On the Power and Limitations of Examples for Description Logic Concepts

Balder ten Cate[1] , **Raoul Koudijs**[2] , **Ana Ozaki**[2,3]

[1] Institute for Logic, Language and Computation (ILLC), University of Amsterdam
[2] University of Bergen
[3] University of Oslo
b.d.tencate@uva.nl, raoul.koudijs@uib.no, anaoz@uio.no

Abstract

Labeled examples (i.e., positive and negative examples) are an attractive medium for communicating complex concepts. They are useful for deriving concept expressions (such as in concept learning, interactive concept specification, and concept refinement) as well as for illustrating concept expressions to a user or domain expert. We investigate the power of labeled examples for describing description-logic concepts. Specifically, we systematically study the existence and efficient computability of *finite characterisations*, i.e., finite sets of labeled examples that uniquely characterize a single concept, for a wide variety of description logics between $\mathcal{EL}$ and $\mathcal{ALCQI}$,both without an ontology and in the presence of a DL-Lite ontology. Finite characterisations are relevant for debugging purposes, and their existence is a necessary condition for exact learnability with membership queries.

1 Introduction

Labeled examples (i.e., positive and negative examples) are an attractive medium for communicating complex concepts. They are useful as data for deriving concept expressions (such as in concept learning, interactive concept specification, and example-driven concept refinement) as well as for illustrating concept expressions to a user or domain expert [Lehmann, 2009; Funk *et al.*, 2019; Funk *et al.*, 2021; Ozaki, 2020; Fanizzi *et al.*, 2008; Iannone *et al.*, 2007; Lehmann and Hitzler, 2010]. Here, we study the utility of labelled examples for characterising description logic concepts, where examples are finite interpretations that can be either positively or negatively labelled.

Example 1.1. In the description logic $\mathcal{EL}$, we may define the concept of an *e-bike* by means of the concept expression

$$C \quad := \quad \text{Bicycle} \sqcap \exists \text{Contains.Battery}$$

Suppose we wish to illustrate C by a collection of positive and negative examples. What would be a good choice of examples? Take the interpretation $\mathcal{I}$ consisting of the following facts.
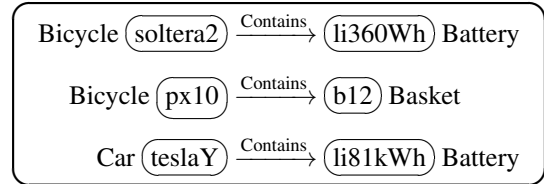
In the context of this interpretation $\mathcal{I}$, it is clear that

- *soltera2* is a positive example for C, and
- *px10* and *teslaY* are negative examples for C

In fact, as it turns out, C is the *only* $\mathcal{EL}$-concept (up to equivalence) that fits these three labeled examples. In other words, these three labeled examples "uniquely characterize" C within the class of all $\mathcal{EL}$-concepts. This shows that the above three labeled examples are a good choice of examples. Adding any additional examples would be redundant. Note, however, that this depends on the choice of description logic. For instance, the richer concept language $\mathcal{ALC}$ allows for other concept expressions such as Bicycle $\sqcap \neg\exists$Contains.Basket that also fit.

Motivated by the above example, we investigate the existence and efficient computability of *finite characterisations*, i.e., finite sets of labeled examples that uniquely characterize a single concept. Finite characterisations are relevant not only for *illustrating* a complex concept to a user (e.g., to verify the correctness of a concept expression obtained using machine learning techniques). Their existence is a necessary condition for *exact learnability with membership queries* [Angluin, 1987]. Furthermore, from a more fundamental point of view, by studying the existence of finite characterisations, we gain insight into the power and limitations of labeled examples as a medium for describing concepts.

Concretely, we systematically study the existence of, and the complexity of computing, finite characterisations for description logic concepts in a wide range of expressive description logics. We look at concept languages $\mathcal{L}(\mathbf{O})$ that are generated by a set of connectives $\mathbf{O}$, where $\{\exists, \sqcap\} \subseteq \mathbf{O} \subseteq \{\forall, \exists, \geq, -, \sqcap, \sqcup, \top, \bot, \neg\}$. In other words, we look at fragments of the description logic $\mathcal{ALCQI}$ that contain at least the $\exists$ and $\sqcap$ constructors from $\mathcal{EL}$.

As our first result, within this framework, we obtain an almost complete classification of the concept languages that admit finite characterisations.

Theorem 1.1. *Let* $\{\exists, \sqcap\} \subseteq \mathbf{O} \subseteq \{\forall, \exists, \geq, -, \sqcap, \sqcup, \top, \bot, \neg\}$.
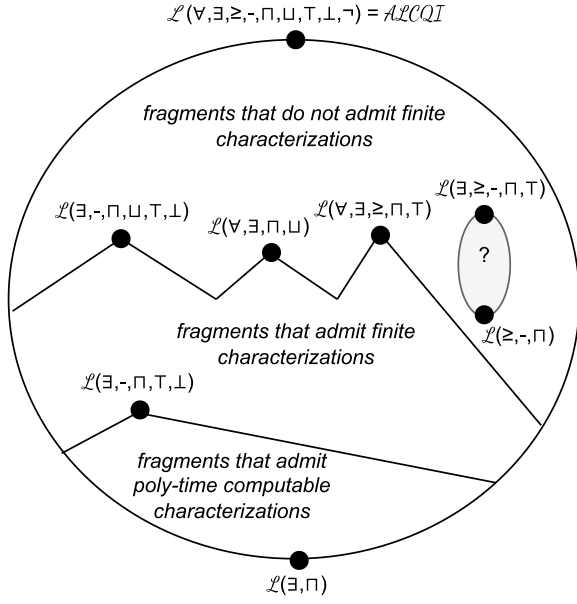
$$\mathcal{L}(\forall,\exists,\geq,-,\sqcap,\sqcup,\top,\bot,\neg) = \mathcal{ALCQI}$$

fragments that do not admit finite characterizations

$\mathcal{L}(\exists,-,\sqcap,\sqcup,\top,\bot)$ $\mathcal{L}(\forall,\exists,\sqcap,\sqcup)$ $\mathcal{L}(\forall,\exists,\geq,\sqcap,\top)$ $\mathcal{L}(\exists,\geq,-,\sqcap,\top)$

?

$\mathcal{L}(\geq,-,\sqcap)$

fragments that admit finite characterizations

$\mathcal{L}(\exists,-,\sqcap,\top,\bot)$

fragments that admit poly-time computable characterizations

$\mathcal{L}(\exists,\sqcap)$

Figure 1: Summary of Thm. 1.1 and Thm. 1.2.

1. If **O** is a subset of $\{\exists,-,\sqcap,\sqcup,\top,\bot\}$, $\{\forall,\exists,\sqcup,\sqcap\}$ or $\{\forall,\exists,\geq,\sqcap,\top\}$ then $\mathcal{L}(\mathbf{O})$ admits finite characterisations.

2. Otherwise $\mathcal{L}(\mathbf{O})$ does not admit finite characterisations, except possibly if $\{\geq,-,\sqcap\} \subseteq \mathbf{O} \subseteq \{\exists,\geq,-,\sqcap,\top\}$.

The above theorem identifies three maximal fragments of $\mathcal{ALCQI}$ that admit finite characterisations. It leaves open the status of essentially only two concept languages, namely $\mathcal{L}(\exists,\geq,-,\sqcap)$ and $\mathcal{L}(\exists,\geq,-,\sqcap,\top)$ (note that $\exists$ is definable in terms of $\geq$). The proof builds on prior results from [Cate and Dalmau, 2022] and [Cate and Koudijs, 2024]. Our main novel technical contributions are a construction of finite characterisations for $\mathcal{L}(\forall,\exists,\geq,\sqcap,\top)$ and complementary negative results for $\mathcal{L}(\geq,\bot), \mathcal{L}(\geq,\sqcup)$ and $\mathcal{L}(\forall,\exists,-,\sqcap)$. The construction of finite characterisations for $\mathcal{L}(\forall,\exists,\geq,\sqcap,\top)$ is non-elementary. We give an elementary (doubly exponential) construction for $\mathcal{L}(\exists,\geq,\sqcap,\top)$, via a novel polynomial-time algorithm for constructing frontiers.

Next, we study which concept languages $\mathcal{L}(\mathbf{O})$ admit *polynomial-time computable characterisations*, i.e., have a polynomial-time algorithm to compute finite characterisations.

Theorem 1.2. *Let* $\{\exists,\sqcap\} \subseteq \mathbf{O} \subseteq \{\forall,\exists,\geq,-,\sqcap,\sqcup,\top,\bot,\neg\}$.

1. If **O** is a subset of $\{\exists,-,\sqcap,\top,\bot\}$, then $\mathcal{L}(\mathbf{O})$ admits polynomial-time computable characterisations.

2. Otherwise, $\mathcal{L}(\mathbf{O})$ does not admit polynomial-time computable characterisations, assuming P$\neq$NP.

The first item follows from known results [Cate and Dalmau, 2022]; we prove the second. Thm. 1.1 and 1.2 are summarized in Figure 1.

Finally, we investigate finite characterisations relative to an ontology $\mathcal{O}$, where we require the examples to be interpretations that satisfy the ontology and we require that every fitting concept is equivalent to the input concept C relative to $\mathcal{O}$.

Theorem 1.3. *Let* $\{\exists,\sqcap\} \subseteq \mathbf{O} \subseteq \{\forall,\exists,\geq,-,\sqcap,\sqcup,\top,\bot,\neg\}$.

1. If **O** is a subset of $\{\exists,\sqcap,\top,\bot\}$, then $\mathcal{L}(\mathbf{O})$ admits finite characterisations w.r.t. all DL-Lite ontologies.

2. Otherwise, $\mathcal{L}(\mathbf{O})$ does not admit finite characterisations w.r.t. all DL-Lite ontologies, except possibly if $\{\exists,\sqcap,\sqcup\} \subseteq \mathbf{O} \subseteq \{\exists,\sqcap,\sqcup,\top,\bot\}$.

In fact, for $\mathcal{L}(\exists,\sqcap,\top,\bot)$-*concepts* C *and DL-Lite ontologies* $\mathcal{O}$ *such that* C *is satisfiable w.r.t.* $\mathcal{O}$, *a finite characterisation can be computed in polynomial time.*

This shows in particular that, if we exclude $\sqcup$ from consideration, then $\mathcal{L}(\exists,\sqcap,\top,\bot)$ is the unique largest fragment admitting finite characterisations under DL-Lite ontologies.

Outline In Section 2, we review relevant definitions and we introduce our framework for finite characterisations. Section 3 studies the existence and polynomial-time computability of finite characterisations without an ontology. Section 4 extends the analysis to the case with DL-Lite ontologies. Section 5 discusses further implications and future directions.

Related Work Finite characterisations were first studied in the computational learning theory literature under the name of teaching sets, with a corresponding notion of teaching dimension, measuring the maximal size of minimal teaching sets of some class of concepts [Goldman and Kearns, 1995]. The existence of finite characterisations is a necessary condition for exact learnability with membership queries.

Several recent works study finite characterisations for *description logic concepts* ([Cate and Dalmau, 2022; Funk *et al.*, 2022] for $\mathcal{ELI}$; [Fortin *et al.*, 2022] for temporal instance queries). We use results from [Cate and Dalmau, 2022; Funk *et al.*, 2022]. Thm. 1.3 appears to contradict a result in [Funk *et al.*, 2022], which states that $\mathcal{ELI}$ admits finite characterisations under DL-Lite ontologies. However, this is due to a difference in the way we define examples, which we will discuss in detail in Section 2.

Finite characterisations have also been studied for a while in the *data management* community (cf. [Mannila and Räihä, 1985; Staworko and Wieczorek, 2015; Cate and Dalmau, 2022; Cate *et al.*, 2024] for queries; [Alexe *et al.*, 2011] for schema mappings), and, a systematic study of finite characterisations for syntactic fragments of *modal logic* was carried out in [Cate and Koudijs, 2024]. We make use of several results from [Cate and Koudijs, 2024], and one of our results also implies an answer to an open problem from this paper.

2 Basic Definitions and Framework

In the following, we first define the syntax and semantics of concept and ontology languages considered in this work.

Concept Languages Let $\mathsf{N_C}$ and $\mathsf{N_R}$ be infinite and mutually disjoint sets of *concept* and *role* names, respectively. We work with finite subsets $\Sigma_\mathsf{C} \subseteq \mathsf{N_C}, \Sigma_\mathsf{R} \subseteq \mathsf{N_R}$ of these. We denote by $\mathcal{L}(\forall,\exists,\geq,-,\sqcap,\sqcup,\top,\bot,\neg)[\Sigma_\mathsf{C},\Sigma_\mathsf{R}]$ the following concept language:

$$C, D ::= A \mid \top \mid \bot \mid (C \sqcap D) \mid (C \sqcup D) \mid$$
$$\neg C \mid (\exists S.C) \mid (\forall S.C) \mid (\geq kS.C)$$

where $k \geq 1$, $A \in \Sigma_{\mathsf{C}}$ and $S = R$ or $S = R^-$ for some $R \in \Sigma_{\mathsf{R}}$. Note that $\exists S.C$ is equivalent to $\geq 1S.C$.

For any set of operators $\mathbf{O} \subseteq \{\forall, \exists, \geq, -, \sqcap, \sqcup, \top, \bot, \neg\}$, we denote by $\mathcal{L}(\mathbf{O})[\Sigma_{\mathsf{C}}, \Sigma_{\mathsf{R}}]$ the fragment of $\mathcal{L}(\forall, \exists, \geq, -, \sqcap, \sqcup, \top, \bot, \neg)$ that only uses the constructs in $\mathbf{O}$, concept names from Σ_{C} and role names from Σ_{R}. We may omit $[\Sigma_{\mathsf{C}}, \Sigma_{\mathsf{R}}]$ from $\mathcal{L}(\mathbf{O})[\Sigma_{\mathsf{C}}, \Sigma_{\mathsf{R}}]$ and simply write $\mathcal{L}(\mathbf{O})$ if the role and concepts names are clear from context or irrelevant.

Observe that $\mathcal{L}(\forall, \exists, \geq, -, \sqcap, \sqcup, \top, \bot, \neg)$ is a notational variant of $\mathcal{ALCQI}$, while, e.g., $\mathcal{L}(\exists, \sqcap, \top)$ corresponds to $\mathcal{EL}$ and $\mathcal{L}(\forall, \exists, \sqcap, \top, \bot)$ to $\mathcal{ALE}$. This notation thus allows us to state results for a large variety of languages for concepts.

Ontology Language   We consider the very popular DL-Lite ontology language [Artale *et al.*, 2009]. DL-Lite concept inclusions (CIs) are expressions of the form $B \sqsubseteq C$, respectively, where B, C are concept expressions built through the grammar rules (we write below $\exists S$ as a shorthand for $\exists S.\top$)

$$B ::= A \mid \exists R \mid \exists R^-, \quad C ::= B \mid \neg B,$$

with $R \in \mathsf{N}_{\mathsf{R}}$ and $A \in \mathsf{N}_{\mathsf{C}}$. We call concepts of the form B above *basic concepts*. A DL-Lite *ontology* is a finite set of DL-Lite CIs. Note that every DL-Lite ontology is satisfiable.

Semantics   An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ w.r.t. a signature $(\Sigma_{\mathsf{C}}, \Sigma_{\mathsf{R}})$ is a structure, in the traditional model-theoretic sense. It consists of a non-empty set $\Delta^{\mathcal{I}}$, called the *domain*, and a function $\cdot^{\mathcal{I}}$ that assigns a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ of the domain to each $A \in \Sigma_{\mathsf{C}}$ and a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ over the domain to each $R \in \Sigma_{\mathsf{R}}$. We extend $(\cdot)^{\mathcal{I}}$ to concept and role expressions as follows.

$$(\top)^{\mathcal{I}} := \Delta^{\mathcal{I}} \quad (\bot)^{\mathcal{I}} := \emptyset \quad (C \sqcap D)^{\mathcal{I}} := C^{\mathcal{I}} \cap D^{\mathcal{I}}$$

$$(\neg C)^{\mathcal{I}} := \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \qquad (C \sqcup D)^{\mathcal{I}} := C^{\mathcal{I}} \cup D^{\mathcal{I}}$$

$$(\exists S.C)^{\mathcal{I}} := \{d \mid \exists e \in \Delta^{\mathcal{I}} \text{ s.t. } (d, e) \in S^{\mathcal{I}} \text{ and } e \in C^{\mathcal{I}}\}$$

$$(\geq kS.C)^{\mathcal{I}} := \{d \mid \exists e_1, \dots, e_k \in \Delta^{\mathcal{I}} \text{ pairwise-distinct s.t.}$$
$$(d, e_i) \in S^{\mathcal{I}} \text{ and } e_i \in C^{\mathcal{I}} \text{ for all } 1 \leq i \leq k\}$$

$$(\forall S.C)^{\mathcal{I}} := \{d \mid \forall e \in \Delta^{\mathcal{I}} \text{ if } (d, e) \in S^{\mathcal{I}} \text{ then } e \in C^{\mathcal{I}}\}$$

$$(R^-)^{\mathcal{I}} := \{(e, d) \mid (d, e) \in R^{\mathcal{I}}\}.$$

We say that an interpretation $\mathcal{I}$ satisfies a CI $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. We write $\mathcal{I} \models \alpha$ if $\mathcal{I}$ satisfies a CI α. Moreover, we say that $\mathcal{I}$ satisfies an ontology $\mathcal{O}$, written $\mathcal{I} \models \mathcal{O}$, if $\mathcal{I} \models \alpha$ for all CIs α in $\mathcal{O}$. A concept C is *satisfiable* w.r.t. an ontology $\mathcal{O}$ (or vice-versa) if there is an interpretation $\mathcal{I} \models \mathcal{O}$ with $C^{\mathcal{I}} \neq \emptyset$; a concept C *subsumes* D relative to $\mathcal{O}$ (denoted $\mathcal{O} \models C \sqsubseteq D$) if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all interpretations $\mathcal{I} \models \mathcal{O}$; and two concepts C, D are *equivalent* relative to $\mathcal{O}$ if $C^{\mathcal{I}} = D^{\mathcal{I}}$ for all interpretations $\mathcal{I} \models \mathcal{O}$. We use $\models C \sqsubseteq D$ as a shorthand for $\emptyset \models C \sqsubseteq D$ when indicating that a subsumption holds relative to the empty ontology.

We also write $C \equiv D$ as a shorthand for $C \equiv_{\emptyset} D$.

A *pointed interpretation* is a pair $(\mathcal{I}, d)$ where $\mathcal{I}$ is an interpretation and $d \in \Delta^{\mathcal{I}}$ is a domain element. We say that a pointed interpretation $(\mathcal{I}, d)$ *satisfies* a concept C if $d \in C^{\mathcal{I}}$. A pointed interpretation $(\mathcal{I}, d)$ is *finite* if $\Delta^{\mathcal{I}}$ is a finite set.

Theorem 2.1. *(Finite Controllability) Let $C, D \in \mathcal{L}(\forall, \exists, \geq, -, \sqcap, \sqcup, \top, \bot, \neg)$ and let $\mathcal{O}$ be a DL-Lite ontology. Then*

$C \not\equiv_{\mathcal{O}} D$ *iff there is a finite pointed interpretation $(\mathcal{I}, d)$ with $\mathcal{I} \models \mathcal{O}$ and $d \in ((C \sqcap \neg D) \sqcup (\neg C \sqcap D))^{\mathcal{I}}$.*

Labeled Examples and Finite characterisations   An *example* is a finite pointed interpretation $(\mathcal{I}, d)$, labelled either positively (with a "+") or negatively (with a "−"). We say that an example $(\mathcal{I}, d)$ is a *positive example* for a concept C if $d \in C^{\mathcal{I}}$, and otherwise $(\mathcal{I}, d)$ is a *negative example* for C. We may omit "for C" when this is clear from the context. We will use the notation $E = (E^+, E^-)$ to denote a finite collection of examples labeled as positive and negative. A concept C *fits* a set of positively and negatively labelled examples if it is satisfied on all the positively labelled examples and not satisfies on any of the negatively labelled examples.[1]

A *finite characterisation* for a concept C w.r.t. concept language $\mathcal{L}$ is a finite collection of labeled examples $E = (E^+, E^-)$ such that C fits E and every concept $D \in \mathcal{L}$ that fits E is equivalent to C. A concept language $\mathcal{L}$ *admits finite characterisations* if for every pair of finite sets $\Sigma_{\mathsf{C}} \subseteq \mathsf{N}_{\mathsf{C}}, \Sigma_{\mathsf{R}} \subseteq \mathsf{N}_{\mathsf{R}}$, every concept $C \in \mathcal{L}[\Sigma_{\mathsf{C}}, \Sigma_{\mathsf{R}}]$ has a finite characterisation w.r.t. $\mathcal{L}[\Sigma_{\mathsf{C}}, \Sigma_{\mathsf{R}}]$. We say that $\mathcal{L}$ *admits polynomial-time computable characterisations* if, furthermore, there is an algorithm that, given an input concept $C \in \mathcal{L}[\Sigma_{\mathsf{C}}, \Sigma_{\mathsf{R}}]$ outputs a finite characterisation $E = (E^+, E^-)$ of C w.r.t. $\mathcal{L}[\Sigma_{\mathsf{C}}, \Sigma_{\mathsf{R}}]$ in time polynomial in the size of the concept $|C|$ and the size of the signature $|\Sigma_{\mathsf{C}}| + |\Sigma_{\mathsf{R}}|$. This implies that E^+ and E^- consist of polynomially many, polynomial-size examples. Therefore, if $\mathcal{L}(\mathbf{O})$ admits polynomial time computable characterisations it has polynomial *teaching size* in the sense of [Telle *et al.*, 2019]. Admitting finite or polynomial time computable characterisations is a monotone property of concept classes.

We defined examples as *finite* pointed interpretations, because they are intended as objects that can be communicated to a user and/or specified by a user. This is justified by Thm. 2.1, which shows that any two non-equivalent concepts can be distinguished by a finite pointed interpretation.

It may not be clear to the reader why the above definition of "admitting finite characterisations" involves a quantification over vocabularies. The following example explains this.

Example 2.1. Consider the concept $C = \exists R.A$. Let $E = (E^+, E^-)$ with $E^+ = \{(\mathcal{I}, d_1)\}$ and $E^- = \{(\mathcal{I}, d_3)\}$, where $\mathcal{I}$ is the following interpretation.
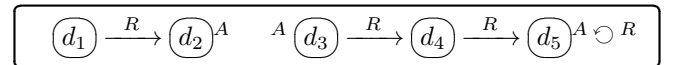
Clearly, C fits E. In fact, it can be shown that C is the only $\mathcal{L}(\exists, \sqcap, \sqcup)$-concept (up to equivalence) that fits E and that uses only the concept name A and role name R. In other words, E is a finite characterisation of C w.r.t. $\mathcal{L}(\exists, \sqcap, \sqcup)[\Sigma_{\mathsf{C}}, \Sigma_{\mathsf{R}}]$ with $\Sigma_{\mathsf{C}} = \{A\}$ and $\Sigma_{\mathsf{R}} = \{R\}$. In contrast, E does *not* uniquely characterize C with respect to $\mathcal{L}(\exists, \sqcap, \sqcup)[\Sigma'_{\mathsf{C}}, \Sigma_{\mathsf{R}}]$ with $\Sigma'_{\mathsf{C}} = \{A, B\}$ since $\exists R.(A \sqcup B)$ fits E and is not equivalent to C. Therefore, what qualifies as a finite characterisations depends on the choice of vocabulary.

---

[1]Equivalently, a concept C fits a set of positively and negatively labelled examples if all the positively labelled examples are positive examples for C, and all the negatively labelled examples are negative examples for C.

The notions of an *example* and of a *finite characterisation* naturally extend to the case with an ontology $\mathcal{O}$, where we now require that each example satisfies $\mathcal{O}$, and we replace *concept equivalence* by *equivalence relative to* $\mathcal{O}$. More precisely, an *example for* $\mathcal{O}$ is a finite pointed interpretation $(\mathcal{I}, d)$ with $\mathcal{I} \models \mathcal{O}$, labelled either positively or negatively. A *finite characterisation* for a concept C w.r.t. a concept language $\mathcal{L}$ *under* $\mathcal{O}$ is a finite collection $E = (E^+, E^-)$ of examples for $\mathcal{O}$ such that for every $D \in \mathcal{L}$ that fits E, $C \equiv_{\mathcal{O}} D$. A concept language $\mathcal{L}$ *admits finite characterisations under* $\mathcal{L}'$ *ontologies* if for every $\mathcal{O}$ in $\mathcal{L}'$ and every pair of finite sets $\Sigma_{\mathsf{C}} \subseteq \mathsf{N}_{\mathsf{C}}, \Sigma_{\mathsf{R}} \subseteq \mathsf{N}_{\mathsf{R}}$, every concept $C \in \mathcal{L}[\Sigma_{\mathsf{C}}, \Sigma_{\mathsf{R}}]$ has a finite characterisation w.r.t. $\mathcal{L}[\Sigma_{\mathsf{C}}, \Sigma_{\mathsf{R}}]$ under $\mathcal{O}$.

Example 2.2. Consider the concept language $\mathcal{L}(\sqcap, \sqcup, \neg)$ and concept names $\Sigma_{\mathsf{C}} = \{\mathsf{Animal}, \mathsf{Cat}, \mathsf{Dog}, \mathsf{Red}\}$. Note that there are no role constructs in the language. Let $\mathcal{O}$ be the DL-Lite ontology that expresses that Cat and Dog are disjoint subconcepts of Animal. Now consider the concept $C = \mathsf{Cat} \sqcap \mathsf{Red}$. Let $\mathcal{I}$ be the interpretation with

- $\Delta^{\mathcal{I}} = \{a, a', b, b', c, c', d, d'\}$,
- $\mathsf{Animal}^{\mathcal{I}} = \{a, a', c, c', d, d'\}$,
- $\mathsf{Cat}^{\mathcal{I}} = \{c, c'\}$, $\mathsf{Dog}^{\mathcal{I}} = \{d, d'\}$, $\mathsf{Red}^{\mathcal{I}} = \{a, b, c, d\}$,

which satisfies $\mathcal{O}$. It can be shown that the collection of labeled examples (E^+, E^-) with $E^+ = \{(\mathcal{I}, c)\}$ and $E^- = \{(\mathcal{I}, a), (\mathcal{I}, a'), (\mathcal{I}, b), (\mathcal{I}, b'), (\mathcal{I}, c'), (\mathcal{I}, d), (\mathcal{I}, d')\}$ is a finite characterisation of C w.r.t. $\mathcal{L}(\sqcap, \sqcup, \neg)$ under $\mathcal{O}$. On the other hand, the same set of examples is not a finite characterisation under the empty ontology, because $\mathsf{Cat} \sqcap \mathsf{Red} \sqcap \neg \mathsf{Dog}$ also fits but is not equivalent to C in the absence of the ontology. Thus, in this case, the ontology helps to reduce the number of examples needed to uniquely characterize C. Moreover, the ontology can rule out unnatural examples, since in this case the characterisation would need to include a negative example satisfying $\mathsf{Cat} \sqcap \mathsf{Red} \sqcap \mathsf{Dog}$.

Remark 2.2. We have defined an example as a pointed finite interpretation $(\mathcal{I}, d)$ and we call such an example a *positive example* for C if $d \in C^{\mathcal{I}}$ and a *negative example* otherwise. In some of the prior literature on learning description logic concepts, examples were instead defined as pointed ABoxes $(\mathcal{A}, a)$ [Funk *et al.*, 2022; Konev *et al.*, 2016; Ozaki *et al.*, 2020] [2], while e.g. [Fortin *et al.*, 2022] also work with finite interpretations (in their case, linear orders) as examples. Such a pointed ABox is then considered a *positive example* for C under an ontology $\mathcal{O}$ if a is a certain answer for C in $\mathcal{A}$ under $\mathcal{O}$ (i.e., $a^{\mathcal{I}} \in C^{\mathcal{I}}$ for all interpretations with $\mathcal{I} \models \mathcal{O}$ and $\mathcal{I} \models \mathcal{A}$), and a *negative example* otherwise. Let us call the latter type of examples "ABox-examples". Consider the concepts $C = \forall R.P$ and $C' = \forall R.Q$. Although they are clearly not equivalent, it is not possible to distinguish them by means of ABox-examples (under, say, an empty ontology), for the simple reason that there does not exist a positive ABox-example for either of these concepts. This motivates

our decision to define examples as pointed interpretations. This is tied to the fact that we study concept languages that include universal quantification and/or negation. Hence, for concept languages contained in $\mathcal{L}(\exists, \geq, \sqcap, \sqcup, \top, \bot)$, such as $\mathcal{EL}$, there is essentially no difference between finite characterisations using ABox-examples and finite characterisations using interpretation-examples in the absence of an ontology.

However, there are some further repercussions for having interpretations instead of ABoxes as examples in the case with ontologies, even for such weaker concept languages. Intuitively, in the presence of a DL-Lite$^{\mathcal{H}}$-ontology $\mathcal{O}$, any ABox consistent with $\mathcal{O}$ can be viewed as a succinct representation of a possibly infinite interpretation (i.e. its *chase*). Conversely, a finite interpretation can always be conceived of as an ABox. The difference shows up as follows: it was shown in [Funk *et al.*, 2022] that $\mathcal{L}(\exists, \sqcap, -)$ admits finite characterisations under DL-Lite$^{\mathcal{H}}$ ontologies in the setting where examples are pointed ABoxes. We see in Section 4 below that the same does not hold in our setting.

Size of Concepts and Examples For any concept $C \in \mathcal{L}(\forall, \exists, \geq, -, \sqcap, \sqcup, \top, \bot)$, we write $|C|$ for the number of symbols in the syntactic expression C. The numbers occurring in C are assumed to be represented using a binary encoding. All our lower bounds also hold when numbers are represented in unary. Let $\mathsf{nr}(C)$ be the the largest natural number occurring in C. Further, let $\mathsf{dp}(C)$ be the *role depth* of C, i.e. the longest sequence of nested quantifiers and/or $\geq$ operators in C. Finally, for interpretations $\mathcal{I}$ we set $|\mathcal{I}| := |\Delta^{\mathcal{I}}|$, and the size of an example $(\mathcal{I}, d)$ is simply $|\mathcal{I}|$.

3 Finite Characterisations without Ontology

Our aim, in this section, is to prove Thm. 1.1 and 1.2. We start by listing relevant known results.

Theorem 3.1. *The following results are known (cf. [Cate and Koudijs, 2024]).*

1. *$\mathcal{L}(\exists, -, \sqcap, \top, \bot)$ admits polynomial-time computable characterisations (from [Cate and Dalmau, 2022]).*

2. *$\mathcal{L}(\exists, -, \sqcap, \sqcup, \top, \bot)$ admits finite characterisations. However, finite characterisation are necessarily of exponential size (from [Alexe* et al.*, 2011]).*

3. *$\mathcal{L}(\forall, \exists, \sqcap, \sqcup)$ admits finite characterisations. However, finite characterisation are necessarily of non-elementary size (from [Cate and Koudijs, 2024]).*

4. *Neither $\mathcal{L}(\forall, \exists, \sqcap, \bot)$ nor $\mathcal{L}(\forall, \exists, \sqcup, \top)$ admit finite characterisations (from [Cate and Koudijs, 2024]).*

To establish Thm. 1.1, it remains to show that $\mathcal{L}(\forall, \exists, \geq, \sqcap, \top)$ admits finite characterisations and that $\mathcal{L}(\geq, \bot), \mathcal{L}(\geq, \sqcup)$ and $\mathcal{L}(\forall, \exists, -, \sqcap)$ do not admit finite characterisations.

3.1 Finite Characterisations for $\mathcal{L}(\forall, \exists, \geq, \sqcap, \top)$

The next results involve a method for constructing examples that forces fitting concepts to have a bounded role depth.

Proposition 3.2. *Fix finite sets $\Sigma_C \subseteq \mathsf{N}_{\mathsf{C}}$ and $\Sigma_R \subseteq \mathsf{N}_{\mathsf{R}}$. For all $n, k \geq 0$, there is a set of examples $E^+_{n,k}$, such that for all $\mathcal{L}(\forall, \exists, \geq, \sqcap, \top)[\Sigma_C, \Sigma_R]$ concept expressions C, the following are equivalent:*

---

[2]By a pointed ABox $(\mathcal{A}, a)$ we mean a finite set $\mathcal{A}$ of assertions of the form $A(b)$ and $R(b, c)$ with A a concept name, R a role name, and b, c individual names (over the given signature), together with an individual name a occurring in some assertion in $\mathcal{A}$.

1. C is equivalent to a concept expression of role depth at most n and maximum number restriction k;

2. C fits the positive examples $E_{n,k}^+$.

We illustrate the result for the special case where $n = 3$ and $k = 2$. It suffices to choose as positive examples the examples $(\mathcal{I}, d_1)$ and $(\mathcal{I}', d_1)$ based on the following interpretations:
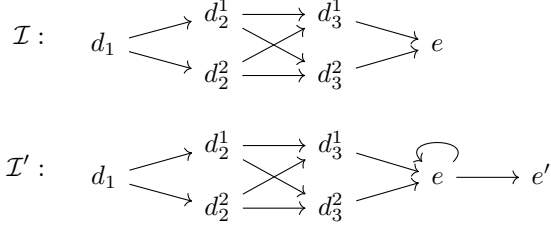
where an arrow $d \to d'$ means that the pair (d, d') belong to the interpretation of every role in Σ_R; and every element except e' belongs to the interpretation of every atomic concept in Σ_C. It can be shown that every concept fitting $(\mathcal{I}, e)$ and $(\mathcal{I}', e)$ as positive is equivalent to a depth 0 concept. It follows by backward induction that an $\mathcal{L}(\forall, \exists, \geq, \sqcap, \top)[\Sigma_C, \Sigma_R]$-concept fits $(\mathcal{I}, d_1)$ and $(\mathcal{I}', d_1)$ as positive examples iff it has role depth at most 3 and maximum number restriction 2.

As a consequence, we obtain the following.

Theorem 3.3. $\mathcal{L}(\forall, \exists, \geq, \sqcap, \top)$ *admits finite characterisations.*

We show this by appending to $E_{n,k}^+$ as many positive and negative examples needed to distinguish a concept from all the finitely many other concepts with depth at most n and number restriction at most k. Hence, we obtain a non-elementary upper bound on the size of finite characterisations for $\mathcal{L}(\forall, \exists, \geq, \sqcap, \top)$, since there are already $\text{tower}(n)$ (i.e. an iterated power $2^{\cdot^{\cdot^{\cdot^2}}}$ of height n) many pairwise non-equivalent $\mathcal{L}(\exists, \sqcap, \top)$ concepts of role depth n. We do not have a matching lower bound for the above language, and it remains open whether there are more efficient approaches. In the next subsection, we show that the fragment $\mathcal{L}(\exists, \geq, \sqcap, \top)$ *does* admit an elementary (doubly exponential) construction of finite characterisations.

3.2 An Elementary Construction of Finite Characterisations for $\mathcal{L}(\exists, \geq, \sqcap, \top)$

Since $\exists$ is expressible as ≥ 1, we restrict our attention to $\mathcal{L}(\geq, \sqcap, \top)$ in what follows.

Theorem 3.4. *There is a polynomial time algorithm for testing if $\models C \sqsubseteq C'$, where $C, C' \in \mathcal{L}(\geq, \sqcap, \top)$.*

The above theorem can be proved by showing that entailments between conjunctions of $\mathcal{L}(\geq, \sqcap, \top)$-concepts can be reduced to entailments between the individual conjuncts.

Next, we will use Theorem 3.4 to show that concepts in $\mathcal{L}(\geq, \sqcap, \top)$ can be brought into a certain normal form in polynomial time. Recall that every $\mathcal{L}(\geq, \sqcap, \top)$-concept expression C that is not a concept name is a conjunction $C_1 \sqcap \cdots \sqcap C_n$ ($n \geq 0$) where each C_i is either an concept name or is of the form $\geq kR.C'$ where C' is again a $\mathcal{L}(\geq, \sqcap, \top)$-concept expression. Here we conveniently view $\top$ as a shorthand for an empty conjunction. We say that such a concept expression C is *irredundant* if it does not contain a conjunction that includes distinct conjuncts C_i, C_j with $\models C_i \sqsubseteq C_j$.

Proposition 3.5. *Every $\mathcal{L}(\geq, \sqcap, \top)$-concept C is equivalent to an irredundant $\mathcal{L}(\geq, \sqcap, \top)$-concept, which can be computed in polynomial time.*

The following two lemmas, intuitively, tell us that, every pointed interpretation satisfying a concept $C \in \mathcal{L}(\geq, \sqcap, \top)$ contains a small pointed sub-interpretation that already satisfies C and falsifies all $\mathcal{L}(\geq, \sqcap, \top)$-concepts falsified in the original interpretation. We write $\mathcal{I} \subseteq \mathcal{I}'$ to denote that $\mathcal{I}$ is a sub-interpretation of $\mathcal{I}'$ (that is, $\Delta^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}'}$ and $X^{\mathcal{I}} \subseteq X^{\mathcal{I}'}$ for all $X \in \Sigma_C \cup \Sigma_R$).

Lemma 3.6. *For every concept $C \in \mathcal{L}(\geq, \sqcap, \top)$, interpretation $\mathcal{I}$, and $d \in C^{\mathcal{I}}$, there is an $\mathcal{I}' \subseteq \mathcal{I}$ with $d \in C^{\mathcal{I}'}$ and $|\mathcal{I}'| \leq |C|^{|C|}$.*

Lem. 3.6 pairs well with the next lemma, showing that $\mathcal{L}(\geq, \sqcap, \top)$-concepts are $\subseteq$-monotone.

Lemma 3.7. *For all concepts $C \in \mathcal{L}(\geq, \sqcap, \top)$ and interpretations $\mathcal{I} \subseteq \mathcal{I}'$, it holds that $C^{\mathcal{I}} \subseteq C^{\mathcal{I}'}$.*

This can be proved by a straightforward induction argument, or, alternatively, follows from the Łos-Tarski theorem. Together these prove the following theorem.

Theorem 3.8. *For every two concepts $C, D \in \mathcal{L}(\geq, \sqcap, \top)$ such that $\not\models C \sqsubseteq D$, there is some pointed interpretation $(\mathcal{I}, d)$ of size $|\mathcal{I}| \leq |C|^{|C|}$ such that $d \in (C \sqcap \neg D)^{\mathcal{I}}$.*

It follows from this theorem that, in order to construct a finite characterization of an $\mathcal{L}(\geq, \sqcap, \top)$-concept, it suffices to pick as our positive examples all positive examples of size at most $|C|^{|C|}$. It only remains to explain how we construct the negative examples. This is done through a novel *frontier* construction construction for $\mathcal{L}(\exists, \geq, \sqcap, \top)$. Intuitively, a frontier for C is finite set of concepts that are strictly weaker than C and that separates C from all concepts strictly weaker than C (cf. Figure 2 for a graphical depiction).

Definition 3.9. *(Frontier) Given a concept language $\mathcal{L}$ and a concept $C \in \mathcal{L}$, a frontier for C w.r.t. $\mathcal{L}$ is a finite set of concepts $\{C_1, \ldots, C_n\} \subseteq \mathcal{L}$ such that:*

(i) $\models C \sqsubseteq C_i$ and $\not\models C_i \sqsubseteq C$ for all $1 \leq i \leq n$, and

(ii) for all $D \in \mathcal{L}$ with $\models C \sqsubseteq D$ and $\not\models D \sqsubseteq C$ we have $\models C_i \sqsubseteq D$ for some $1 \leq i \leq n$.

Frontiers relative to an ontology $\mathcal{O}$ are defined similarly, except that each $\models$ or $\not\models$ is replaced by "$\mathcal{O} \models$", resp. "$\mathcal{O} \not\models$".

Theorem 3.10. *For each $C \in \mathcal{L}(\geq, \sqcap, \top)$, one can construct a frontier $\mathcal{F}(C)$ for C w.r.t. $\mathcal{L}(\geq, \sqcap, \top)$ in polynomial time.*

By combining Thm. 3.10 and 3.8, we obtain the main result of this section (where Lem. 3.6 is used to bound the size of the positive examples and Thm. 3.10 is used to construct the negative examples).

Theorem 3.11. $\mathcal{L}(\geq, \sqcap, \top)$ *admits finite characterisations of doubly exponential size.*

Thm. 3.15 below implies an exponential lower bound for the same problem, leaving us with a gap of one exponential.
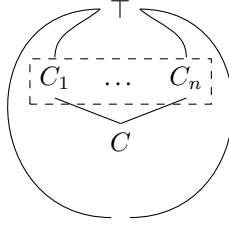
Figure 2: A frontier in the concept subsumption hierarchy

Remark 3.12. The polynomial-time frontier construction from Thm. 3.10 is interesting in its own right, particulary because frontiers are closely related to *upward refinement operators*, a core component of DL learning systems such as the ELTL variant of DL Learner [Bühmann *et al.*, 2016]. We can show that the function ρ mapping each $\mathcal{L}(\geq, \sqcap, \top)$-concept to its frontier is an *ideal upward refinement operator* as defined in [Lehmann and Hitzler, 2007]. This follows from the fact that, for every $\mathcal{L}(\geq, \sqcap, \top)$-concept C there are only finitely many $\mathcal{L}(\geq, \sqcap, \top)$-concepts D, up to equivalence, for which $\models C \sqsubseteq D$. The latter, in turn, holds because $\models C \sqsubseteq D$ implies that $\mathsf{dp}(D) \leq \mathsf{dp}(C)$ and $\mathsf{nr}(D) \leq \mathsf{nr}(C)$.

3.3 Negative Results

The following theorem summarizes our negative result pertaining to the existence of finite characterisations.

Theorem 3.13. $\mathcal{L}(\geq, \sqcup)$ *and* $\mathcal{L}(\geq, \bot)$ *and* $\mathcal{L}(\forall, \exists, -, \sqcap)$ *do not admit finite characterisations.*

In the remainder of this section, we establish lower bounds on the size and/or the complexity of computing finite characterisations. The following lemma will be helpful for this.

Lemma 3.14. *If* $\sqcap \in \mathbf{O}$ *and* (E^+, E^-) *is a finite characterisation of a concept* $C \in \mathcal{L}(\mathbf{O})$ *w.r.t.* $\mathcal{L}(\mathbf{O})$, *then it must be that, for all* $C' \in \mathcal{L}(\mathbf{O})$, *if* $\not\models C \sqsubseteq C'$ *then* C' *is falsified on some positive example in* E^+.

Using this lemma, we show:

Theorem 3.15. $\mathcal{L}(\geq, \sqcap)$ *does not admit polynomial size characterisations. In fact, there are* $\mathcal{L}(\geq, \sqcap)$-*concepts* $(C_i)_{i \in \mathbb{N}}$ *of constant depth and number restriction such that every finite characterisation of* C_i *has size at least* $2^{|C_i|}$.

The next lemma establishes a connection between the problem of computing finite characterisations and the problem of checking subsumptions between concepts.

Lemma 3.16. *If* $\mathcal{L}(\mathbf{O})$ *admits polynomial characterisations,* $\sqcap \in \mathbf{O}$ *and* $\mathcal{L}(\mathbf{O})$ *has a polynomial time model checking problem (in combined complexity), then subsumptions between* $\mathcal{L}(\mathbf{O})$ *concepts can be tested in polynomial time.*

This allows us to use hardness results from the literature on subsumption algorithms to establish hardness of computing certain example sets. Note that the converse of Lem. 3.16 fails, since by Theorem 3.4 $\mathcal{L}(\geq, \sqcap, \top)$ has a polynomial time subsumption problem but does not admit polynomial-time computable characterisations (Thm. 3.15).

The following subsumption hardness result follows from [Donini *et al.*, 1992].

Proposition 3.17. *Subsumption checking for* $\mathcal{L}(\forall, \exists, \sqcap)$ *is* NP-*hard.*

Using Lem. 3.16 we can the establish our next result, resolving an open question from [Cate and Koudijs, 2024].

Corollary 3.18. $\mathcal{L}(\forall, \exists, \sqcap)$ *does not admit polynomial time computable characterisations, unless* $\mathsf{P} = \mathsf{NP}$.

Theorems 1.1 and 1.2 follow from the results proved in this section and prior results that we recalled in Thm. 3.1.

4 Finite Characterisations With Ontologies

In this section we are concerned with finite characterisations of concepts w.r.t. ontologies.

4.1 $\mathcal{L}(\exists, \sqcap, \top, \bot)$ Admits Finite Characterisations Under DL-Lite Ontologies

We show that $\mathcal{L}(\exists, \sqcap, \top, \bot)$ admits finite characterisations under DL-Lite ontologies. The proof uses the following key lemma from [Funk *et al.*, 2022]. Recall the definition of a frontier relative to an ontology (Def. 3.9).

Lemma 4.1. *([Funk* et al.*, 2022]) There is a polynomial-time algorithm that takes as input a* $\mathcal{L}(\exists, \sqcap, \top)$-*concept* C *and a DL-Lite ontology* $\mathcal{O}$ *such that* C *is satisfiable w.r.t.* $\mathcal{O}$, *and outputs a frontier for* C *w.r.t.* $\mathcal{L}(\exists, \sqcap, \top)$ *relative to* $\mathcal{O}$.

Our second ingredient is a polynomial-time *canonical model construction* for $\mathcal{L}(\exists, \sqcap, \top)$-concepts w.r.t. DL-Lite ontologies. Intuitively, one can think of a canonical model, for a concept C and ontology $\mathcal{O}$, as a minimal positive example, i.e., a positive example for C under $\mathcal{O}$ that "makes as few other concepts true as possible". We show that every $\mathcal{L}(\exists, \sqcap, \top)$ concept that is satisfiable w.r.t. a given DL-Lite ontology $\mathcal{O}$ admits a polynomial-time computable canonical model w.r.t. $\mathcal{O}$ that satisfies its CIs.

Theorem 4.2. *There is a polynomial-time algorithm that takes a* $\mathcal{L}(\exists, \sqcap, \top)$-*concept* C *and a DL-Lite ontology* $\mathcal{O}$, *where* C *is satisfiable w.r.t.* $\mathcal{O}$, *and that produces a pointed interpretation* $(\mathcal{I}_{C,\mathcal{O}}, d_C)$ *(which we will call the* canonical model *of* C *under* $\mathcal{O}$) *with the following properties.*

1. *for all* $\mathcal{L}(\exists, \sqcap, \top)$-*concepts* D, $d_C \in D^{\mathcal{I}_{C,\mathcal{O}}}$ *iff* $\mathcal{O} \models C \sqsubseteq D$. *In particular, we have that* $d_C \in C^{\mathcal{I}_{C,\mathcal{O}}}$.

2. *Moreover,* $\mathcal{I}_{C,\mathcal{O}} \models \mathcal{O}$.

The proof is based on previous constructions of canonical model for query answering [Kontchakov *et al.*, 2010]. Canonical models for query answering usually differ from those for satisfying the ontology. In the case of the $\mathcal{EL}$ ontology language (consisting of CIs $C \sqsubseteq D$ where $C, D \in \mathcal{L}(\exists, \sqcap, \top)$), it was shown that a polynomial canonical model for $\mathcal{EL}$ concepts can be constructed [Baader *et al.*, 2005; Lutz and Wolter, 2010], but it cannot be ensured that this canonical model satisfies the ontology. In fact there are $\mathcal{EL}$ ontologies for which there is no *finite* canonical model satisfying the ontology [Guimarães *et al.*, 2023, Example 33].

Thm. 4.2 *does* produce models satisfying the DL-Lite ontology, and this is important for our purpose. By combining the frontier construction with the canonical model construction, we can construct finite characterisations in polynomial time

for concepts that are satisfiable w.r.t. a given ontology. For inconsistent concepts, we separately prove the following.

Lemma 4.3. *Let $\mathcal{O}$ be a DL-Lite ontology. Then the unsatisfiable concept $\bot$ admits an exponentially sized finite characterisation w.r.t. $\mathcal{L}(\exists, \sqcap, \top, \bot)$ under $\mathcal{O}$.*

Given the above, we obtain the main result of this section.

Theorem 4.4. $\mathcal{L}(\exists, \sqcap, \top, \bot)$ *admits finite characterisations under DL-Lite ontologies. Moreover, for concepts $C \in \mathcal{L}(\exists, \sqcap, \top, \bot)$ that are satisfiable w.r.t. the given ontology $\mathcal{O}$, a finite characterisation can be computed in polynomial time (in $|C|$ and $|\mathcal{O}|$).*

Remark 4.5. The restriction to satisfiable concepts in the above theorem is essential given that the ontology is treated as part of the input. Specifically, let $\mathcal{O}$ be the ontology $\mathcal{O} := \{A_0 \sqsubseteq \neg A_0\} \cup \{A_i \sqsubseteq \neg B_i \mid i = 1 \ldots n\}$. It can be shown, for the concept A_0 (which is unsatisfiable w.r.t. $\mathcal{O}$), that every finite characterisation must be of size at least 2^n (see e.g. [Funk *et al.*, 2021] for a similar style of counterexample). If the ontology is treated as fixed in the complexity analysis, however, a finite characterisation could be computed in polynomial time for *all* $\mathcal{L}(\exists, \sqcap, \top, \bot)$-concepts.

4.2 Negative Results

The following two negative results establish that we cannot further enrich the concept language.

Theorem 4.6. *Let $\mathcal{O}$ be the DL-Lite ontology $\{A \sqsubseteq \neg A\}$. Then $\mathcal{L}(\geq)$ and $\mathcal{L}(\forall, \exists, \sqcap)$ do not admit finite characterisations under $\mathcal{O}$.*

Theorem 4.7. *Let $\mathcal{O}$ be the DL-Lite ontology $\{A \sqsubseteq \exists R, \exists R^- \sqsubseteq A\}$. Then $\mathcal{L}(\exists, -, \sqcap)$ does not admit finite characterisations under $\mathcal{O}$.*

These results, together with Thm. 4.4, imply Thm. 1.3, as they show that no extension of $\mathcal{L}(\exists, \sqcap, \top, \bot)$ with $\geq$ or $\forall$ admits finite characterisations under DL-Lite ontologies. Our positive result in Section 4.1 also cannot be extended to DL-Lite$^{\mathcal{H}}$ ontologies (that is, DL-Lite extended with role inclusions [Artale *et al.*, 2009]). This contrasts with a recent positive result for $\mathcal{EL}$ with DL-Lite$^{\mathcal{H}}$ ontologies using ABox-examples [Funk *et al.*, 2022].

Theorem 4.8. *Let $\mathcal{O}$ be the DL-Lite$^{\mathcal{H}}$ ontology $\{A \sqsubseteq \exists R, \exists R^- \sqsubseteq A, R^- \sqsubseteq S\}$. Then $\mathcal{L}(\exists, \sqcap)$ does not admit finite characterisations under $\mathcal{O}$.*

5 Summary and Discussion

We systematically studied the existence and complexity of computing finite characterisations for concept languages $\mathcal{L}(\mathbf{O})$ with $\{\exists, \sqcap\} \subseteq \mathbf{O} \subseteq \{\forall, \exists, \geq, -, \sqcap, \sqcup, \top, \bot, \neg\}$, both in the absence of an ontology and in the presence of a DL-Lite ontology. While we classified most concept languages, there are a few cases left open, detailed below.

In passing, we obtained a polynomial-time algorithm for checking subsumptions between $\mathcal{L}(\exists, \geq, \sqcap, \top)$ concepts, and a polynomial time algorithm for computing frontiers of $\mathcal{L}(\exists, \geq, \sqcap, \top)$-concepts (which gives rise to a polynomial-time ideal upward refinement operator, cf. Remark 3.12).

Repercussions for Exact Learnability As we mentioned in the introduction, the existence of polynomial-time computable characterisations is a necessary precondition for the existence of polynomial-time exact learning algorithms with membership queries. It was shown in [Cate and Dalmau, 2022] that $\mathcal{L}(\exists, -, \sqcap, \top, \bot)$ is indeed polynomial-time exactly learnable with membership queries. Thm. 1.2 therefore implies that $\mathcal{L}(\exists, -, \sqcap, \top, \bot)$ is a maximal fragment of $\mathcal{ALCQI}$ that is efficiently exactly learnable with membership queries, in the absence of an ontology.

Conversely, every algorithm for effectively computing finite characterisations also gives rise to an effective exact learning algorithm with membership queries, although not an efficient one. Thus, Thm. 1.1 also tells us something about the limits of (not-necessarily efficient) exact learnability with membership queries, without an ontology.

Finally, in the same way, Thm. 1.3 tells us something about the limits of exact learnability with membership queries where the membership queries can only be asked for examples satisfying a given DL-Lite ontology. The authors are not aware of prior work on such a notion of learnability.

Open questions Open questions include whether $\mathcal{L}(\exists, \geq, -, \sqcap)$ or $\mathcal{L}(\exists, \geq, -, \top)$ admit finite characterisations (under the empty ontology) and whether concept languages between $\mathcal{L}(\exists, \sqcap, \sqcup)$ and $\mathcal{L}(\exists, \sqcap, \sqcup, \top, \bot)$ admit finite characterisations under DL-Lite ontologies. Moreover, we do not yet have tight bounds on the complexity of computing finite characterisations for many of the concept languages we study that admit finite characterisations. In particular, for concept languages between $\mathcal{L}(\forall, \exists, \sqcap)$ and $\mathcal{L}(\forall, \exists, \geq \sqcap, \top)$ we have an NP lower bound (from Corollary 3.18) and a non-elementary upper bound (from Thm. 3.3). In Thm. 3.11 we managed to improve this to a doubly exponential upper bound for $\mathcal{L}(\exists, \geq, \sqcap, \top)$, while an exponential lower bound for $\mathcal{L}(\exists, \geq, \sqcap)$ follows from Thm. 3.15. It remains to find tight bounds for these fragments.

Since our setting includes $\forall$ and $\neg$, we were forced to work with finite pointed interpretations as examples (cf. Remark 2.2). If one discards these connectives it becomes natural to adopt pointed ABoxes as examples, as in [Funk *et al.*, 2022]. Many of our negative results for the existence of finite characterisations under DL-Lite ontologies rely on examples being finite pointed interpretations satisfying the ontology, except Thm. 4.6 because for the ontology in question, there is essentially no difference between a (consistent) ABox and a finite interpretation. Our study therefore only leaves open the possibility of positive results in the ABox-as-examples setting under DL-Lite ontologies for fragments of $\mathcal{L}(\exists, -, \sqcap, \sqcup, \top, \bot)$.

It would be of interest to investigate finite characterisations under ontology languages other than DL-Lite. The recent paper [Cate *et al.*, 2024] implies some positive results for $\mathcal{L}(\exists, \sqcap, \sqcup, \top, \bot)$ relative to ontologies defined by certain universal Horn formulas (e.g. symmetry and transitivity).

One of the motivations of our study was to generate examples to illustrate a concept to a user (e.g, for debugging purposes or for educational purposes). Finding practical solutions for this require more research, e.g., on how to generate "natural" examples (cf. [Glavic *et al.*, 2021] for an overview of related problems and techniques in data management).

Acknowledgments

References

[Alexe *et al.*, 2011] Bogdan Alexe, Balder ten Cate, Ph. G. Kolaitis, and Wang Chiew Tan. Characterizing schema mappings via data examples. *ACM Trans. Database Syst.*, 36:23:1–23:48, 2011.

[Angluin, 1987] Dana Angluin. Queries and concept learning. *Mach. Learn.*, 2(4):319–342, 1987.

[Artale *et al.*, 2009] Alessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyaschev. The DL-Lite family and relations. *J. Artif. Intell. Res.*, 36:1–69, 2009.

[Baader *et al.*, 2005] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the EL envelope. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *IJCAI*, pages 364–369. Professional Book Center, 2005.

[Bühmann *et al.*, 2016] Lorenz Bühmann, Jens Lehmann, and Patrick Westphal. DL-Learner - A framework for inductive learning on the semantic web. *J. Web Sem.*, 39:15–24, 2016.

[Cate and Dalmau, 2022] Balder ten Cate and Víctor Dalmau. Conjunctive queries: Unique characterizations and exact learnability. *ACM Trans. Database Syst.*, 47(4), nov 2022.

[Cate and Koudijs, 2024] Balder ten Cate and Raoul Koudijs. Characterising modal formulas with examples, 2024. To appear in ACM Transactions on Computational Logic.

[Cate *et al.*, 2024] Balder ten Cate, Víctor Dalmau, and Jakub Opršal. Right-adjoints for datalog programs. In *Proceedings of ICDT 2024*, 2024.

[Donini *et al.*, 1992] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, Bernhard Hollunder, Werner Nutt, and Alberto Marchetti Spaccamela. The complexity of existential quantification in concept languages. *Artificial Intelligence*, 53(2):309–327, 1992.

[Fanizzi *et al.*, 2008] Nicola Fanizzi, Claudia d'Amato, and Floriana Esposito. DL-FOIL concept learning in description logics. In *Proc. of ILP*, pages 107–121, 2008.

[Fortin *et al.*, 2022] Marie Fortin, Boris Konev, Vladislav Ryzhikov, Yury Savateev, Frank Wolter, and Michael Zakharyaschev. Unique Characterisability and Learnability of Temporal Instance Queries. In *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning*, pages 163–173, 8 2022.

[Funk *et al.*, 2019] Maurice Funk, Jean Christoph Jung, Carsten Lutz, Hadrien Pulcini, and Frank Wolter. Learning description logic concepts: When can positive and negative examples be separated? In *IJCAI*, pages 1682–1688, 2019.

[Funk *et al.*, 2021] Maurice Funk, Jean Christoph Jung, and Carsten Lutz. Actively learning concept and conjunctive queries under $\mathcal{EL}^r$-ontologies. In *IJCAI*, pages 1887–1893, 2021.

[Funk *et al.*, 2022] Maurice Funk, Jean Christoph Jung, and Carsten Lutz. Frontiers and exact learning of ELI queries under dl-lite ontologies. In *IJCAI*, pages 2627–2633, 2022.

[Glavic *et al.*, 2021] Boris Glavic, Alexandra Meliou, and Sudeepa Roy. *Trends in Explanations - Understanding and Debugging Data-driven Systems*. Foundations and Trends in Databases. Now Publishers, Inc., 2021.

[Goldman and Kearns, 1995] S.A. Goldman and M.J. Kearns. On the complexity of teaching. *Journal of Computer and System Sciences*, 50(1):20–31, 1995.

[Guimarães *et al.*, 2023] Ricardo Guimarães, Ana Ozaki, and Jandson S. Ribeiro. Eviction and reception for description logic ontologies (preliminary results). In Kai Sauerwald and Matthias Thimm, editors, *NMR]*, volume 3464 of *CEUR Workshop Proceedings*, pages 43–51. CEUR-WS.org, 2023.

[Iannone *et al.*, 2007] Luigi Iannone, Ignazio Palmisano, and Nicola Fanizzi. An algorithm based on counterfactuals for concept learning in the semantic web. *Appl. Intell.*, 26(2):139–159, 2007.

[Konev *et al.*, 2016] Boris Konev, Ana Ozaki, and Frank Wolter. A model for learning description logic ontologies based on exact learning. In Dale Schuurmans and Michael P. Wellman, editors, *AAAI*, pages 1008–1015. AAAI Press, 2016.

[Kontchakov *et al.*, 2010] Roman Kontchakov, Carsten Lutz, David Toman, Frank Wolter, and Michael Zakharyaschev. The combined approach to query answering in DL-Lite. In *Proc. of KR*, 2010.

[Lehmann and Hitzler, 2007] Jens Lehmann and Pascal Hitzler. Foundations of refinement operators for description logics. In *International Conference on Inductive Logic Programming*, 2007.

[Lehmann and Hitzler, 2010] Jens Lehmann and Pascal Hitzler. Concept learning in description logics using refinement operators. *Mach. Learn.*, 78:203–250, 2010.

[Lehmann, 2009] Jens Lehmann. Dl-learner: Learning concepts in description logics. *Journal of Machine Learning Research*, 10:2639–2642, 2009.

[Lutz and Wolter, 2010] Carsten Lutz and Frank Wolter. Deciding inseparability and conservative extensions in the description logic $\mathcal{EL}$. *J. Symb. Comput.*, 45(2):194–228, 2010.

[Mannila and Räihä, 1985] Heikki Mannila and Kari-Jouko Räihä. Test data for relational queries. *Proceedings of the fifth ACM SIGACT-SIGMOD symposium on Principles of database systems*, 1985.

[Ozaki *et al.*, 2020] Ana Ozaki, Cosimo Persia, and Andrea Mazzullo. Learning query inseparable $\mathcal{ELH}$ ontologies. In *AAAI*, pages 2959–2966. AAAI Press, 2020.

[Ozaki, 2020] Ana Ozaki. Learning description logic ontologies: Five approaches. where do they stand? *Künstliche Intell.*, 34(3):317–327, 2020.

[Staworko and Wieczorek, 2015] Slawomir Staworko and Piotr Wieczorek. Characterizing xml twig queries with examples. In *International Conference on Database Theory*, 2015.

[Telle *et al.*, 2019] Jan Arne Telle, José Hernández-Orallo, and C. Ferri. The teaching size: computable teachers and learners for universal languages. *Machine Learning*, 108:1653–1675, 2019.