{A} \rangle)$ for an ABox $\mathcal{A}$, a tuple $\mathbf{t}$, and a number n can be done in* coNP*.*

The proof goes the same lines as the proof of Lem. 2 except that we bound n_0 by $|\mathbb{D}| + |\mathcal{T}|$, and include into $\mathbb{D}^*$ the active domain $\mathbb{D}$ and all homomorphic images of the aggregation variable z to $\mathcal{I}$. The following summarises the lemmas.

Theorem 2. *The problem DL-Lite$_{\mathcal{X}}$ Cntd-*Aggregate Certain Answers *$(\mathcal{T}, q)$ is* coNP*-complete in data complexity for any $\mathcal{X} \in \{core, \mathcal{R}\}$.*

5 Combined Complexity of Counting Queries

As pointed out in Sec. 4 data complexity is the most used measure of algorithms in any database settings. However, combined complexity has its own value for understanding fundamental properties of problems. In this section we study the combined complexity of computing aggregate certain answers. Formally, let $\mathcal{X} \in \{core, \mathcal{R}\}$ and f be a counting aggregate function. Now we are interested in the following family of problems.

DL-Lite$_{\mathcal{X}}$ f-AGGREGATE CERTAIN ANSWERS
Input: KB $\mathcal{K}$ over *DL-Lite*$_{\mathcal{X}}$, f query q, tuple $\mathbf{t}$, and number $n \in \mathbb{N}^{\infty}$.
Question: Is $n \in Cert(q, \mathbf{t}, \mathcal{K})$?

5.1 Count Queries

We start again with count queries. Recall the algorithm to compute the certain answers for count queries explained in the proof of Lem. 2. Note that, if one takes into consideration the size of the query and the TBox, then this algorithm naturally gives a coNExpTime upper bound; the only difference is that in this case the number of neighbourhoods is of exponential size (w.r.t. q and $\mathcal{T}$), and thus the instance we need to guess is of exponential size. Next we show that this bound is tight for *DL-Lite*$_{\mathcal{R}}$.

Lemma 5. *The decision problem DL-Lite*$_{\mathcal{R}}$ *Count*-AGGREGATE CERTAIN ANSWERS *is* coNExpTime-*hard.*

The proof is by a reduction from the complement of the satisfiability problem for first-order logic (FO) formulas in the Bernays-Schöfinkel class [17]. This class contains all FO formulae of form $\exists \mathbf{x} \, \forall \mathbf{y} \, \psi(\mathbf{x}, \mathbf{y})$, with ψ a quantifier-free formula not using function symbols or equalities. The reduction is inspired by the techniques used in [18] to show coNExpTime-hardness of query answering problems in data exchange context.

Unfortunately, the reduction above uses role inclusions in the TBox, i.e. it is applicable only to *DL-Lite*$_{\mathcal{R}}$. We leave open the exact complexity of the *DL-Lite*$_{core}$ *Count*-ACQ ANSWERING problem, although it is not difficult to adapt the results of the following section to obtain a Π_2^p lower bound. We have the summarizing theorem.

Theorem 3. *(1) The problem DL-Lite*$_{core}$ *Count*-AGGREGATE CERTAIN ANSWERS *is in* coNExpTime. *(2) The problem DL-Lite*$_{\mathcal{R}}$ *Count*-AGGREGATE CERTAIN ANSWERS *is* coNExpTime-*complete.*

5.2 Count Distinct Queries

Just as we did for count queries, we can easily obtain a coNExpTime upper bound for count distinct ones from the proof of Lem. 4. However, for *DL-Lite*$_{core}$ we

DL-Lite	Data complexity		Combined complexity	
	Count	*Cntd*	*Count*	*Cntd*
core	coNP-c	coNP-c	in coNExp	Π_2^p-c
$\mathcal{R}$	coNP-c	coNP-c	coNExp-c	coNExp-c

Table 1. A summary of the complexity results. Here "-c" stands for "-complete" and coNExp – for coNExpTime.

can improve the complexity by almost one exponential. The idea is to redefine the sub-interpretations used in the proof of Lem. 4 to have them of polynomial size, while keeping the possibility of merging them.

Lemma 6. *There exists a Π_2^p-algorithm which solves the problem DL-Lite$_{core}$ Cntd-*Aggregate Certain Answers.

In this case we have the matching lower bound.

Lemma 7. *The problem DL-Lite$_{core}$ Cntd-*Aggregate Certain Answers *is Π_2^p-hard.*

The proof is by reduction from the Π_2^p-complete $\forall\exists$ 3-SAT problem [19].

The remaining question is whether the algorithm for computing aggregate certain answers over *DL-Lite*$_{\mathcal{R}}$ knowledge bases is optimal. We settle this with our last lemma, shown by a reduction similar to the one in the proof of Lem. 5.

Lemma 8. *The decision problem DL-Lite$_{\mathcal{R}}$ Cntd-*Aggregate Certain Answers *is* coNExpTime-*hard.*

Summing up, we have our last theorem.

Theorem 4. *(1) The problem DL-Lite$_{core}$ Cntd-*Aggregate Certain Answers *is Π_2^p-complete. (2) The problem DL-Lite$_{\mathcal{R}}$ Cntd-*Aggregate Certain Answers *is* coNExpTime-*complete.*

6 Conclusion

In this paper we have defined an intuitive semantics for counting aggregate queries over ontologies and explored the computational complexity of the corresponding problems. The results, summarized in Table 1, show that the problems are decidable, but intractable. Hence, heuristics and approximations for answering ACQs are on high demand from the practical point of view, with applications, for instance, in the definition of general aggregation in SPARQL under entailment regimes. We consider the epistemic semantics as one of such approximations, since it has lower data complexity but does not always provide the desired answer. Our work settles the theoretical foundations for further discussion.

References

1. Calvanese, D., Giacomo, G.D., Lembo, D., Lenzerini, M., Poggi, A., Rodriguez-Muro, M., Rosati, R., Ruzzi, M., Savo, D.F.: The MASTRO system for ontology-based data access. Semantic Web **2**(1) (2011) 43–53
2. Kontchakov, R., Lutz, C., Toman, D., Wolter, F., Zakharyaschev, M.: The combined approach to ontology-based data access. In: IJCAI. (2011) 2656–2661
3. Cuenca Grau, B., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., Sattler, U.: Owl 2: The next step for OWL. Web Semant. **6**(4) (November 2008) 309–322
4. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The DL-Lite family. J. of Automated Reasoning **39**(3) (2007) 385–429
5. Artale, A., Calvanese, D., Kontchakov, R., Zakharyaschev, M.: The DL-Lite family and relations. J. Artif. Intell. Res. (JAIR) **36** (2009) 1–69
6. Bienvenu, M., Ortiz, M., Simkus, M.: Answering expressive path queries over lightweight DL knowledge bases. In Kazakov, Y., Lembo, D., Wolter, F., eds.: Description Logics. Volume 846 of CEUR Workshop Proceedings., CEUR-WS.org (2012)
7. Rosati, R.: The limits of querying ontologies. In Schwentick, T., Suciu, D., eds.: ICDT. Volume 4353 of Lecture Notes in Computer Science., Springer (2007) 164–178
8. Gutiérrez-Basulto, V., Ibáñez-García, Y.A., Kontchakov, R.: An update on query answering with restricted forms of negation. In: Proceedings of the 6th international conference on Web Reasoning and Rule Systems. RR'12, Berlin, Heidelberg, Springer-Verlag (2012) 75–89
9. Cohen, S., Nutt, W., Sagiv, Y.: Deciding equivalences among conjunctive aggregate queries. Journal of the ACM **54**(2) (2007)
10. Glimm, B., Ogbuji, C., Hawke, S., Herman, I., Parsia, B., Polleres, A., Seaborne, A.: SPARQL 1.1 entailment regimes (2013) W3C Recommendation 21 March 2013, `http://www.w3.org/TR/2013/REC-sparql11- entailment-20130321/`.
11. Calvanese, D., Kharlamov, E., Nutt, W., Thorne, C.: Aggregate queries over ontologies. In Elmasri, R., Doerr, M., Brochhausen, M., Han, H., eds.: ONISW, ACM (2008) 97–104
12. Vardi, M.Y.: The complexity of relational query languages (extended abstract). In: STOC. (1982) 137–146
13. Arenas, M., Bertossi, L., Chomicki, J., He, X., Raghavan, V., Spinrad, J.: Scalar aggregation in inconsistent databases. Theor. Comput. Sci. **296**(3) (March 2003) 405–434
14. Libkin, L.: Data exchange and incomplete information. In Vansummeren, S., ed.: PODS, ACM (2006) 60–69
15. Afrati, F., Kolaitis, P.G.: Answering aggregate queries in data exchange. In: Proceedings of the twenty-seventh ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. PODS '08, New York, NY, USA, ACM (2008) 129–138
16. Kostylev, E.V., Reutter, J.L.: Answering counting aggregate queries over ontologies of the DL-Lite family. In: Proc. of the 27th AAAI Conf. on Artificial Intelligence (AAAI). (2013)
17. Börger, E., Grädel, E., Gurevich, Y.: The Classical Decision Problem. Springer, Berlin (2001)

18. Arenas, M., Barceló, P., Reutter, J.L.: Query languages for data exchange: Beyond unions of conjunctive queries. Theory Comput. Syst. **49**(2) (2011) 489–564
19. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman (1979)

Appendix

This appendix contains full proofs for all the lemmas of the paper.

Lemma 1. *There exist a DL-Lite$_{core}$ TBox $\mathcal{T}$ and a count ACQ q without free variables such that checking whether $n \in Cert(q, \mathbf{t}_\emptyset, \mathcal{K})$, where $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, for an ABox $\mathcal{A}$, a number n, and the empty tuple $\mathbf{t}_\emptyset$ is* coNP*-hard.*

Proof. Let A, B be atomic concepts and E, P be atomic roles. Fix a TBox $\mathcal{T} = \{A \sqsubseteq \exists P, \exists P^- \sqsubseteq B\}$ and count ACQ

$$q(Count()) \text{ :- } \exists y_1 \ldots y_4\, B(y_1) \wedge E(y_2, y_3) \wedge P(y_2, y_4) \wedge P(y_3, y_4).$$

Consider the complement of the NP-complete 3-colouring problem with an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of vertices and $\mathcal{E}$ is the set of edges, as input and positive output iff the graph has no 3-colouring.

Let $\mathbb{D} = \mathcal{V} \cup \{r, g, b, a\}$. Let $\mathcal{A}$ contain assertions $E(u, v)$ and $E(v, u)$ for each $(u, v) \in \mathcal{E}$, the assertion $A(v)$ for each $v \in \mathcal{V}$, the assertion $B(c)$ for each $c \in \{r, g, b\}$, and assertions $E(a, a)$, $P(a, r)$.

Note that, from the construction we have that the count is at least 3 in every model $\mathcal{I}$ of the KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, i.e. $\mathcal{I} \models q(\mathbf{t}_\emptyset, 3)$.

Next we show that $4 \in Cert(q, \mathbf{t}_\emptyset, \mathcal{K})$ iff $\mathcal{G}(\mathcal{V}, \mathcal{E})$ has no 3-colouring.

($\Leftarrow$) Assume for the sake of contradiction that $\mathcal{G}$ has no 3-colouring, but $4 \notin Cert(q, \mathbf{t}_\emptyset, \mathcal{K})$. It means that there exists a model $\mathcal{I}$ for $\mathcal{K}$ such that $\mathcal{I} \models q(\mathbf{t}_\emptyset, 3)$ (we know from the observation above that the count cannot be any number less than 3).

Since ACQs are monotone, it is safe to assume that for every vertex $v \in \mathcal{V}$, there is a single pair (v, e) in $P^\mathcal{I}$, for some element $e \in \mathbb{D}^\mathcal{I}$.

Let us define the following coloring $\sigma : \mathcal{V} \to \{\mathsf{red}, \mathsf{green}, \mathsf{blue}\}$ of $\mathcal{G}$: for each vertex $v \in \mathcal{V}$, we have that

$$\begin{array}{lll} \sigma(v) = & \mathsf{red} & \text{iff } \mathcal{I} \models P(v, r), \\ \sigma(v) = & \mathsf{green} & \text{iff } \mathcal{I} \models P(v, g), \text{ and} \\ \sigma(v) = & \mathsf{blue} & \text{iff } \mathcal{I} \models P(v, b). \end{array}$$

All that is left to do is to show that σ is indeed a proper 3-colouring, which results in a contradiction.

First, we show that σ assigns a colour to each vertex in $\mathcal{V}$. For the sake of contradiction, assume the contrary. Then there must be a vertex v such that $P^\mathcal{I}$ does not contain any of $(v, r), (v, g)$, or (v, b). Since we know that $A(v)$ holds in $\mathcal{I}$, and since $A \sqsubseteq \exists P$, it follows that there is an element e distinct from r, g and b and such that $P^\mathcal{I}(v, e)$. But then $B^\mathcal{I}(e)$ holds since $\exists P^- \sqsubseteq B$. We can then construct a fourth evaluation h from the core of q to $\mathbb{D}^\mathcal{I}$: $h(y_1) = e$, $h(y_2) = h(y_3) = a$, and $h(y_4) = r$. This contradicts the fact that $\mathcal{I} \models q(\mathbf{t}_\emptyset, 3)$.

Next we show that σ is indeed a correct colouring. Assume for the sake of contradiction that this is not the case. Then there is an edge $(u, v) \in \mathcal{E}$ such that $\sigma(u) = \sigma(v)$. Let us assume without loss of generality that $\sigma(u) = \sigma(v) = \mathsf{red}$.

From the definition of σ, it means that the pairs (u, r) and (v, r) belong to $P^{\mathcal{I}}$. We can then construct a fourth evaluation h from the core of q to $\mathbb{D}^{\mathcal{I}}$: $h(y_1) = r$, $h(y_2) = u$, $h(y_3) = v$, and $h(y_4) = r$. This contradicts to the fact that $\mathcal{I} \models q(\mathbf{t}_\emptyset, 3)$.

We obtain that σ is a 3-colouring, which is a contradiction.

($\Rightarrow$) Assume that $4 \in \mathit{Cert}(q, \mathbf{t}_\emptyset, \mathcal{K})$, and assume for the sake of contradiction that there is a 3-colouring σ of $\mathcal{G}$.

Next we construct a model $\mathcal{I}$ of $\mathcal{K}$ such that $\mathcal{I} \models q(\mathbf{t}_\emptyset, 3)$, which results in a contradiction:

- $B^{\mathcal{I}} = \{r, g, b\}$; $A^{\mathcal{I}} = \{v \mid v \in \mathcal{V}\}$;
- $E^{\mathcal{I}} = \{(a, a)\} \cup \{(u, v) \mid (u, v) \in \mathcal{E} \text{ or } (v, u) \in \mathcal{E}\}$; and
- $P^{\mathcal{I}} = \{(v, r) \mid v \in \mathcal{V} \text{ and } \sigma(v) = \mathsf{red}\} \cup \{(v, g) \mid v \in \mathcal{V} \text{ and } \sigma(v) = \mathsf{green}\} \cup \{(v, b) \mid v \in \mathcal{V} \text{ and } \sigma(v) = \mathsf{blue}\}$.

Clearly $\mathcal{I}$ is indeed a model of $\mathcal{K}$. We know that there are 3 evaluations from the core of q to $\mathbb{D}^{\mathcal{I}}$, resulting of mapping the variable y_1 to either r, or b, or g; the variables y_2, y_3 to a; and the variable y_4 to r. From the definition of $B^{\mathcal{I}}$ and $E^{\mathcal{I}}$, any other evaluation must send y_2 and y_3 to some u and v such that $E^{\mathcal{I}}$ contains (u, v), i.e. $\{u, v\}$ is an edge in $\mathcal{G}$. But it means that there is an element $e \in \{r, g, b\}$ such that both (u, e) and (v, e) are in $P^{\mathcal{I}}$, and, therefore, that $\sigma(u) = \sigma(v)$. This contradicts our initial assumption. □

Lemma 2. *Let $\mathcal{T}$ be a fixed DL-Lite$_\mathcal{R}$ TBox and $q(\mathbf{x}, \mathit{Count}())$ be a fixed count ACQ. Checking whether $n \in \mathit{Cert}(q, \mathbf{t}, \mathcal{K})$, where $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, for an ABox $\mathcal{A}$, a tuple $\mathbf{t}$, and a number n can be done in* coNP.

Proof. The knowledge base $\mathcal{K}$ always has a model over number of elements no bigger than $|\mathbb{D}| + |\mathcal{T}|$. There exists at most $(|\mathbb{D}| + |\mathcal{T}|)^{|q|}$ evaluations from the core of q to this model. Hence, w.l.o.g. we may assume that $n \leq (|\mathbb{D}| + |\mathcal{T}|)^{|q|}$ (which is polynomial since q is fixed), where $|q|$ is the number of atoms in the body of q, because otherwise the answer to our decision problem is trivially "no".

Let k be a fixed constant, and consider the following simple algorithm: check all interpretations $\mathcal{J}$ over number of elements $|\mathbb{D}|^k$, whether $\mathcal{J} \models \mathcal{K}$ and $\mathcal{J} \models q(\mathbf{t}, n)$. This algorithm clearly runs in coNP, since checking whether $\mathcal{J} \models \mathcal{K}$ and $\mathcal{J} \models q(\mathbf{t}, n)$ can be done in polynomial time (because q is fixed). Hence, for the proof it is enough to prove the following statement. There exists a constant k that depends only on q and $\mathcal{T}$ for which if there is a model $\mathcal{I}$ of $\mathcal{K}$ such that $\mathcal{I} \models q(\mathbf{t}, n_0)$ for a number $n_0 < n$, then there exists a model $\bar{\mathcal{I}}$ of $\mathcal{K}$ over $|\mathbb{D}|^k$ elements such that $\bar{\mathcal{I}} \models q(\mathbf{t}, \bar{n})$ for some number $\bar{n} \leq n_0$.

The remainder of the proof is devoted to show this statement. Fix a model $\mathcal{I}$ of $\mathcal{K}$ such that $\mathcal{I} \models q(\mathbf{t}, n_0)$ for a number $n_0 < n$. There exists a homomorphism $f : \mathit{Can}(\mathcal{K}) \to \mathcal{I}$, where $\mathit{Can}(\mathcal{K})$ is the *canonical model* of $\mathcal{K}$ (see the definition in e.g. [4]). W.l.o.g. we assume that this homomorphism is surjective, i.e. $f(\mathit{Can}(\mathcal{K})) = \mathcal{I}$; since otherwise we could drop elements and assertions of

$\mathcal{I}$ which are not in the image of f, without increasing n_0. Essentially it means, that f "merges" some elements of $Can(\mathcal{K})$ to obtain $\mathcal{I}$.

Let $\mathbb{D}^*$ be all elements of $\mathbb{D}^{\mathcal{I}}$ which are either constants from $\mathbb{D}$ or images of variables by evaluations from the core of q to $\mathbb{D}^{\mathcal{I}}$. We can construct an interpretation $\hat{\mathcal{I}}$ with the domain $\mathbb{D}^{\hat{\mathcal{I}}} = \cup_{d\in\mathbb{D}^{\mathcal{I}}\setminus\mathbb{D}^*} f^{-1}(d) \cup \mathbb{D}^*$ and with a surjective homomorphism from $Can(\mathcal{K})$ so that $\hat{\mathcal{I}} \models \mathcal{K}$ and $\hat{\mathcal{I}} \models q(\mathbf{t}, \bar{n})$ for some $\bar{n} \leq n_0$. Intuitively, $\hat{\mathcal{I}}$ is obtained from $\mathcal{I}$ by "unmerging" all elements in the unonimous part of $\mathcal{I}$, which are not bounded by evaluations from the core of q.

For every element $d \in \mathbb{D}^{\hat{\mathcal{I}}}\setminus\mathbb{D}^*$ define the *neighbourhood* $\mathcal{N}_q(d)$ as a sub-interpretation of $\mathbb{D}^{\hat{\mathcal{I}}}$ induced by all elements reachable from d by an (undirected) path though roles of length no more than $|q|$ and without intermediate nodes from $\mathbb{D}^*$. Define equivalence $\mathcal{N}_q(d) \sim \mathcal{N}_q(d')$ if there exists an isomorphism between the neighbourhoods $\mathcal{N}_q(d)$ and $\mathcal{N}_q(d')$ preserving $\mathbb{D}^*$.

Note that every element of the canonical model which is not in $\mathbb{D}$, has at most $|\mathcal{T}| + 1$ immediate neighbours. Hence each $d \in \mathbb{D}^{\hat{\mathcal{I}}}\setminus\mathbb{D}^*$ also has at most $|\mathcal{T}| + 1$ immediate neighbours in $\hat{\mathcal{I}}$. Moreover, it holds that $|\mathbb{D}^*| \leq n_0|q| + |\mathbb{D}|$. So, the size of each $\mathcal{N}_q(d)$ is of order $|\mathbb{D}^{\hat{\mathcal{I}}}|^{p(|q|,|\mathcal{T}|)}$, where p is a fixed polynomial that depends only on q and $\mathcal{T}$. This is of polynomial size, since the query and TBox are assumed to be fixed. Furthermore, we obtain that there is only a polynomial number of equivalence classes induced by $\sim$. Hence, there exists a constant k such that the model $\bar{\mathcal{I}}$ obtained from $\hat{\mathcal{I}}$ by merging all d_1, d_2 such that

1. $\mathcal{N}_q(d_1) \sim \mathcal{N}_q(d_2)$, and
2. in the canonical model $Can(\mathcal{K})$ the distances from $\mathbb{D}$ to d_1 and d_2 are the same modulo $|q| + 1$,

has no more than $(|q|+1)|\mathbb{D}^{\mathcal{I}}|^k$ underlying elements. Moreover, since such merging does not create new evaluations from the core of q to $\mathbb{D}^{\hat{\mathcal{I}}}$, as required, we have that $\bar{\mathcal{I}} \models q(\mathbf{t}, \bar{n})$ for the number $\bar{n} \leq n_0$. Note that the condition 2 guarantees that all the cycles created by the merging are longer than any cycle in the query. □

Lemma 3. *There exist a DL-Lite$_{core}$ TBox $\mathcal{T}$ and a count distinct ACQ q without free variables such that checking whether $n \in Cert(q, \mathbf{t}_\emptyset, \mathcal{K})$, where $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, for an ABox $\mathcal{A}$ and a number n is* coNP*-hard.*

Proof. Consider the TBox $\mathcal{T} = \{\exists E \sqsubseteq \exists P\}$ and count distinct ACQ

$$q(Cntd(z)) \text{ :- } \exists y_1 \ldots y_4 \; P(y_1, z) \wedge R(y_1, y_2) \wedge P(y_2, y_3) \wedge P(y_4, y_3) \wedge E(y_4, y_2),$$

where E, P, and R are atomic roles.

Consider the complement of the NP-complete 3-colouring problem with the input graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ as in the proof of Lem. 1.

Let $\mathbb{D}$ contain the set of elements $\{v, v_1, v_2, v_3, v_4, v_5\}$ for each $v \in \mathcal{V}$, and elements $a, a_1, a_2, a_3, r, g, b$. Let $\mathcal{A}$ contain

- assertions $E(u,v)$ and $E(v,u)$ for each $(u,v) \in \mathcal{E}$;
- assertions $R(v,v_1), P(v_1,v_2), P(v_3,v_2), E(v_3,v_1), R(v_4,v)$, $P(v_4,v_5)$ for each $v \in \mathcal{V}$; and
- assertions $R(a,a_1), P(a_1,a_2), P(a_3,a_2), E(a_3,a_1), P(a,r), P(a,g)$, and $P(a,b)$.

Similarly to the proof of Lem. 1 from the construction we have that the count distinct is at least 3 in every model $\mathcal{I}$ of the KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, i.e. $\mathcal{I} \models q(\mathbf{t}_\emptyset, 3)$.

The proof of the fact that $4 \in \mathit{Cert}(q, \mathbf{t}_\emptyset, \mathcal{K})$ iff $\mathcal{G}(\mathcal{V}, \mathcal{E})$ has no 3-colouring goes the same lines as the proof of the Lem. 1.

For the direction ($\Leftarrow$), we again assume for the sake of contradiction that $\mathcal{G}$ has no 3-colouring, but $4 \notin \mathit{Cert}(q, \mathbf{t}_\emptyset, \mathcal{K})$, i.e. there exists a model $\mathcal{I}$ of $\mathcal{K}$ such that $\mathcal{I} \models q(\mathbf{t}_\emptyset, 3)$. From $\mathcal{I}$ we show how to construct a coloring for $\mathcal{G}$. From this coloring and our assumption that there is no 3-colouring, we conclude that either there exists a vertex v such that $P^{\mathcal{I}}(v,e)$ for some element $e \notin \{r,g,b\}$, or there exists an edge $\{u,v\}$ in $\mathcal{E}$, such that $P^{\mathcal{I}}(u,e)$ and $P^{\mathcal{I}}(v,e)$. In the first case, there exists a evaluation h from the core of q to $\mathbb{D}^{\mathcal{I}}$ such that $h(z) = e$, which implies $4 \in \mathit{Cert}(q, \mathbf{t}_\emptyset, \mathcal{K})$. In the second case, here exists a evaluation h from the core of q to $\mathbb{D}^{\mathcal{I}}$ such that $h(z) = v_5$, which again implies $4 \in \mathit{Cert}(q, \mathbf{t}_\emptyset, \mathcal{K})$. However, this contradicts the assumption.

For the direction ($\Rightarrow$) we construct a model $\mathcal{I}$ of $\mathcal{K}$ such that $\mathcal{I} \models q(\mathbf{t}_\emptyset, 3)$ exactly as in the proof of Lem. 1. □

Lemma 4. *Let $\mathcal{T}$ be a fixed DL-Lite$_\mathcal{R}$ TBox and $q(\mathbf{x}, \mathit{Cntd}(z))$ be a fixed count distinct ACQ. Checking whether $n \in \mathit{Cert}(q, \mathbf{t}, \langle \mathcal{T}, \mathcal{A} \rangle)$ for an ABox $\mathcal{A}$, a tuple $\mathbf{t}$, and a number n can be done in* coNP.

Proof. The proof goes almost the same lines as the proof of Lem. 2.

The knowledge base $\mathcal{K}$ always has a model over number of elements no bigger than $|\mathbb{D}| + |\mathcal{T}|$. There exists at most $|\mathbb{D}| + |\mathcal{T}|$ images of the variable z by evaluations from the core of q to this model. Hence, w.l.o.g. we may assume that $n \leq |\mathbb{D}| + |\mathcal{T}|$, because otherwise the answer to our decision problem is trivially "no".

Consider again the following simple algorithm: check all interpretations $\mathcal{J}$ over number of elements $|\mathbb{D}|^k$, where k is the constant defined below, whether $\mathcal{J} \models \mathcal{K}$ and $\mathcal{J} \models q(\mathbf{t}, n)$. This algorithm clearly runs in coNP, since checking whether $\mathcal{J} \models \mathcal{K}$ and $\mathcal{J} \models q(\mathbf{t}, n)$ can be done in polynomial time (because q is fixed). Hence, again, it is enough to prove that if there exists a model $\mathcal{I}$ of $\mathcal{K}$ such that $\mathcal{I} \models q(\mathbf{t}, n_0)$ for a number $n_0 < n$ then there exists a model $\bar{\mathcal{I}}$ of $\mathcal{K}$ over $|\mathbb{D}|^k$ elements such that $\bar{\mathcal{I}} \models q(\mathbf{t}, \bar{n})$ for some number $\bar{n} \leq n_0$.

Fix a model $\mathcal{I}$ of $\mathcal{K}$ such that $\mathcal{I} \models q(\mathbf{t}, n_0)$ for a number $n_0 < n$. There exists a homomorphism $f : \mathit{Can}(\mathcal{K}) \to \mathcal{I}$, and by the same reasons as in the proof of Lem. 2 we assume that this homomorphism is surjective, i.e. $f(\mathit{Can}(\mathcal{K})) = \mathcal{I}$.

Let $\mathbb{D}^*$ be all elements of $\mathbb{D}^{\mathcal{I}}$ which are either constants from $\mathbb{D}$ or images of the variable z by evaluations from the core of q to $\mathbb{D}^{\mathcal{I}}$. We can construct an interpretation $\hat{\mathcal{I}}$ with the domain $\mathbb{D}^{\hat{\mathcal{I}}} = \cup_{d \in \mathbb{D}^{\mathcal{I}} \setminus \mathbb{D}^*} f^{-1}(d) \cup \mathbb{D}^*$ and with a surjective homomorphism from $Can(\mathcal{K})$ so that $\hat{\mathcal{I}} \models \mathcal{K}$ and $\hat{\mathcal{I}} \models q(\mathbf{t}, \bar{n})$ for some $\bar{n} \leq n_0$. Essentially, $\hat{\mathcal{I}}$ is obtained from $\mathcal{I}$ by "unmerging" all the elements in the unonimous part of $\mathcal{I}$, which are not images of the variable z by evaluations from the core of q.

Construct the model $\bar{\mathcal{I}}$ of $\mathcal{K}$ from $\hat{\mathcal{I}}$ exactly in the same way as in the proof of Lem. 2, by merging elements with isomorphic neighbourhoods $\mathcal{N}_q(d)$ and the same distance from $\mathbb{D}$ modulo $|q| + 1$. Since $|\mathbb{D}^*| \leq n_0 + |\mathbb{D}|$, and there are only polynomial number of neighbourhoods of polynomial size, there exists a constant k such that the model $\bar{\mathcal{I}}$ has no more than $(|q| + 1)|\mathbb{D}|^k$ underlying elements. Moreover, since such merging does not create new images of the variable z by evaluations from the core of q to $\mathbb{D}^{\hat{\mathcal{I}}}$, as required, we have that $\bar{\mathcal{I}} \models q(\mathbf{t}, \bar{n})$ for the number $\bar{n} \leq n_0$. □

Lemma 5. *The decision problem DL-Lite$_{\mathcal{R}}$ Count-*AGGREGATE CERTAIN ANSWERS *is* coNExpTime*-hard.*

Proof. The coNExpTime-hardness is established by a reduction from the satisfiability problem for the Bernays-Schönfinkel class of Boolean FO formulas, which is known to be NExpTime-complete (see, e.g., [17]), to the complement of the counting problem. Formally, the Bernays-Schönfinkel class of Boolean FO formulas is defined as the class of all FO formulas of the form $\exists \mathbf{x}\, \forall \mathbf{y}\, \psi(\mathbf{x}, \mathbf{y})$, where ψ is quantifier-free and does not contain function symbols and equalities.

Let $\exists \mathbf{x}\, \forall \mathbf{y}\, \psi(\mathbf{x}, \mathbf{y})$ be such a formula in the Bernays-Schönfinkel class. Let also $\mathbf{x} = x_1, \ldots, x_n$ and $\mathbf{y} = y_1, \ldots, y_m$. For the sake of readability, we first assume that ψ mentions a single relation symbol P, of arity r. Later we explain how to modify the proof to work with any arbitrary relational vocabulary. Finally, w.l.o.g. we assume that ψ is not atomic and let $\psi^1, \ldots, \psi^p$ be an enumeration of all the sub-formulas of ψ such that $\psi^p = \psi$.

Next we will show how to construct in polynomial time a *DL-Lite*$_{\mathcal{R}}$ KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ and a Boolean *Count*-ACQ q such that $2 \in Cert(q, \mathbf{t}_\emptyset, \mathcal{K})$ (where $\mathbf{t}_\emptyset$ is the empty tuple) iff $\exists \mathbf{x}\, \forall \mathbf{y}\, \psi(\mathbf{x}, \mathbf{y})$ is satisfiable, i.e. has a model. Along the proof we will use the following property of this formula: either it is unsatisfiable, or it has a model with at most n elements (see, e.g. [17]).

We start with the construction of the Boolean *Count*-ACQ

$$q(Count()) \text{ :- } \exists \mathbf{s}\ A^1(s^1) \wedge \cdots \wedge A^n(s^n) \wedge S(s_1) \wedge TV_F^p(s_2, s_3) \wedge C(s_3) \wedge \bigwedge_{\ell=1}^{p} \phi^\ell,$$

where $\mathbf{s}$ is the tuple of all variables in the query q, including the ones appearing below, and

- for each ℓ, $1 \leq \ell \leq p$ such that the sub-formula $\psi^\ell \equiv P(z_{h_1}, \ldots, z_{h_r})$ is atomic, we have that

$$\begin{aligned}\phi^\ell \equiv\ & R_0(u_1^\ell) \wedge R(u_1^\ell, v_{1,1}^\ell) \wedge \gamma_1^\ell \wedge TV_R(v_{1,r}^\ell, v_1^\ell) \wedge C(v_1^\ell)\ \wedge \\ & F_0(u_1^\ell) \wedge F(u_1^\ell, w_{1,1}^\ell) \wedge \delta_1^\ell \wedge TV_\ell(w_{1,m}^\ell, w_1^\ell) \wedge U(w_1^\ell)\ \wedge \\ & R_0(u_2^\ell) \wedge R(u_2^\ell, v_{2,1}^\ell) \wedge \gamma_2^\ell \wedge TV_R(v_{2,r}^\ell, v_2^\ell) \wedge U(v_2^\ell)\ \wedge \\ & F_0(u_2^\ell) \wedge F(u_2^\ell, w_{2,1}^\ell) \wedge \delta_2^\ell \wedge TV_\ell(w_{2,m}^\ell, w_2^\ell) \wedge C(w_2^\ell),\end{aligned}$$

where γ_j^ℓ, $1 \leq j \leq 2$, contains

the atom $R(v_{j,k}^\ell, v_{j,k+1}^\ell)$ for each k, $1 \leq k \leq r-1$,

the atom $V(v_{j,k}^\ell, t_{j,k}^\ell)$ for each k, $1 \leq k \leq r$,

the atom $A^i(t_{j,k}^\ell)$ for each k, $1 \leq k \leq r$, where i, $1 \leq i \leq n$, is the number such that z_{h_k} (i.e. the k'th variable of ψ^ℓ) is x_i, and

the atom $V(w_{j,i}^\ell, t_{j,k}^\ell)$ for each k, $1 \leq k \leq r$, where i, $1 \leq i \leq n$, is the number such that z_{h_k} is y_i,

and δ_j^ℓ, $1 \leq j \leq 2$, contains the atom $F(w_{j,k}^\ell, w_{j,k+1}^\ell)$ for each k, $1 \leq k \leq m-1$.

- for each ℓ, $1 \leq \ell \leq p$, such that $\psi^\ell = \psi^{\ell_1} \vee \psi^{\ell_2}$, we have that

$$\begin{aligned}\phi^\ell \quad \equiv \quad & TV_F^\ell(u_1^\ell, v_1^\ell) \wedge TV_F^{\ell_1}(u_1^\ell, w_1^\ell) \wedge TV_F^{\ell_2}(u_1^\ell, t_1^\ell) \wedge U(v_1^\ell) \wedge C(w_1^\ell) \wedge C(t_1^\ell) \wedge \\ & \quad TV_F^\ell(u_2^\ell, v_2^\ell) \wedge TV_F^{\ell_1}(u_2^\ell, w_2^\ell) \wedge C(v_2^\ell) \wedge U(w_2^\ell) \wedge \\ & \qquad TV_F^\ell(u_3^\ell, v_3^\ell) \wedge TV_F^{\ell_2}(u_3^\ell, t_3^\ell) \wedge C(v_3^\ell) \wedge U(t_3^\ell);\end{aligned}$$

- for each ℓ, $1 \leq \ell \leq p$, such that $\psi^\ell = \neg\psi^{\ell_1}$, we have that

$$\begin{aligned}\phi^\ell \quad \equiv \quad & TV_F^\ell(u_1^\ell, v_1^\ell) \wedge TV_F^{\ell_2}(u_1^\ell, w_1^\ell) \wedge U(v_1^\ell) \wedge U(w_1^\ell) \wedge \\ & \qquad TV_F^\ell(u_2^\ell, v_2^\ell) \wedge TV_F^{\ell_1}(u_2^\ell, w_2^\ell) \wedge C(v_2^\ell) \wedge C(w_2^\ell).\end{aligned}$$

Next we define the KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ and start with the ABox $\mathcal{A}$.

1. The active domain contains the constants $a^1, \ldots, a^n$ (corresponding to the elements of the model of ψ). The ABox $\mathcal{A}$ for every i, $1 \leq i \leq n$, contains the assertion $A^i(a^i)$.
2. The active domain contains the constants 0 and 1 (corresponding to the truth values **false** and **true**). The ABox $\mathcal{A}$ contains the assertions $S(0)$, $S(1)$, $C(0)$ and $U(1)$.
3. The active domain contains the constant c (which starts the "computation" of ψ). The ABox $\mathcal{A}$ contains the assertions $START_R(c)$ and $START_F(c)$.
4. The active domain contains the constants $b, e_1, \ldots, e_n, d_1, \ldots, d_m, f, g^{q+1}, \ldots,$ g^p. These constants, along with the following assertions, force every model of $\mathcal{K}$ to have one homomorphism from (the existential closure of) ϕ^ℓ. The ABox $\mathcal{A}$ contains the assertions:
   - $R_0(b), R(b, e_1)$ and $R(e_k, e_{k+1})$ for each i, $1 \leq k \leq r-1$;
   - $V(e_k, a^i)$ for each k and i, $1 \leq k \leq r$, $1 \leq i \leq n$;
   - $V(e_k, f)$ for each k, $1 \leq k \leq r$;
   - $TV_R(e_r, 0)$ and $TV_R(e_r, 1)$;
   - $F_0(b), F(b, d_1)$ and $F(d_k, d_{k+1})$ for each i, $1 \leq k \leq m-1$;
   - $V(d_k, f)$ for each k, $1 \leq k \leq m$;
   - $TV_F^\ell(d_m, 0)$ and $TV_F^\ell(d_m, 1)$ for each ℓ, $1 \leq \ell \leq p$ such that ψ^ℓ is atomic;
   - $TV_F^\ell(g^\ell, 0), TV_F^{\ell_1}(g^\ell, 0), TV_F^{\ell_2}(g^\ell, 0), TV_F^\ell(g^\ell, 1), TV_F^{\ell_1}(g^\ell, 1), TV_F^{\ell_2}(g^\ell, 1)$ for each ℓ, $1 \leq \ell \leq p$, such that $\psi^\ell = \psi^{\ell_1} \vee \psi^{\ell_2}$;

- $TV_F^\ell(g^\ell, 0), TV_F^{\ell_1}(g^\ell, 0), TV_F^\ell(g^\ell, 1)$ and $TV_F^{\ell_1}(g^\ell, 1)$ for each ℓ, $1 \leq \ell \leq p$, such that $\psi^\ell = \neg\psi^{\ell_1}$.

Finally, the TBox $\mathcal{T}$ consists of the following three parts.

1. The first part essentially assigns elements to the existential variables $\mathbf{x}$:
   - for each i, $1 \leq i \leq n$, the TBox $\mathcal{T}$ contains the assertions $\exists(V^i)^- \sqsubseteq A^i$ and $V^i \sqsubseteq V$.
2. The second part essentially assigns a truth value to every fact $P(a^{i_1}, \dots, a^{i_r})$:
   - for each i, $1 \leq i \leq n$, the TBox $\mathcal{T}$ contains the assertions $START_R \sqsubseteq R_0$ and $START_R \sqsubseteq \exists R_1^i$;
   - for each i, j and k, $1 \leq i \leq n$, $1 \leq j \leq n$, $1 \leq k \leq r-1$, the TBox $\mathcal{T}$ contains the assertion $\exists(R_k^i)^- \sqsubseteq R_{k+1}^j$;
   - for each i and k, $1 \leq i \leq n$, $1 \leq k \leq r$, the TBox $\mathcal{T}$ contains the assertions $\exists(R_k^i)^- \sqsubseteq \exists V^i$ and $R_k^i \sqsubseteq R$;
   - for each i, $1 \leq i \leq n$, the TBox $\mathcal{T}$ contains the assertion $\exists(R_r^i)^- \sqsubseteq \exists TV_R$;
   - the TBox $\mathcal{T}$ contains the assertion $\exists(TV_R)^- \sqsubseteq S$.
3. The third part essentially assigns a truth value to every sub-formula ψ_ℓ of ψ for every element assignment of variables $\mathbf{y}$:
   - for each i, $1 \leq i \leq n$, the TBox $\mathcal{T}$ contains the assertions $START_F \sqsubseteq F_0$ and $START_F \sqsubseteq \exists F_1^i$;
   - for each i, j and k, $1 \leq i \leq n$, $1 \leq j \leq n$, $1 \leq k \leq m-1$, the TBox $\mathcal{T}$ contains the assertion $\exists(F_k^i)^- \sqsubseteq F_{k+1}^j$;
   - for each i and k, $1 \leq i \leq n$, $1 \leq k \leq m$, the TBox $\mathcal{T}$ contains the assertions $\exists(F_k^i)^- \sqsubseteq \exists V^i$ and $F_k^i \sqsubseteq F$;
   - for each i and ℓ, $1 \leq i \leq n$, $1 \leq \ell \leq p$, the TBox $\mathcal{T}$ contains the assertion $\exists(F_m^i)^- \sqsubseteq \exists TV_F^\ell$;
   - for each ℓ, $1 \leq \ell \leq p$, the TBox $\mathcal{T}$ contains the assertion $\exists(TV_F^\ell)^- \sqsubseteq S$.

Having the construction of the KB $\mathcal{K}$ completed next we show the correctness of the reduction. We start with the following claim.

Claim. Let $\mathcal{I}$ be a model for the KB $\mathcal{K}$ above. Then there are exactly two evaluations from the core of the *Count*-ACQ q to $\mathcal{I}$ that map all the existential variables $\mathbf{s}$ of q to the constants in the active domain of $\mathcal{A}$.

Proof (of claim). Both evaluations can be constructed as follows: map the variable s_1 of q to either 0 or 1 (we know this is valid since $\mathcal{A}$ contains assertions $S(1)$ and $S(0)$). In the same fashion, map each variable s^i, $1 \leq i \leq n$, of q to the constant a^i (this is again valid since $\mathcal{A}$ contains all the assertions $A^i(a^i)$). For the rest of the variables of q one can check that all of them can be mapped to constants b, d, e, f, g (with proper indexes) that appear already in the assertions of $\mathcal{A}$. That no other evaluations can be constructed in this fashion can be checked by a straightforward inspection of the construction. □

It follows from the above claim that the core of the query q has at least two evaluations to each model of $\mathcal{K}$, and thus that $2 \leq m(q, \mathbf{t}_\emptyset, \mathcal{K})$ (recall that $\mathbf{t}_\emptyset$ is the empty tuple). This property is crucial for this proof.

We now show that $2 \in \mathit{Cert}(q, \mathbf{t}_\emptyset, \mathcal{K})$ iff $\exists \mathbf{x} \forall \mathbf{y}\, \psi(\mathbf{x}, \mathbf{y})$ is satisfiable.

($\Rightarrow$) Assume that $2 \in \mathit{Cert}(q, \mathbf{t}_\emptyset, \mathcal{K})$. Then there is a model $\mathcal{I}$ for $\mathcal{K}$ such that there are exactly two evaluations from the core of q to $\mathcal{I}$ (we know that there are at least two of them). Without loss of generality we assume that $\mathcal{I}$ is minimal, in the sense that all the tuples in the interpretation of concepts and roles in $\mathcal{I}$ witness a certain TBox or ABox assertions. If not, one can always remove this extra tuples, ending up with a model with no greater number of evaluations, since removing them can never create extra evaluations from the core of q to $\mathcal{I}$.

We can deduce several properties of the model $\mathcal{I}$.

- The interpretation of S in $\mathcal{I}$ contains only elements 0 and 1. Therefore, these are the only constants in the ranges (i.e. second components) of interpretations of TV_R and TV_F^ℓ, $1 \leq \ell \leq p$, in $\mathcal{I}$. Otherwise, one can construct additional homomorphisms by mapping the variable s_1 in the atom $S(s_1)$ of the body of q to one of the other elements.
- By similar reasons, the interpretation of each A^i, $1 \leq i \leq n$, contains exclusively the constant a^i, and therefore the range of the interpretation of V^i in $\mathcal{I}$ does the same.

Let now $\mathcal{M}$ be a model over the vocabulary of ψ consisting of the elements $\{a^1, \dots, a^n\}$, and assume that the interpretation $P^\mathcal{M}$ of the only relation P over $\mathcal{M}$ is as follows: each r-tuple $(a^{i_1}, \dots, a^{i_r})$ of elements in $\{a^1, \dots, a^n\}$ belongs to $P^\mathcal{M}$ if and only if there are elements $\bar{a}^{i_1}, \dots, \bar{a}^{i_r}$ in $\mathcal{I}$ so that the following assertions holds in $\mathcal{I}$:

$$\begin{aligned}&R_1^{i_1}(c, \bar{a}^{i_1}), R_2^{i_2}(\bar{a}^{i_1}, \bar{a}^{i_2}), \dots, R_r^{i_r}(\bar{a}^{i_{r-1}}, \bar{a}^{i_r}),\\ &\quad V(\bar{a}^{i_1}, a^{i_1}), \dots, V(\bar{a}^{i_r}, a^{i_r}), TV_R(\bar{a}^{i_r}, 1).\end{aligned}$$

We now show that all possible assignments for variables $y_1, \dots, y_m$ in ψ to variables from $\{a^1, \dots, a^n\}$ satisfy ψ, in the case when each x_i is assigned the element a_i, for $1 \leq i \leq n$. In order to do that, we prove the following claim by induction.

Claim. A sub-formula ψ^ℓ of ψ is satisfied by the assignment $\tau : \mathbf{x} \cup \mathbf{y} \to \{a^1, \dots, a^n\}$ such that $\tau(x_i) = a^i$, $1 \leq i \leq n$, and $\tau(y_k) = a^{\sigma(k)}$, $1 \leq k \leq m$, where σ is a mapping from $\{1, \dots, m\}$ to $\{1, \dots, n\}$, if and only if there are elements $\hat{a}^{k_1}, \dots, \hat{a}^{k_m}$ in $\mathcal{I}$ so that the following assertions are in $\mathcal{I}$:

$$\begin{aligned}&F_1^{\sigma(1)}(c, \hat{a}^{k_1}), F_2^{\sigma(2)}(\hat{a}^{k_1}, \hat{a}^{k_2}), \dots, F_m^{\sigma(m)}(\hat{a}^{k_{m-1}}, \hat{a}^{k_m}),\\ &\quad V(\hat{a}^{k_1}, a^{\sigma(1)}), \dots, V(\hat{a}^{k_m}, a^{\sigma(m)}), TV_F^\ell(\hat{a}^{k_m}, 1).\end{aligned}$$

Proof (of claim).

Basis. The base case is when ψ^ℓ is an atomic formula of form $P(z_{h_1}, \dots, z_{h_r})$, where each z_{h_k} is one of $x_1, \dots, x_n, y_1, \dots, y_m$. Assume for the sake of contradiction that there are constants $\hat{a}^{k_1}, \dots, \hat{a}^{k_m}$ in the domain of $\mathcal{I}$ so that $\mathcal{I}$ does contain the required assetions, yet ψ^ℓ is not satisfied with the assignment τ (the other direction is symmetrical). If ψ^ℓ is not true for the assignment τ, then the r-tuple $\tau(z_{h_1}, \dots, z_{h_r})$ is not in the interpretation of P over $\mathcal{M}$. By the construction of $\mathcal{M}$ and the properties deduced from $\mathcal{I}$, it must be the case that the following facts hold in $\mathcal{I}$:

$$R_1^{i_1}(c, \bar{a}^{i_1}), R_2^{i_2}(\bar{a}^{i_1}, \bar{a}^{i_2}), \dots, R_r^{i_r}(\bar{a}^{i_{r-1}}, \bar{a}^{i_r}),$$
$$V(\bar{a}^{i_1}, a^{i_1}), \dots, V(\bar{a}^{i_r}, a^{i_r}), TV_R(\bar{a}^{i_r}, 0)$$

for some elements $\bar{a}^{i_1}, \dots, \bar{a}^{i_r}$ in $\mathcal{I}$, and such that for each h_k, $1 \leq k \leq r$, we have that $\tau(z_{h_k}) = a^{h_k}$. From this one can easily check that there is a homomorphism from the subquery ϕ^ℓ to the aforementioned atoms, which results in extra evaluations from the core of q to $\mathcal{I}$.

Inductive step. For the inductive case, assume that $\psi^\ell = \psi^{\ell_1} \vee \psi^{\ell_2}$, $q < \ell \leq p$ (the case when $\psi^\ell = \neg\psi^{\ell_1}$ is analogous). Moreover, assume for the sake of contradiction that there are constants $\hat{a}^{k_1}, \dots, \hat{a}^{k_m}$ in $\mathcal{I}$ so that the $\mathcal{I}$ does contain the facts in the statement of the claim, yet ψ^ℓ is not satisfied with the assignment τ (the other direction is symmetrical). If ψ^ℓ is not satisfied, then neither ψ^{ℓ_1} nor ψ^{ℓ_2} are satisfied. By the induction hypothesis and the construction of $\mathcal{M}$, there are constants $\hat{a}_1^{k_1}, \dots, \hat{a}_1^{k_m}$ and $\hat{a}_2^{k_1}, \dots, \hat{a}_2^{k_m}$ such that the facts

$$F_1^{\sigma(1)}(c, \hat{a}_1^{k_1}), F_2^{\sigma(2)}(\hat{a}_1^{k_1}, \hat{a}_1^{k_2}), \dots, F_m^{\sigma(m)}(\hat{a}_1^{k_{m-1}}, \hat{a}_1^{k_m}),$$
$$V(\hat{a}_1^{k_1}, a^{\sigma(1)}), \dots, V(\hat{a}_1^{k_m}, a^{\sigma(m)}), TV_F^\ell(\hat{a}_1^{k_m}, 0),$$

and

$$F_1^{\sigma(1)}(c, \hat{a}_2^{k_1}), F_2^{\sigma(2)}(\hat{a}_2^{k_1}, \hat{a}_2^{k_2}), \dots, F_m^{\sigma(m)}(\hat{a}_2^{k_{m-1}}, \hat{a}_2^{k_m}),$$
$$V(\hat{a}_2^{k_1}, a^{\sigma(1)}), \dots, V(\hat{a}_2^{k_m}, a^{\sigma(m)}), TV_F^\ell(\hat{a}_2^{k_m}, 0)$$

belong to $\mathcal{I}$. In other words, these facts cause an extra witness for the subquery ϕ^ℓ in q. □

Having established the claim above, we recall that the body of the query q contains the atoms $TV_F^p(s_2, s_3) \wedge C(s_3)$. Thus, if the sub-formula ψ^p which is ψ itself by the convention, does not hold for some assignment τ, then $\mathcal{I}$ contains the fact $TV_F^p(\hat{a}, 0)$ for some constant $\hat{a}$, and then at least one extra evaluation can be constructed from the core of q to $\mathcal{I}$. Thus, it must be the case that ψ^p is satisfied by all possible assignments of $y_1, \dots, y_m$ to $a^1, \dots, a^n$. It follows that $\exists \mathbf{x}\, \forall \mathbf{y}\, \psi(\mathbf{x}, \mathbf{y})$ is satisfiable.

($\Leftarrow$) Assume that $\exists \mathbf{x}\, \forall \mathbf{y}\, \psi(\mathbf{x}, \mathbf{y})$ is satisfiable. Then, as we have mentioned, there is a model $\mathcal{M}$ with at most n elements, that satisfies this formula. Assume without loss of generality that this model has exactly n elements, say $a_1, \dots, a_n$, and the satisfying assignment for ψ assigns each x_i, $1 \leq i \leq n$, to a^i. We now construct an interpretation $\mathcal{I}$, which is a model of the KB $\mathcal{K}$, such that there are only two evaluations from the core of q to $\mathcal{I}$. Define the model $\mathcal{I}$ as follows.

- The interpretation of each concept A^i, $1 \leq i \leq n$, and S in $\mathcal{I}$ is as required by the ABox $\mathcal{A}$: each A^i contains a^i and S contains 1 and 0.
- The interpretation of each role R^i_k, $1 \leq i \leq n$, $1 \leq k \leq r$, and R is as in the $\mathcal{A}$, plus extra pairs as in the canonical model in of $\mathcal{K}$. It is convenient to note that the canonical model contains a tree of R's of height r and width n. Each of the leaves of this tree has to be in the domain (i.e. the first component) of the interpretation of TV_R. We add the following to $\mathcal{I}$. For each path from c to some leaf $\bar{a}^{i_r}$ in $\mathcal{I}$ of the form

  $$R_0(c), R_1^{i_1}(c, \bar{a}^{i_1}), R_2^{i_2}(\bar{a}^{i_1}, \bar{a}^{i_2}), \ldots, R_r^{i_r}(\bar{a}^{i_{r-1}}, \bar{a}^{i_r}),$$

  such that $\bar{a}^{i_1}, \ldots, \bar{a}^{i_r}$ the model $\mathcal{I}$ contains the assertions

  $$V^{i_1}(\bar{a}^{i_1}, a^{i_1}), \ldots, V^{i_r}(\bar{a}^{i_r}, a^{i_r}).$$

  Also, the interpretation of TV_R in $\mathcal{I}$ contains the pair $(\bar{a}^{i_r}, 1)$ if the interpretation of the relation P in the model $\mathcal{M}$ contains the tuple $(a^{i_1}, \ldots, a^{i_r})$, and $(\bar{a}^{i_r}, 0)$ otherwise.
- The interpretation of each role F^i_k, $1 \leq i \leq n$, $1 \leq k \leq m$, and F in $\mathcal{I}$ is as in the $\mathcal{A}$, plus extra pairs as in the canonical model of $\mathcal{K}$. Again, it is convenient to note that the canonical model contains a tree of F's of height m and width n. Each of the leaves of this tree has to be in the domain (the first component) of the interpretation of TV_F^ℓ for all $1 \leq \ell \leq p$. We add the following to $\mathcal{I}$. For each path from c to some leaf $\hat{a}^{i_m}$ in $\mathcal{I}$, of form

  $$F_0(c), F_1^{i_1}(c, \hat{a}^{i_1}), F_2^{i_2}(\hat{a}^{i_1}, \hat{a}^{i_2}), \ldots, F_m^{i_m}(\hat{a}^{i_{m-1}}, \hat{a}^{i_m}),$$

  such that $\hat{a}^{i_1}, \ldots, \hat{a}^{i_m}$ are different constants the model $\mathcal{I}$ contains the following assertions

  $$V^{i_1}(\hat{a}^{i_1}, a^{i_1}), \ldots, V^{i_m}(\hat{a}^{i_m}, a^{i_r}).$$

  Also, the interpretation of each TV_F^ℓ in $\mathcal{I}$ contains the pair $(\hat{a}^{i_m}, 1)$ if the sub-formula ψ^ℓ of ψ holds in the model $\mathcal{M}$ for the evaluation of $y_1, \ldots, y_m$ to $a^{i_1}, \ldots, a^{i_m}$, and $(\hat{a}^{i_m}, 0)$ otherwise.

It is now a cumbersome, but straightforward task to show that there are only two evaluations from the core of q to $\mathcal{I}$. The proof of course makes use of the fact that any other evaluation must map at least one variable of q to a constant that does not appear in the active domain of $\mathcal{K}$.

The only thing left to complete the proof is to explain how to extend the construction above in the case when the vocabulary of ψ contains several relations $P_1, \ldots, P_N$. Essentially, one needs a *tree like* construct as above for each such relation. The KB can then be adapted in the expected way. Note that if the arities of these relations are not the same then the lengths of these trees need to be adapted accordingly. □

Lemma 6. *There exists a Π_2^p-algorithm which solves the problem DL-Lite$_{core}$ Cntd-*Aggregate Certain Answers.

Proof. The combined complexity of the algorithm from the proof of Lem. 4 is exponential, since neighbourhoods $\mathcal{N}_q(d)$ can be of exponential size (if the TBox is not fixed), and the number k is not a fixed constant any more. Next we show how to redefine these neighbourhoods to have them polynomial in size while keeping the possibility of merging them without increasing the number $\bar{n}$. Having this fact proved, we show the correctness of the following algorithm for some constant number k': check all interpretations $\mathcal{J}$ over number of elements $|\mathbb{D}|^{k'}$, whether $\mathcal{J} \models \mathcal{K}$ and $\mathcal{J} \models q(\mathbf{t}, n)$. This algorithm clearly runs in Π_2^p, since checking whether $\mathcal{J} \models \mathcal{K}$ and $\mathcal{J} \models q(\mathbf{t}, n)$ can be done in NP (because n is bounded by $|\mathbb{D}| + |\mathcal{T}|$, as in the proof of Lem. 4).

Note, that the construction below works only for *DL-Lite$_{core}$*, and increasing the expressivity to *DL-Lite$_{\mathcal{R}}$* leads to an increase in the complexity of the problem.

For every pair of variables u, v from the body $\phi(\mathbf{x}, \mathbf{y}, z)$ of the input *Cntd*-ACQ q let $\mathcal{L}_q(u, v)$ be the subset of all atoms in $\phi(\mathbf{x}, \mathbf{y}, z)$ which use only variables on simple (possibly undirected) paths from u to v over roles of $\mathcal{K}$.

Consider the model $\hat{\mathcal{I}}$ of $\mathcal{K}$ and the set of domain elements $\mathbb{D}^*$, built on the base of a witnessing model $\mathcal{I}$, as in the proof of Lem. 4. For every $d \in \mathbb{D}^{\hat{\mathcal{I}}} \backslash \mathbb{D}^*$ define the $*$-*neighbourhood* $\mathcal{N}_q^*(d)$ as a sub-interpretation of $\mathbb{D}^{\hat{\mathcal{I}}}$ induced by all elements d' such that there exists $u, v \in \mathbf{x} \cup \mathbf{y} \cup z$ and a homomorphism h from $\mathcal{L}_q(u, v)$ to $\mathbb{D}^{\hat{\mathcal{I}}}$ such that $h(u) = d$, $h(v) = d'$ and $h(w) \notin \mathbb{D}^*$ for all $w \neq v$.

Since every element $d_1 \in \mathbb{D}^{\hat{\mathcal{I}}} \backslash \mathbb{D}^*$ and every role (atomic, or its inversion) R have at most one d_2 such that $\hat{\mathcal{I}} \models R(d_1, d_2)$, every pair u, v induces at most one element in every $\mathcal{N}_q^*(d)$.[3] Hence the $*$-neighbourhood $\mathcal{N}_q^*(d)$ is of polynomial size, and there are only polynomial number of such $*$-neighbourhoods which are not isomorphic by the definition in the the proof of Lem. 4). However, merging d_1 and d_2 in $\hat{\mathcal{I}}$ such that $\mathcal{N}_q^*(d_1) \sim \mathcal{N}_q^*(d_2)$ and with the same distance from $\mathbb{D}$ modulo $|q| + 1$, does not create new images of z by evaluations from the core of q, so it does not increase the number $\bar{n}$.

The construction above implies that there exists a constant k' as required in the beginning of the proof. □

Lemma 7. *The problem DL-Lite$_{core}$ Cntd-*Aggregate Certain Answers *is Π_2^p-hard.*

Proof. The hardness is established by a reduction from ∀∃ 3-SAT, which is the problem of verifying, given a Boolean formula ψ in 3-CNF with variables partitioned into tuples $\mathbf{x}$ and $\mathbf{z}$, whether it is true that for every truth assignment of the variables $\mathbf{x}$, there exists a truth assignment of the variables $\mathbf{z}$ so that ψ is satisfied with the overall assignment. This problem is well known to be Π_2^p-complete.

[3] This is the argument which is not valid for *DL-Lite$_{\mathcal{R}}$*.

Let ψ be such a formula of the form $\forall \mathbf{x}\, \exists \mathbf{z} \bigwedge_{1 \leq k \leq \ell} \psi_k$, where each ψ_k ($1 \leq k \leq \ell$) is a clause containing exactly three literals. We denote the variables of each ψ_k from $\mathbf{x} \cup \mathbf{z}$ by y_k^1, y_k^2 and y_k^3. Let also $\mathbf{x} = x_1, \ldots, x_n$ and $\mathbf{z} = z_1, \ldots, z_m$. Based on ψ, we show how to construct in polynomial time a *DL-Lite*$_{core}$ knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ and a Boolean *Cntd*-ACQ q such that $3 \geq m(q, \mathbf{t}_\emptyset, \mathcal{K})$ (where $\mathbf{t}_\emptyset$ is the empty tuple), if and only if for every truth assignment of the variables in $\mathbf{x}$, there exists a truth assignment of the variables in $\mathbf{z}$ so that ψ is satisfied with the overall assignment.

Let us begin by defining the query. Consider roles R, V, S_1, S_2, S_3 and concepts C_k^1, C_k^2, C_k^3 for each $1 \leq k \leq \ell$. The *Cntd*-ACQ is

$$q(Cntd(u)) \text{ :- } \exists x_1 \cdots \exists x_n\ \exists z_1 \cdots \exists z_m\ \exists c_1 \cdots \exists c_\ell\ \exists v_{x_1} \cdots \exists v_{x_n}\ \exists v_{z_1} \cdots \exists v_{z_m}\ \exists s\ \phi,$$

where

$$\begin{aligned} \phi = {} & V(s,u) \wedge \\ & \bigwedge_{1 \leq k \leq \ell} \Big(R(s,c_k) \wedge S_1(c_k, v_{y_k^1}) \wedge S_2(c_k, v_{y_k^2}) \wedge S_3(c_k, v_{y_k^3}) \wedge C_k^1(y_k^1) \wedge C_k^2(y_k^2) \wedge C_k^3(y_k^3) \Big) \wedge \\ & \qquad V(x_1, v_{x_1}) \wedge \cdots \wedge V(x_n, v_{x_n}) \wedge V(z_1, v_{z_1}) \wedge \cdots \wedge V(z_m, v_{z_m}). \end{aligned}$$

Next we define the knowledge base $\mathcal{K}$ and start with the ABox $\mathcal{A}$.

1. The active domain contains the constants $\hat{x}_1, \ldots, \hat{x}_n$, $\hat{z}_1, \ldots, \hat{z}_m$, $\hat{c}_1, \ldots, \hat{c}_\ell$, $\hat{v}_{x_1}, \ldots, \hat{v}_{x_n}$, $\hat{v}_{z_1}, \ldots,$ $\hat{v}_{z_m}$, and $\hat{s}$. The ABox $\mathcal{A}$ contains assertions which are copies of all atoms of the formula ϕ, except $V(s,u)$, in the way that every existential variable a (except u) of q is "frozen" into the constant $\hat{a}$.
2. The active domain contains constants 0 and 1. The ABox $\mathcal{A}$ contains the assertions $V(\hat{s}, 0)$ and $V(\hat{s}, 1)$.
3. The active domain contains constants $\bar{x}_1, \ldots, \bar{x}_n$. For each $1 \leq i \leq n$ and $1 \leq k \leq \ell$ the ABox $\mathcal{A}$ contains the assertion $R(\bar{x}_i, \hat{c}_k)$. For each $1 \leq i \leq n$ it also contains the assertion $X_i(\bar{x}_i)$, where X_i is a new concept assigned to the constant $\bar{x}_i$.
4. The active domain contains constants $\bar{z}_1, \ldots, \bar{z}_m$. For each $1 \leq j \leq m$ the ABox $\mathcal{A}$ contains the assertions $V(\bar{z}_j, 0)$ and $V(\bar{z}_j, 1)$.
5. For every $1 \leq k \leq \ell$ the ABox $\mathcal{A}$ contains the assertions $C_k^1(\bar{y}_k^1)$, $C_k^2(\bar{y}_k^2)$ and $C_k^3(\bar{y}_k^3)$ where $\bar{y}_k^1, \bar{y}_k^2$ and $\bar{y}_k^3$ are the constants among $\bar{x}_1, \ldots, \bar{x}_n, \bar{z}_1, \ldots, \bar{z}_m$ such that y_k^1, y_k^2 and y_k^3 are the variables of the clause ψ_k in the formula ψ.
6. For each $1 \leq k \leq \ell$ and $1 \leq p \leq 7$ the active domain contains a constant $\bar{c}_k^p$. For each $1 \leq k \leq \ell$, let $\sigma_1, \ldots, \sigma_7$ be an enumeration of all satisfying assignments of the clause ψ_k, that uses variables y_k^1, y_k^2, y_k^3. For each such assignment σ_p, $1 \leq p \leq 7$, the ABox $\mathcal{A}$ contains assertions $S_1(\bar{c}_k^p, \sigma_p(y_k^1))$, $S_2(\bar{c}_k^p, \sigma_p(y_k^2))$ and $S_3(\bar{c}_k^p, \sigma_p(y_k^3))$. Here we abuse notation and assume that $\sigma_p(y_k^i)$, $1 \leq i \leq 3$, evaluates to either the constant 0 or the constant 1.
7. The active domain contains constants d_1 and d_2. The ABox $\mathcal{A}$ contains assertions $V(d_1, d_2)$ and $R(d_1, \bar{c}_k^p)$ for each $1 \leq k \leq \ell$ and $1 \leq p \leq 7$.

Finally, the TBox $\mathcal{T}$ contains the assertion $X_i \sqsubseteq \exists V$ for each $1 \leq i \leq n$.

Next we show that $3 \in Cert(q, \mathbf{t}_\emptyset, \mathcal{K})$, where $\mathbf{t}_\emptyset$ is the empty tuple, holds if and only if for every truth assignment of the variables $\mathbf{x}$, there exists a truth assignment of the variables $\mathbf{z}$ so that ψ is satisfied with the overall assignment.

($\Rightarrow$) Let $3 \in Cert(q, \mathbf{t}_\emptyset, \mathcal{K})$, but assume for the sake of contradiction that there is a truth assignment σ_x for the variables in $\mathbf{x}$ such that ψ is not satisfiable under any assignment for the variables in $\mathbf{z}$.

Construct the following interpretation $\mathcal{I}$ for $\mathcal{K}$.

1. The interpretation of all roles and concepts except for V corresponds precisely to $\mathcal{A}$. That is, for each role R different from V, we have that $(a, b) \in R^{\mathcal{I}}$ if and only if $R(a, b)$ is an assertion in $\mathcal{A}$, and likewise for all concepts C.
2. For each $1 \leq i \leq n$, the pair $(\bar{x}_i, \sigma_x(\bar{x}_i))$ belongs to $V^{\mathcal{I}}$. Note that here we abuse terminology once again, the value of each $\sigma_x(\bar{x}_i)$ is either the constant 1 or the constant 0 in the active domain.

It is clear that $\mathcal{I}$ is a model of $\mathcal{K}$. From the assumption that $3 \geq m(q, \mathbf{t}, \mathcal{K})$ and the construction of $\mathcal{I}$, it must be the case that $\bar{q}(d_2)$ holds in $\mathcal{I}$ (recall, that $\bar{q}$ is the core of the *Cntd*-ACQ q). From this fact it is possible to derive a contradiction as follows. Let d be a constant in $\mathcal{I}$ such that $\bar{q}(d)$, $\bar{q}(0)$ and $\bar{q}(1)$ hold in $\mathcal{I}$ (we know from the construction of $\mathcal{K}$ that $\bar{q}(0)$ and $\bar{q}(1)$ must already hold in all models of $\mathcal{K}$). But again from the interpretation of V in $\mathcal{I}$ we know that d can only be the constant d_2, since the remainder of the pairs in V have either 0 or 1 in the range (second) position. It follows that there must be a mapping τ from the variables of ϕ to elements in $\mathcal{I}$ such that

- the mapping τ sends the variable u to the constant d_2, and
- $\bar{q}(\tau(u))$ holds in $\mathcal{I}$.

We can, however, determine more properties of τ from the construction of $\mathcal{I}$ and $\mathcal{K}$. In fact, it follows that each variable x_i, $1 \leq i \leq n$, is indeed mapped by τ to the constant $\bar{x}_i$ in $\mathcal{I}$, and that v_{x_i} is mapped to the corresponding valuation of x_i according to σ_x. It follows from the construction of ϕ that the following assignment σ_z of the variables $\mathbf{z}$ in ψ is such that σ_x, σ_z satisfy ψ: σ_z assigns the value 1 to z_j iff the variable v_{z_j} in ϕ is mapped to 1, according to τ.

($\Leftarrow$) Assume that for every truth assignment of the variables $\mathbf{x}$, there exists a truth assignment of the variables $\mathbf{z}$ so that each ψ_k is satisfied with the overall assignment, yet assume for the sake of contradiction that there is a model $\mathcal{I}$ of $\mathcal{K}$ such that only $\bar{q}(1)$ and $\bar{q}(0)$ hold in $\mathcal{I}$ (those hold by the construction of $\mathcal{K}$).

From the construction of $\mathcal{A}$, it is not difficult to see that, for each pair $(\bar{x}_i, a)$ in $V^{\mathcal{I}}$ it must be the case that a is either 1 or 0 (otherwise it violates the assumption previously mentioned, since this would give an extra witness for the variable u). Construct the following valuation σ_x for the variables in $\mathbf{x}$. For each $1 \leq i \leq n$, $\sigma_x(x_i) = 1$, if the pair $(\bar{x}_i, 1)$ is in $V^{\mathcal{I}}$, and $\sigma_x(x_i) = 0$ otherwise.

From the original assumption, there must be an assignment σ_z of the variables in $\mathbf{z}$ such that σ_x, σ_z satisfy ψ. We now show that $\bar{q}(d_2)$ must hold in $\mathcal{I}$. To that extent, construct the following mapping τ from the variables in q to constants in $\mathcal{I}$: τ maps every variable $x_1, \ldots, x_n$ and $z_1, \ldots, z_m$ to the corresponding constants $\bar{x}_1, \ldots, \bar{x}_n$ and $\bar{z}_1, \ldots, \bar{z}_m$ in $\mathcal{I}$, and maps each v_{x_i} (and v_{z_j}) to 1 if and only if σ_x (and σ_z) assigns the value 1 to x_i (and z_j). Moreover, for each clause ψ_k, it maps each variable c_k to the corresponding constant $\bar{c}_k^p$ such that the p-th satisfying assignment for ψ_k is the one witnessed by σ_x and σ_z. It is then clear that $\bar{q}(d_2)$ hold in $\mathcal{I}$, which violates our original assumption. □

Lemma 8. *The decision problem DL-Lite$_\mathcal{R}$ Cntd-*Aggregate Certain Answers *is* coNExpTime*-hard.*

Proof. Same to the proof of Lem. 5, this proof is by reduction from the complement of the satisfiability problem for Boolean FO formulas in the Bernays-Schönfinkel class.

Let $\exists \mathbf{x}\, \forall \mathbf{y}\, \psi(\mathbf{x}, \mathbf{y})$ be a formula in the Bernays-Schönfinkel class. Let also $\mathbf{x} = x_1, \ldots, x_n$ and $\mathbf{y} = y_1, \ldots, y_m$. For the sake of readability, we again assume that ψ mentions a single relation symbol P of arity r and also that $r \geq m$. The adaptation to the general case can be done exactly the same way as in the proof of Lem. 5. Finally, w.l.o.g. we assume that ψ is not atomic and let $\psi^1, \ldots, \psi^p$ be an enumeration of all the sub-formulas of ψ such that $\psi^p = \psi$, and $\psi^1, \ldots, \psi^{p_0}$ is the set of atomic sub-formulas of ψ. It is also important to recall the following property of the Bernays-Schönfinkel class: either a formula in it is unsatisfiable, or it has a model with at most n elements (see, e.g. [17]).

Next we will show how to construct in polynomial time a *DL-Lite*$_\mathcal{R}$ KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ and a Boolean *Cntd*-ACQ $q(\mathit{Cntd}(z))$ such that $n{+}2 \in \mathit{Cert}(q, \mathbf{t}_\emptyset, \mathcal{K})$ (where $\mathbf{t}_\emptyset$ is the empty tuple) iff $\exists \mathbf{x}\, \forall \mathbf{y}\, \psi(\mathbf{x}, \mathbf{y})$ is satisfiable, i.e. has a model.

The underlying idea of the reduction is the same that of the proof of Lem. 5. The main difference, however, is that we are now dealing with *Cntd*-ACQs, and thus we need to identify just one counting variable in the query that needs to witness each required evaluation, whereas in Lem. 5 we had the flexibility to allow these evaluations to match each model in different ways.

In other words, this means the following. Recall that in the proof of Lem. 5 we built a knowledge base such that its models represented different possibilities for instances of the vocabulary of the FO formula $\exists \mathbf{x}\, \forall \mathbf{y}\, \psi(\mathbf{x}, \mathbf{y})$. For this reduction we will need, in our query, a single *control* part of the query that needs to match different parts of these knowledge bases, to check whether these models represent well formed evaluations for ψ or not.

The control is coded into the ABox $\mathcal{A}$, which description we now begin.

Let the active domain contains the constants

$$\begin{array}{ccc} d, c_R, c_F, c_1, \ldots, c_p, c_{p+1}, & b_0, \ldots, b_p, & s_R^{-r+2}, \ldots, s_R^0, \\ s^{-r+2}, \ldots, s_F^0, \ldots, s_F^{r-n}, & e_R, e_F, & 0, 1, \end{array}$$

the constants e_ℓ^1, e_ℓ^2 for each number ℓ, $1 \leq \ell \leq p$, such that ψ^ℓ is either atomic or of the form $\neg\psi_{\ell_1}$, and the constants e_ℓ^1, e_ℓ^2, e_ℓ^3, e_ℓ^4 for each number ℓ, $1 \leq \ell \leq p$, such that ψ^ℓ is of the form $\psi_{\ell_1} \vee \psi_{\ell_2}$.

Consider concepts $CNTR$ and $START$, and roles A_1, A_2, A_3, B_1, B_2, B_3, B_4, $D_1, \dots, D_{p_0}$, and C. Let the ABox $\mathcal{A}$ contains the following assertions:

- $CNTR(c_R)$, $CNTR(c_F)$, $CNTR(d)$, and $CNTR(c_1), \dots, CNTR(c_{p+1})$;
- $C(c_R, e_R), C(c_F, e_F)$ and $C(c_{p+1}, e_{p+1})$;
- $C(c_\ell, e_\ell^1)$ and $C(c_\ell, e_\ell^2)$, for each $1 \leq \ell \leq p$ such that ψ^ℓ is either atomic or of the form $\neg\psi_{\ell_1}$;
- $C(c_\ell, e_\ell^1)$, $C(c_\ell, e_\ell^2)$, $C(c_\ell, e_\ell^3)$, and $C(c_\ell, e_\ell^4)$, for each $p_0 < \ell \leq p$ such that ψ^ℓ is of the form $\psi^\ell = \psi_{\ell_1} \vee \psi_{\ell_2}$.

The intuition for the role C will be clear when we describe our query, but it is part of the *control* for the query. Let the ABox $\mathcal{A}$ also contains assertions

- $A_1(c_R, d), A_2(c_R, d), A_1(c_F, d)$, $A_2(c_F, d)$ and $A_1(c_{p+1}, d), A_2(c_{p+1}, d)$;
- $A_1(c_\ell, d)$ and $A_2(c_\ell, d)$ for each $1 \leq \ell \leq p_0$ (i.e. such that ψ^ℓ is atomic);
- $A_1(c_\ell, c_{\ell_1})$ and $A_2(c_\ell, d)$ for each $p_0 < \ell \leq p$ such that $\psi^\ell = \neg\psi_{\ell_1}$;
- $A_1(c_\ell, c_{\ell_1})$ and $A_2(c_\ell, c_{\ell_2})$ for each $p_0 < \ell \leq p$ such that $\psi^\ell = \psi_{\ell_1} \vee \psi_{\ell_2}$;
- $A_3(c, d)$ for each $c \in \{c_R, c_F, c_1, \dots, c_p\}$;
- $A_3(c_{p+1}, c_{p+1})$.

Here the constant d will play the role of a *dummy constant.* In other words, it will indicate when certain parts of q are to be activated. The ABox $\mathcal{A}$ also contains the assertions

- $B_1(e_R, d)$, $B_2(e_R, d)$, $B_3(e_R, d)$, and $B_4(e_R, d)$;
- $B_1(e_F, d)$, $B_2(e_F, d)$, $B_3(e_F, d)$, and $B_4(e_F, d)$;
- $B_1(e_{p+1}, d)$, $B_2(e_{p+1}, d)$, $B_3(e_{p+1}, d)$, and $B_4(e_{p+1}, d)$;
- $B_1(c_\ell^1, 1), B_2(c_\ell^1, 0), B_3(c_\ell^1, d)$, $B_4(c_\ell^1, d)$, and
  $B_1(c_\ell^2, 0), B_2(c_\ell^2, 1), B_3(c_\ell^2, d)$, $B_4(c_\ell^2, d)$ for each $1 \leq \ell \leq p_0$ (i.e. such that ψ^ℓ is atomic);
- $B_2(c_\ell^1, 1), B_3(c_\ell^1, 1), B_1(c_\ell^1, d)$, $B_4(c_\ell^1, d)$, and
  $B_2(c_\ell^2, 0), B_3(c_\ell^2, 0), B_1(c_\ell^2, d)$, $B_4(c_\ell^2, d)$ for each $p_0 < \ell \leq p$ such that $\psi^\ell = \neg\psi_{\ell_1}$;
- $B_1(c_\ell^1, d)$, $B_2(c_\ell^1, 1), B_3(c_\ell^1, 0)$, $B_4(c_\ell^1, 0)$,
  $B_1(c_\ell^2, d)$, $B_2(c_\ell^2, 0), B_3(c_\ell^2, 1)$, $B_4(c_\ell^2, 0)$,
  $B_1(c_\ell^3, d)$, $B_2(c_\ell^3, 0), B_3(c_\ell^3, 0)$, $B_4(c_\ell^3, 1)$, and
  $B_1(c_\ell^4, d)$, $B_2(c_\ell^4, 0), B_3(c_\ell^4, 0)$, $B_4(c_\ell^4, 0)$, for each $p_0 < \ell \leq p$ such that $\psi^\ell = \psi_{\ell_1} \vee \psi_{\ell_2}$.

At this point it is useful to describe the intuition behind the control of the query. Elements that belong to $CNTR$ determine which part of the query in Lem. 5 are they going to simulate. Our query shall contain an atom of the form $CNTR(c)$. When the variable c is mapped to c_R or c_F, the query simulates the part that assigns elements and truth values to the relations and atomic subformulas of ψ. When c is mapped to some c_ℓ, $1 \leq \ell \leq p$, it checks instead that the assignment for the i-th subformula is consistent. Finally, when mapping c to c_{p+1} we check the satisfiability of the formula. The dummy constant d can be replaced in different parts of the query when the control does not need this query. This is regulated with the help of the roles A_i, B_i, C_i and D_i.

Next, the ABox $\mathcal{A}$ contains the following assertions

- $D_1(c, d), \ldots, D_{p_0}(c, d)$ for each $e \in \{c_R, c_F, c_{p^{at}+1}, \ldots, c_{p+1}\}$;
- $D_j(c_\ell, d)$ and $D_\ell(c_\ell, c_R)$, for each $1 \leq \ell \leq p_0$ (i.e. such that ψ^ℓ is atomic) and for each $j \neq \ell$.

Next we need to add to $\mathcal{A}$ some assertions that help create dummy witnesses for our evaluations. These are comprised by the following assertions, where R_0, R, TV and V are fresh roles:

- $R_0(b_0, d), R(b_0, b_1), R(b_1, b_2), \ldots, R(b_{p-1}, b_p)$;
- $V(b_i, a)$ for each $a \in \{a_1, \ldots, a_n, 0, 1\}$ and $1 \leq i \leq p$;
- $TV(b_p, a)$ for each $a \in \{a_1, \ldots, a_n, 0, 1\}$, plus $TV(b_p, \bot)$.

Here $\bot$ is another fresh constant. The query q is constructed below in such a way that there is an evaluation assigning this constant if and only if ψ is satisfiable.

Finally, we add to $\mathcal{A}$ the following necessary assertions so that each model correctly represents instances for the vocabulary of ψ (here we use roles similar to the $\mathcal{A}$ in the proof of Lem. 5):

- $START_R(s_R^0)$ and $START_F(s_F^{r-n})$;
- $R(s_R^{-r+2}, s_R^{-r+3}), \ldots, R(s_R^{-1}, s_R^0)$ and $R(s_F^{-r+2}, s_F^{-r+3}), \ldots, R(s_F^{-1}, s_F^0)$;
- $R(s_F^0, s_F^1), \ldots, R(s_F^{r-n-1}, s_F^{r-n})$;
- $R_0(s_R^i, c_R)$ for each $-r+2 \leq i \leq 0$ and $F_0(s_F^i, c_F)$ for each $-r+2 \leq i \leq 0$;
- $F_0(s_F^0, c)$ for each $c \in \{c_1, \ldots, c_p\}$.

The remaining assertions in $\mathcal{A}$ are similar to the proof of Lem. 5, where A^i is a fresh concept for each $1 \leq i \leq n$ and S and Z also fresh concepts:

- $A^i(a_i)$ for each $1 \leq i \leq n$;
- $S(0)$, $S(1)$, and $Z(0)$.

The TBox $\mathcal{T}$ is similar to that of the proof of Lem. 5, and divided into three parts.

1. The first part essentially assigns elements to the existential variables $\mathbf{x}$. The TBox $\mathcal{T}$ contains inclusions
   - $\exists (V^i)^- \sqsubseteq A^i$ and $V^i \sqsubseteq V$ for each i, $1 \leq i \leq n$,
   - $TV \sqsubseteq V$.
2. The second part essentially assigns a truth value to every fact $P(a^{i_1}, \ldots, a^{i_r})$. The TBox $\mathcal{T}$ contains inclusions
   - $START_R \sqsubseteq \exists R_1^i$ for each i, $1 \leq i \leq n$;
   - $\exists R_k^{i\,-} \sqsubseteq R_{k+1}^j$ for each i, j and k, $1 \leq i \leq n$, $1 \leq j \leq n$, $1 \leq k \leq r-1$;

- $\exists {R^i_k}^- \sqsubseteq \exists V^i$ and $R^i_k \sqsubseteq R$ for each i and k, $1 \leq i \leq n$, $1 \leq k \leq r$;
- $\exists {R^i_r}^- \sqsubseteq \exists TV_R$ for each i, $1 \leq i \leq n$;
- $\exists TV_R^- \sqsubseteq S$.

3. The third part essentially assigns a truth value to every sub-formula ψ_ℓ of ψ for every assignment of variables $\mathbf{y}$. Note that we also use role R in this part. The TBox $\mathcal{T}$ contains inclusions
   - $START_F \sqsubseteq \exists F^i_1$ for each i, $1 \leq i \leq n$;
   - $\exists {F^i_k}^- \sqsubseteq F^j_{k+1}$ for each i, j and k, $1 \leq i \leq n$, $1 \leq j \leq n$, $1 \leq k \leq m-1$;
   - $\exists {F^i_k}^- \sqsubseteq \exists V^i$ and $F^i_k \sqsubseteq R$ for each i and k, $1 \leq i \leq n$, $1 \leq k \leq m$;
   - $\exists {F^i_m}^- \sqsubseteq \exists TV^\ell_F$ for each i and ℓ, $1 \leq i \leq n$, $1 \leq \ell \leq p$;
   - $\exists {TV^\ell_F}^- \sqsubseteq S$ for each ℓ, $1 \leq \ell \leq p$.

We now turn to define the Boolean count distinct ACQ $q(Cntd(z))$. Consider the following parametrized conjunctions

$$\begin{aligned}\gamma^R[v, w_1, \ldots, w_r, w] = R_0(u_0, v) \wedge R(u_0, u_1) \wedge R(u_1, u_2) \wedge \cdots \wedge R(u_{r-1}, u_r) \wedge \\ V(u_1, w_1) \wedge \cdots \wedge V(u_r, w_r) \wedge TV(u_r, w),\end{aligned}$$

and, for each $1 \leq \ell \leq p$,

$$\begin{aligned}\gamma^F_\ell[v, w_1, \ldots, w_m, w] = R_0(u_0, v) \wedge R(u_0, u_1) \wedge R(u_1, u_2) \wedge \cdots \wedge R(u_{m-1}, u_m) \wedge \\ V(u_1, w_1) \wedge \cdots \wedge V(u_m, w_m) \wedge TV^\ell(u_m, w).\end{aligned}$$

In the following definition of q, we shall consider various instantiations of these conjunctions, and we always assume that all the variables u_i are instantiated each time by different, fresh variables.

As the first example of such instantiations, consider variables $\alpha_1, \ldots, \alpha_\ell$. For each atomic sub-formula $\psi^\ell = P(z_1, \ldots, z_r)$,over variables $z_1, \ldots, z_r$ from $\mathbf{x} \cup \mathbf{y}$, we define

$$\phi^\ell = \gamma^R[\alpha_\ell, \tau(z)_1, \ldots, \tau(z)_r, w_1]$$

where τ is a mapping that maps each x_i to variable s_i and each y_j to variable t_j.

The ACQ $q(Cntd(z))$ is defined as

$$\exists \mathbf{s}\, CNTR(c) \wedge \gamma^R[c, v_1, \ldots, v_r, z] \wedge \xi_1 \wedge \xi_2 \wedge \xi_3 \wedge \xi_4,$$

where $\mathbf{s}$ is the tuple of all variables in q except z, and

$$\begin{aligned}\xi_1 = A^1(s^1) \wedge A^2(s^2) \wedge \ldots \wedge A^n(s^n) \wedge \\ D_1(c, \alpha_1) \wedge \ldots \wedge D_{p_0}(c, c_{p_0}) \wedge \\ \phi^1 \wedge \ldots \wedge \phi^{p_0} \wedge \\ \gamma^F_\ell[c, t_1, \ldots, t_m, w_2] \wedge C(c, e) \wedge B_1(e, w_1) \wedge B_2(e, w_2)\end{aligned}$$

(this conjunction ensures that the truth values of all p_0 atomic sub-formulas are consistent);

$$\xi_2 = \gamma^F_{\ell_1}[c', t_1, \ldots, t_m, w_3] \wedge A_1(c, c') \wedge B_3(e, w_3)$$

(this conjunction ensures the correct assignment of all sub-formulas ψ^{ℓ} of form $\neg\psi_{\ell_1}$);

$$\xi_3 = \gamma^F_{\ell_2}[c'', t_1, \ldots, t_m, w_4] \wedge A_2(c, c'') \wedge B_4(e, w_4)$$

(this conjunction assigns truth values of sub-formulas $\psi^{\ell} = \psi^{\ell_1} \vee \psi^{\ell_2}$);

$$\xi_4 = \gamma^F_p[c''', t_1, \ldots, t_m, w_5] \wedge Z(w_5) \wedge A_3(c, c''')$$

(and this conjunction verifies that all truths assignments do not satisfy ψ).

One can now show, by combining the techniques of the proofs of Lem. 5 and 7, that $n+2 \in Cert(q, \mathbf{t}_{\emptyset}, \mathcal{K})$ iff $\exists\mathbf{x}\,\forall\mathbf{y}\,\psi(\mathbf{x}, \mathbf{y})$ is satisfiable, i.e. has a model. □