Logical Separability of Incomplete Data under Ontologies

Jean Christoph Jung[1] , **Carsten Lutz**[1] , **Hadrien Pulcini**[2] , **Frank Wolter**[2]
[1]University of Bremen, Germany
[2]University of Liverpool, UK
{jeanjung,clu}@uni-bremen.de, {h.pulcini,wolter}@liverpool.ac.uk

Abstract

Finding a logical formula that separates positive and negative examples given in the form of labeled data items is fundamental in applications such as concept learning, reverse engineering of database queries, and generating referring expressions. In this paper, we investigate the existence of a separating formula for incomplete data in the presence of an ontology. Both for the ontology language and the separation language, we concentrate on first-order logic and three important fragments thereof: the description logic $\mathcal{ALCI}$, the guarded fragment, and the two-variable fragment. We consider several forms of separability that differ in the treatment of negative examples and in whether or not they admit the use of additional helper symbols to achieve separation. We characterize separability in a model-theoretic way, compare the separating power of the different languages, and determine the computational complexity of separability as a decision problem.

1 Introduction

There are several scenarios in which the aim is to find some kind of logical formula that separates positive from negative examples given in the form of labeled data items. In concept learning in description logic (DL), the aim is to automatically construct a concept description that can then be used, for instance, in ontology engineering (Lehmann and Hitzler 2010). In reverse engineering of database queries, also known as query by example (QBE), one seeks to find a query from example answers and non-answers provided by a user who is able to give such examples, but not to formulate the query (Martins 2019). In generating referring expression (GRE), the aim is to find a formula that separates a single positive data item from all other data items and can thus be used as a uniquely identifying description of the data item (Krahmer and van Deemter 2012). And in entity comparison, the separation is between a single positive and a single negative data item, aiming to summarize the differences between the two (Petrova et al. 2017).

In this paper, we consider the separation of positive and negative examples given in the form of data items, in the presence of an ontology. As usual when data and ontologies are combined, we assume that the data is incomplete and adopt an open world semantics. This matches the setup of concept learning for DLs and of QBE and GRE for ontology-mediated queries which have both received recent interest (Borgida, Toman, and Weddell 2016; Gutiérrez-Basulto, Jung, and Sabellek 2018). It also encompasses entity comparison under ontologies. While separating formulas are often required to have additional properties such as providing a good abstraction of the positive examples (in QBE) or being comprehensible (in GRE), a fundamental question common to these applications is when and whether a separating formula exists at all. It is this question of separability that we concentrate on in the present paper.

We assume that a labeled knowledge base (KB) $(\mathcal{K}, P, N)$ is given, $\mathcal{K} = (\mathcal{O}, \mathcal{D})$, where $\mathcal{O}$ is an ontology, $\mathcal{D}$ a database, P a set of positive examples, and N a set of negative examples. All examples are tuples of constants of the same length. Due to the open world semantics, different choices are possible regarding the definition of a formula φ that separates $(\mathcal{K}, P, N)$. While it is uncontroversial to demand that $\mathcal{K} \models \varphi(\vec{a})$ for all $\vec{a} \in P$, for negative examples $\vec{b} \in N$ it makes sense to demand that $\mathcal{K} \not\models \varphi(\vec{b})$, but also that $\mathcal{K} \models \neg\varphi(\vec{b})$. When φ is formulated in logic $\mathcal{L}$, we refer to the former as *weak $\mathcal{L}$-separability* and to the latter as *strong $\mathcal{L}$-separability*. Moreover, one might or might not admit the use of helper symbols in φ that do not occur in $\mathcal{K}$, giving rise to *projective* and *non-projective* versions of separability. While it might be debatable whether the use of helper symbols is natural in separating formulas, they arise very naturally when studying the separating power of different logics used as a separation language. We study all four cases that emerge from these choices. Projective weak separability has already been studied for a variety of DLs in (Funk et al. 2019) and some first observations on strong separability were presented in the same paper.

We study ontologies and separating formulas formulated in first-order logic (FO), its guarded negation fragment (GNFO), its guarded fragment (GF), its two-variable fragment FO^2, and the DL $\mathcal{ALCI}$—a fragment of both GF and FO^2. As separating formulas, we additionally consider unions of conjunctive queries (UCQs). With $(\mathcal{L}, \mathcal{L}_S)$-separability, we mean $\mathcal{L}_S$-separability of labeled $\mathcal{L}$-KBs. We aim to characterize $(\mathcal{L}, \mathcal{L}_S)$-separability in a model-theoretic way, to compare the separating power of different languages $\mathcal{L}_S$, and to determine the decidability and complexity of $(\mathcal{L}, \mathcal{L}_S)$-separability as a decision problem.

We start with weak separability. Our first main result provides a characterization of (weak) (FO, FO)-separability in terms of homomorphisms. It implies that projective and non-projective $(\mathrm{FO}, \mathcal{L}_S)$-separability coincide for all FO-fragments $\mathcal{L}_S$ situated between FO and UCQ (such as GNFO), and that moreover $(\mathrm{FO}, \mathcal{L}_S)$-separability coincides for all such $\mathcal{L}_S$. Note that this is due to the open world semantics. Our result also lifts the link between separability and UCQ-evaluation on KBs first observed in (Funk et al. 2019) to a more general setting. As a first application, we use it to show that $(\mathrm{GNFO}, \mathrm{GNFO})$-separability is decidable and 2ExpTime-complete.

We then proceed to study $(\mathcal{L}, \mathcal{L})$-separability for the fragments $\mathcal{L} \in \{\mathcal{ALCI}, \mathrm{GF}, \mathrm{FO}^2\}$. Note that these fragment do not contain UCQ and thus the above results do not apply. In fact, the projective and non-projective cases do not coincide for any of these $\mathcal{L}$. We start with projective $(\mathcal{ALCI}, \mathcal{ALCI})$-separability. It is implicit in (Funk et al. 2019) that this is the same as (projective and non-projective) $(\mathcal{ALCI}, \mathrm{UCQ})$-separability and thus, by the results above, also as $(\mathcal{ALCI}, \mathrm{FO})$-separability. It is proved in in (Funk et al. 2019) that this separability problem is NExpTime-complete in combined complexity and it is claimed to be Π_2^p-complete in data complexity where the ontology is assumed to be fixed. We first correct the latter statement and show that the problem is NExpTime-complete also in data complexity. We then turn to the technically more intricate case of non-projective $(\mathcal{ALCI}, \mathcal{ALCI})$-separability, observe that it does not coincide with the projective case, and characterize it using a mix of homomorphisms, bisimulations, and types. This allows us to show that non-projective $(\mathcal{ALCI}, \mathcal{ALCI})$-separability is also NExpTime-complete, both in combined complexity and in data complexity.

For projective and non-projective $(\mathrm{GF}, \mathrm{GF})$-separability, we establish characterizations that parallel those for $\mathcal{ALCI}$ except that bisimulations are replaced with (a form of) guarded bisimulations. The proofs are significantly more subtle. As in the $\mathcal{ALCI}$-case, projective $(\mathrm{GF}, \mathcal{L}_S)$-separability coincides with $(\mathrm{GF}, \mathrm{UCQ})$-separability and thus also with $(\mathrm{GF}, \mathrm{FO})$-separability. We additionally observe that is also coincides with projective $(\mathrm{GF}, \mathrm{openGF})$-separability where openGF is a 'local' version of GF that arguably is a natural choice for separation (Hernich et al. 2020). A main result is then that projective and non-projective $(\mathrm{GF}, \mathrm{GF})$-separability are 2ExpTime-complete in combined complexity. We next show that, in contrast, $(\mathrm{FO}^2, \mathrm{FO}^2)$-separability and $(\mathrm{FO}^2, \mathrm{FO})$-separability are both undecidable. Moreover, they coincide neither in the projective nor in the non-projective case. These results are linked in an interesting way to the fact that FO^2 has the finite model property but is not finitely controllable for UCQs.

We then switch to strong separability, first observing that in marked contrast to the weak case, projective strong $(\mathcal{L}, \mathcal{L}_S)$-separability coincides with non-projective strong $(\mathcal{L}, \mathcal{L}_S)$-separability for all choices of $\mathcal{L}$ and $\mathcal{L}_S$ relevant to this paper. We establish a characterization of strong $(\mathrm{FO}, \mathrm{FO})$-separability in terms of KB unsatisfiability and show that strong $(\mathrm{FO}, \mathrm{FO})$-separability coincides with strong $(\mathrm{FO}, \mathrm{UCQ})$-separability and consequently with strong $(\mathrm{FO}, \mathcal{L}_S)$-separability for all $\mathcal{L}_S$ between FO and UCQ. We next consider the same FO-fragments $\mathcal{ALCI}, \mathrm{GF}, \mathrm{FO}^2$ as before and show that for each of these fragments $\mathcal{L}$, strong $(\mathcal{L}, \mathcal{L})$-separability coincides with strong $(\mathcal{L}, \mathrm{FO})$-separability and thus the connection to KB unsatisfiability applies. This allows us to derive tight complexity bounds for stong strong $(\mathcal{L}, \mathcal{L})$-separability. For $\mathcal{ALCI}$, ExpTime-completeness in combined complexity and coNP-completeness in data complexity was shown in (Funk et al. 2019). We prove completeness for 2ExpTime and NExpTime in combined complexity for GF and FO^2, respectively, and coNP-completeness in data complexity in both cases. Note that strong $(\mathrm{FO}^2, \mathrm{FO}^2)$-separability thus turns out to be decidable, in contrast to the weak case.

2 Related Work and Applications

We discuss in more detail related work and applications of our results, starting with concept learning in DL as first proposed in (Badea and Nienhuys-Cheng 2000). Inspired by inductive logic programming, refinement operators are used to construct a concept that generalizes positive examples while not encompassing any negative ones. An ontology may or may not be present. There has been significant interest in this approach, both for weak separation (Lehmann and Haase 2009; Lehmann and Hitzler 2010; Lisi and Straccia 2015; Sarker and Hitzler 2019) and strong separation (Fanizzi, d'Amato, and Esposito 2008; Lisi 2012). Prominent systems include the DL Leaner (Bühmann et al. 2018; Bühmann, Lehmann, and Westphal 2016), DL-Foil, YinYang, and pFOIL-DL (Fanizzi et al. 2018; Iannone, Palmisano, and Fanizzi 2007; Straccia and Mucci 2015). A method for generating strongly separating concepts based on bisimulations has been developed in (Ha et al. 2012; Tran, Nguyen, and Hoang 2015; Divroodi et al. 2018) and an approach based on answer set programming was proposed in (Lisi 2016). Algorithms for DL concept learning typically aim to be complete, that is, to find a separating concept whenever there is one. Complexity lower bounds for separability as studied in this paper then point to an inherent complexity that no such algorithm can avoid. Undecidability even means that there can be no learning algorithm that is both terminating and complete. The complexity of deciding separability in DL concept learning was first investigated in (Funk et al. 2019). Computing least common subsumers (LCS) and most specific concepts (MSC) can be viewed as DL concept learning in the case that only positive, but no negative example are available (Cohen, Borgida, and Hirsh 1992; Nebel 1990; Baader, Küsters, and Molitor 1999; Zarrieß and Turhan 2013). A recent study of LCS and MSC from a separability angle is in (Jung, Lutz, and Wolter 2020).

Query by example is an active topic in database research since many years, see e.g. (Tran, Chan, and Parthasarathy 2009; Zhang et al. 2013; Weiss and Cohen 2017; Kalashnikov, Lakshmanan, and Srivastava 2018; Deutch and Gilad 2019; Staworko and Wieczorek 2012) and (Martins 2019) for a recent survey. In this context, separability has also received attention (Arenas and Diaz 2016; Barceló and Romero 2017; Kimelfeld and Ré 2018). A crucial difference to the present paper is that QBE in classical databases uses a closed world semantics under which there is a unique natural way to treat negative examples: simply demand that the separating formula evaluates to false there. Thus, the distinction between weak and strong separability, and also between projective and non-projective separability does not arise. Moreover, the separating power of many logics is much higher under a closed world semantics; for instance, FO-separability is far from coinciding with UCQ-separability. QBE for ontology-mediated querying (Gutiérrez-Basulto, Jung, and Sabellek 2018; Ortiz 2019) and for SPARQL queries (Arenas, Diaz, and Kostylev 2016), in contrast, makes an open world semantics. The former is captured by the framework studied in the current article. In fact, our results imply that the existence of a separating UCQ is decidable for ontology languages such as $\mathcal{ALCI}$ and the guarded fragment. The corresponding problem for CQs is undecidable even when the ontology is formulated in the inexpressive description logic $\mathcal{ELI}$ (Funk et al. 2019; Jung, Lutz, and Wolter 2020).

Generating referring expressions has originated from linguistics (Krahmer and van Deemter 2012) and has recently received interest in the context of ontology-mediated querying (Areces, Koller, and Striegnitz 2008; Borgida, Toman, and Weddell 2016; Toman and Weddell 2019). GRE fits into the framework used in this paper since a formula that separates a single data item from all other items in the KB can serve as a referring expression for the former. Both weak and strong separability are conceivable: weak separability means that the positive data item is the only one that we are certain to satisfy the separating formula and strong separability means that in addition we are certain that the other data items do not satisfy the formula. Approaches to GRE such as the ones in (Borgida, Toman, and Weddell 2016) aim for even stronger guarantees as the positive example must in a sense also be separated from all 'existential objects', that is, objects that are not explicitly mentioned in the database, but whose existence is asserted by the ontology. Such a strong guarantee, however, cannot be achieved in the ontology languages studied here (Toman and Weddell 2019).

In *entity comparison*, one aims to compare two selected data items, highlighting both their similarities and their differences. An approach to entity comparison in RDF graphs is presented in (Petrova et al. 2017; Petrova et al. 2019). There, SPARQL queries are used to describe both similarities and differences, under an open world semantics. The 'computing similarities' part of this approach is closely related to the LCS and MSC mentioned above. The 'computing differences' is closely related to QBE and fits into the framework studied in this paper. In fact, it corresponds to separation with a single positive and a single negative example, and with an empty ontology.

3 Preliminaries

Let Σ_{full} be a set of *relation symbols* that contains countably many symbols of every arity $n \geq 1$ and let Const be a countably infinite set of *constants*. A *signature* is a set of relation symbols $\Sigma \subseteq \Sigma_{\text{full}}$. We write $\vec{a}$ for a tuple $(a_1, \dots, a_n)$ of constants and set $[\vec{a}] = \{a_1, \dots, a_n\}$. A *database* $\mathcal{D}$ is a finite set of *ground atoms* $R(\vec{a})$, where $R \in \Sigma_{\text{full}}$ has arity n and $\vec{a}$ is a tuple of constants from Const of length n. We use $\text{cons}(\mathcal{D})$ to denote the set of constant symbols in $\mathcal{D}$.

Denote by FO the set of first-order (FO) formulas constructed from constant-free atomic formulas $x = y$ and $R(\vec{x})$, $R \in \Sigma_{\text{full}}$, using conjunction, disjunction, negation, and existential and universal quantification. As usual, we write $\varphi(\vec{x})$ to indicate that the free variables in the FO-formula φ are all from $\vec{x}$ and call a formula *open* if it has at least one free variable and a *sentence* otherwise. Note that we do not admit constants in FO-formulas. While many results presented in this paper should lift to the case with constants, dealing with constants introduces significant technical complications that are outside the scope of this paper.

A *fragment* of FO is a set of FO formulas that is closed under conjunction. We consider various such fragments. A *conjunctive query (CQ)* takes the form $q(\vec{x}) = \exists \vec{y} \varphi$ where φ is a conjunction of atomic formulas $x = y$ and $R(\vec{y})$. We assume w.l.o.g. that if a CQ contains an equality $x = y$, then x and y are free variables. A *union of conjunctive queries (UCQ)* is a disjunction of CQs that all have the same free variables. In the context of CQs and UCQs, we speak of *answer variables* rather than of free variables. A CQ q is *rooted* if every variable in it is reachable from an answer variable in the Gaifman graph of q viewed as a hypergraph and a UCQ is *rooted* if every CQ in it is. We write (U)CQ also to denote the class of all (U)CQs.

In the *guarded fragment (GF)* of FO (Andréka, Németi, and van Benthem 1998; Grädel 1999), formulas are built from atomic formulas $R(\vec{x})$ and $x = y$ by applying the Boolean connectives and *guarded quantifiers* of the form

$$\forall \vec{y}(\alpha(\vec{x}, \vec{y}) \rightarrow \varphi(\vec{x}, \vec{y})) \text{ and } \exists \vec{y}(\alpha(\vec{x}, \vec{y}) \wedge \varphi(\vec{x}, \vec{y}))$$

where $\varphi(\vec{x}, \vec{y})$ is a guarded formula and $\alpha(\vec{x}, \vec{y})$ is an atomic formula or an equality $x = y$ that contains all variables in $[\vec{x}] \cup [\vec{y}]$. The formula α is called the *guard of the quantifier*. An extension of GF that preserves many of the nice of properties of GF is the *guarded negation fragment* GNFO of FO which contains both GF and UCQ. GNFO is obtained by imposing a guardedness condition on negation instead of on quantifiers, details can be found in (Bárány, ten Cate, and Segoufin 2015). The *two-variable fragment* FO^2 of FO contains every formula in FO that uses only two fixed variables x and y (Grädel, Kolaitis, and Vardi 1997).

For $\mathcal{L}$ an FO-fragment, an *$\mathcal{L}$-ontology* is a finite set of $\mathcal{L}$-sentences. An *$\mathcal{L}$-knowledge base (KB)* is a pair $(\mathcal{O}, \mathcal{D})$, where $\mathcal{O}$ is an $\mathcal{L}$-ontology and $\mathcal{D}$ a database. For any syntactic object O such as a formula, an ontology, and a KB, we use $\text{sig}(O)$ to denote the set of relation symbols that occur in O and $||O||$ to denote the *size* of O, that is, the number of symbols needed to write it with names of relations, variables, and constants counting as a single symbol.

As usual, KBs $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ are interpreted in *relational structures* $\mathfrak{A} = (\text{dom}(\mathfrak{A}), (R^{\mathfrak{A}})_{R \in \Sigma_{\text{full}}}, (c^{\mathfrak{A}})_{c \in \text{Const}})$ where $\text{dom}(\mathfrak{A})$ is the non-empty *domain* of $\mathfrak{A}$, each $R^{\mathfrak{A}}$ is a relation over $\text{dom}(\mathfrak{A})$ whose arity matches that of R, and $c^{\mathfrak{A}} \in \text{dom}(\mathfrak{A})$ for all $c \in \text{Const}$. Note that we do not make the *unique name assumption (UNA)*, that is $c_1^{\mathfrak{A}} = c_2^{\mathfrak{A}}$ might hold even when $c_1 \neq c_2$. This is essential for several of our results. A structure $\mathfrak{A}$ is a *model of a KB* $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ if it satisfies all sentences in $\mathcal{O}$ and all ground atoms in $\mathcal{D}$. A KB $\mathcal{K}$ is *satisfiable* if there exists a model of $\mathcal{K}$.

Description logics are fragments of FO that only support relation symbols of arities one and two, called concept names and role names. DLs come with their own syntax, which we introduce next (Baader et al. 2003; Baader et al. 2017). A *role* is a role name or an *inverse role* R^- with R a role name. For uniformity, we set $(R^-)^- = R$. $\mathcal{ALCI}$-*concepts* are defined by the grammar

$$C, D \ ::= \ A \mid \neg C \mid C \sqcap D \mid \exists R.C$$

where A ranges over concept names and R over roles. As usual, we write $\bot$ to abbreviate $A \sqcap \neg A$ for some fixed concept name A, $\top$ for $\neg\bot$, $C \sqcup D$ for $\neg(\neg C \sqcap \neg D)$, $C \rightarrow D$ for $\neg C \sqcup D$, and $\forall R.C$ for $\neg \exists R.\neg C$. An $\mathcal{ALCI}$-*concept inclusion (CI)* takes the form $C \sqsubseteq D$ where C and D are $\mathcal{ALCI}$-concepts. An $\mathcal{ALCI}$-*ontology* is a finite set of $\mathcal{ALCI}$-CIs. An $\mathcal{ALCI}$-*KB* $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ consists of an $\mathcal{ALCI}$-ontology $\mathcal{O}$ and a database $\mathcal{D}$ that uses only unary and binary relation symbols. We sometimes also mention the fragment $\mathcal{ALC}$ of $\mathcal{ALCI}$ in which inverse roles are not available.

To obtain a semantics, every $\mathcal{ALCI}$-concept C can be translated into an GF-formula $C^\dagger$ with one free variable x:

$$\begin{array}{rcl} A^\dagger &=& A(x) \\ (\neg\varphi)^\dagger &=& \neg\varphi^\dagger \\ (C \sqcap D)^\dagger &=& C^\dagger \wedge D^\dagger \\ (\exists R.C)^\dagger &=& \exists y\, (R(x,y) \wedge C^\dagger[y/x]) \\ (\exists R^-.C)^\dagger &=& \exists y\, (R(y,x) \wedge C^\dagger[y/x]). \end{array}$$

The *extension* $C^\mathfrak{A}$ of a concept C in a structure $\mathfrak{A}$ is defined as $C^\mathfrak{A} = \{a \in \mathrm{dom}(\mathfrak{A}) \mid \mathfrak{A} \models C^\dagger(a)\}$. A CI $C \sqsubseteq D$ translates into the GF-sentence $\forall x\, (C^\dagger(x) \rightarrow D^\dagger(x))$. By reusing variables, we can even obtain formulas and ontologies from GF $\cap$ FO^2. We write $\mathcal{O} \models C \sqsubseteq D$ if $C^\mathfrak{A} \subseteq D^\mathfrak{A}$ holds in every model $\mathfrak{A}$ of $\mathcal{O}$. Concepts C and D are *equivalent* w.r.t. an ontology $\mathcal{O}$ if $\mathcal{O} \models C \sqsubseteq D$ and $\mathcal{O} \models D \sqsubseteq C$.

We close this section with introducing homomorphisms. A *homomorphism* h from a structure $\mathfrak{A}$ to a structure $\mathfrak{B}$ is a function $h : \mathrm{dom}(\mathfrak{A}) \rightarrow \mathrm{dom}(\mathfrak{B})$ such that $\vec{a} \in R^\mathfrak{A}$ implies $h(\vec{a}) \in R^\mathfrak{B}$ for all relation symbols R and tuples $\vec{a}$ and with $h(\vec{a})$ being defined component-wise in the expected way. Note that homomorphisms need not preserve constant symbols. Every database $\mathcal{D}$ gives rise to the finite structure $\mathfrak{A}_\mathcal{D}$ with $\mathrm{dom}(\mathfrak{A}_\mathcal{D}) = \mathrm{cons}(\mathcal{D})$ and $\vec{a} \in R^{\mathfrak{A}_\mathcal{D}}$ iff $R(\vec{a}) \in \mathcal{D}$. A homomorphism from database $\mathcal{D}$ to structure $\mathfrak{A}$ is a homomorphism from $\mathfrak{A}_\mathcal{D}$ to $\mathfrak{A}$. A *pointed structure* takes the form $\mathfrak{A}, \vec{a}$ with $\mathfrak{A}$ a structure and $\vec{a}$ a tuple of elements of $\mathrm{dom}(\mathfrak{A})$. A homomorphism from $\mathfrak{A}, \vec{a}$ to pointed structure $\mathfrak{B}, \vec{b}$ is a homomorphism h from $\mathfrak{A}$ to $\mathfrak{B}$ with $h(\vec{a}) = \vec{b}$. We write $\mathfrak{A}, \vec{a} \rightarrow \mathfrak{B}, \vec{b}$ if such a homomorphism exists.

4 Fundamental Results

We introduce the problem of (weak) separability in its projective and non-projective version. We then give a fundamental characterization of (FO, FO)-separability which has the consequence that UCQs have the same separating power as FO. This allows us to settle the complexity of deciding separability in GNFO.

Definition 1 *Let $\mathcal{L}$ be a fragment of FO. A* labeled $\mathcal{L}$-KB *takes the form $(\mathcal{K}, P, N)$ with $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ an $\mathcal{L}$-KB and $P, N \subseteq$ cons$(\mathcal{D})^n$ non-empty sets of* positive *and* negative examples*, all of them tuples of the same length n.*

An FO-formula $\varphi(\vec{x})$ with n free variables (weakly) separates *$(\mathcal{K}, P, N)$ if*

1. *$\mathcal{K} \models \varphi(\vec{a})$ for all $\vec{a} \in P$ and*
2. *$\mathcal{K} \not\models \varphi(\vec{a})$ for all $\vec{a} \in N$.*

Let $\mathcal{L}_S$ be a fragment of FO. We say that $(\mathcal{K}, P, N)$ is projectively $\mathcal{L}_S$-separable *if there is an $\mathcal{L}_S$-formula $\varphi(\vec{x})$ that separates $(\mathcal{K}, P, N)$ and* (non-projectively) $\mathcal{L}_S$-separable *if there is such a $\varphi(\vec{x})$ with* $\mathsf{sig}(\varphi) \subseteq \mathsf{sig}(\mathcal{K})$*.*

The following example illustrates the definition.

Example 1 Let $\mathcal{K}_1 = (\emptyset, \mathcal{D})$ where

$$\begin{array}{rcl} \mathcal{D} &=& \{\mathsf{born\_in}(a,c), \mathsf{citizen\_of}(a,c), \mathsf{born\_in}(b,c_1), \\ && \mathsf{citizen\_of}(b,c_2), \mathsf{Person}(a)\}. \end{array}$$

Then $\mathsf{Person}(x)$ separates $(\mathcal{K}_1, \{a\}, \{b\})$. As any citizen is a person, however, this separating formula is not natural and it only separates because of incomplete information about b. This may change with knowledge from the ontology. Let

$$\mathcal{O} = \{\forall x (\exists y (\mathsf{citizen\_of}(x,y)) \rightarrow \mathsf{Person}(x))\}$$

and $\mathcal{K}_2 = (\mathcal{O}, \mathcal{D})$. Then $\mathcal{K}_2 \models \mathsf{Person}(b)$ and so $\mathsf{Person}(x)$ no longer separates. However, the more natural formula

$$\varphi(x) = \exists y (\mathsf{born\_in}(x,y) \wedge \mathsf{citizen\_of}(x,y)),$$

separates $(\mathcal{K}_2, \{a\}, \{b\})$. Thus $(\mathcal{K}_2, \{a\}, \{b\})$ is non-projectively $\mathcal{L}$-separable for $\mathcal{L} = $ CQ and $\mathcal{L} = $ GF.

In the projective case, one admits symbols that are not from $\mathsf{sig}(\mathcal{K})$ as helper symbols in separating formulas. Their availability sometimes makes inseparable KBs separable. Note that in (Funk et al. 2019), helper symbols are generally admitted and the results depend on this assumption.

Example 2 The separating formula $\varphi(x)$ in Example 1 cannot be expressed as an $\mathcal{ALCI}$-concept. Using a helper concept name A, we obtain the separating concept

$$C = \forall \mathsf{born\_in}.A \rightarrow \exists \mathsf{citizen\_of}.A.$$

and thus $(\mathcal{K}_2, \{a\}, \{b\})$ is projectively $\mathcal{ALCI}$-separable. Note that C can be refuted at b because one can make A true at c_1 and false at c_2. For separation, it is thus important that A is not constrained by $\mathcal{O}$. Person is a concept name that, despite being in $\mathsf{sig}(\mathcal{K}_2)$, is also sufficiently unconstrained by $\mathcal{O}$ to act as a helper symbol: by replacing A by Person in C, one obtains a (rather unnatural) concept that witnesses also non-projective $\mathcal{ALCI}$-separability of $(\mathcal{K}_2, \{a\}, \{b\})$.

As we only study FO-fragments $\mathcal{L}_S$ that are closed under conjunction, a labeled KB $(\mathcal{K}, P, N)$ is (projectively) $\mathcal{L}_S$-separable if and only if all $(\mathcal{K}, P, \{\vec{b}\})$, $\vec{b} \in N$, are (projectively) $\mathcal{L}_S$-separable. In fact, a formula that separates $(\mathcal{K}, P, N)$ can be obtained by taking the conjunction of formulas that separate $(\mathcal{K}, P, \{\vec{b}\})$, $\vec{b} \in N$. We thus mostly consider labeled KBs with single negative examples.

Each choice of an ontology language $\mathcal{L}$ and a separation language $\mathcal{L}_S$ give rise to a separability problem and a projective separability problem, defined as follows.

PROBLEM :	(Projective) $(\mathcal{L}, \mathcal{L}_S)$-separability
INPUT :	A labeled $\mathcal{L}$-KB $(\mathcal{K}, P, N)$
QUESTION :	Is $(\mathcal{K}, P, N)$ (projectively) $\mathcal{L}_S$-separable?

We study both the *combined complexity* and the *data complexity* of separability. In the former, the full labeled KB $(\mathcal{K}, P, N)$ is taken as the input. In the latter, only $\mathcal{D}$ and the examples P, N are regarded as the input while $\mathcal{O}$ is assumed to be fixed.

Our first result provides a characterization of (FO, FO)-separability in terms of homomorphisms, linking it to UCQ-separability and in fact to UCQ evaluation on KBs.We first give some preliminaries. With every pointed database $\mathcal{D}, \vec{a}$, where $\vec{a} = (a_1, \ldots, a_n)$, we associate a CQ $\varphi_{\mathcal{D},\vec{a}}(\vec{x})$ with free variables $\vec{x} = (x_1, \ldots, x_n)$ that is obtained from $\mathcal{D}, \vec{a}$ as follows: view each $R(c_1, \ldots, c_m) \in \mathcal{D}$ as an atom $R(x_{c_1}, \ldots, x_{c_m})$, existentially quantify all variables x_c with $c \in \text{cons}(\mathcal{D}) \setminus [\vec{a}]$, replace every variable x_c such that $a_i = c$ for some i with the variable x_i such that i is minimal with $a_i = c$, and finally add $x_i = x_j$ whenever $a_i = a_j$. For a pointed database $\mathcal{D}, \vec{a}$, we write $\mathcal{D}_{\text{con}(\vec{a})}$ to denote the restriction of $\mathcal{D}$ to those constants that are reachable from some constant in $\vec{a}$ in the Gaifman graph of $\mathcal{D}$.

Theorem 1 *Let $(\mathcal{K}, P, \{\vec{b}\})$ be a labeled FO-KB, $\mathcal{K} = (\mathcal{O}, \mathcal{D})$. Then the following conditions are equivalent:*

1. *$(\mathcal{K}, P, \{\vec{b}\})$ is projectively UCQ-separable;*
2. *$(\mathcal{K}, P, \{\vec{b}\})$ is projectively FO-separable;*
3. *there exists a model $\mathfrak{A}$ of $\mathcal{K}$ such that for all $\vec{a} \in P$: $\mathcal{D}_{con(\vec{a})}, \vec{a} \not\to \mathfrak{A}, \vec{b}^{\mathfrak{A}}$;*
4. *the UCQ $\bigvee_{\vec{a}\in P} \varphi_{\mathcal{D}_{con(\vec{a})},\vec{a}}$ separates $(\mathcal{K}, P, \{\vec{b}\})$.*

Proof. "1 $\Rightarrow$ 2" and "4 $\Rightarrow$ 1" are trivial and "3 $\Rightarrow$ 4" is straightforward. We thus concentrate on "2 $\Rightarrow$ 3". Assume that $(\mathcal{K}, P, \{\vec{b}\})$ is separated by an FO-formula $\varphi(\vec{x})$. Then there is a model $\mathfrak{A}$ of $\mathcal{K}$ such that $\mathfrak{A} \not\models \varphi(\vec{b})$. Let $\vec{a} \in P$. Since $\mathcal{K} \models \varphi(\vec{a})$, there is no model $\mathfrak{B}$ of $\mathcal{K}$ and such that $\mathfrak{B}, \vec{a}^{\mathfrak{B}}$ and $\mathfrak{A}, \vec{b}^{\mathfrak{A}}$ are isomorphic, meaning that there is an isomorphism τ from $\mathfrak{B}$ to $\mathfrak{A}$ with $\tau(\vec{a}^{\mathfrak{B}}) = \vec{b}^{\mathfrak{A}}$. $\mathfrak{A}$ satisfies Condition 3. Assume to the contrary that there is a homomorphism h from $\mathcal{D}_{\text{con}(\vec{a})}, \vec{a}$ to $\mathfrak{A}, \vec{b}^{\mathfrak{A}}$ for some $\vec{a} \in P$. Let the structure $\mathfrak{B}$ by obtained from $\mathfrak{A}$ by setting $c^{\mathfrak{B}} = h(c)$ for all $c \in \text{cons}(\mathcal{D}_{\text{con}(\vec{a})})$ and $c^{\mathfrak{B}} = c^{\mathfrak{A}}$ for all remaining constants c. This construction relies on not making the UNA. $\mathfrak{B}$ is a model of $\mathcal{K}$ since $\mathcal{O}$ does not contain constants. It is easy to verify that $\mathfrak{B}, \vec{a}^{\mathfrak{B}}$ and $\mathfrak{A}, \vec{b}^{\mathfrak{A}}$ are isomorphic and thus we have obtained a contradiction. ❑

Note that the UCQ in Point 4 of Theorem 1 is a concrete separating formula. It is only of size polynomial in the size of the KB, but not very illuminating. It also contains no helper symbols[1] and thus we obtain the following.

Corollary 1 *$(FO, \mathcal{L}_S)$-separability coincides with projective $(FO, \mathcal{L}_S)$-separability for all FO-fragments $\mathcal{L}_S \supseteq$ UCQ. Moreover, $(FO, \mathcal{L}_S)$-separability coincides for all such $\mathcal{L}_S$.*

Theorem 1 also implies that for all $(\mathcal{L}, \mathcal{L}_S)$ with $\mathcal{L}$ a fragment of FO such that $\mathcal{L}_S \supseteq$ UCQ, $(\mathcal{L}, \mathcal{L}_S)$-separability can be mutually polynomially reduced with rooted UCQ evaluation on $\mathcal{L}$-KBs. This is the problem to decide, given a rooted UCQ q, an $\mathcal{L}$-KB $\mathcal{K} = (\mathcal{O}, \mathcal{D})$, and a tuple $\vec{a}$ of constants from $\mathcal{D}$, whether $\mathcal{K} \models q(\vec{a})$ (Baader et al. 2017). A connection of this kind was first observed in (Funk et al. 2019).

Since rooted UCQ evaluation on FO-KBs is undecidable, so is (FO, FO)-separability. However, rooted UCQ evaluation is decidable in 2ExpTime on GNFO-KBs (Bárány, ten Cate, and Segoufin 2015) and 2ExpTime-hardness is straightforward to show by reduction from satisfiability in GNFO. Since GNFO $\supseteq$ UCQ, we thus obtain the following.

Theorem 2 *$(GNFO, GNFO)$-separability coincides with $(GNFO, \mathcal{L}_S)$-separability for all FO-fragments $\mathcal{L}_S \supseteq$ UCQ. It further coincides with projective $(GNFO, \mathcal{L}_S)$-separability for all these $\mathcal{L}_S$ and is 2ExpTime-complete in combined complexity.*

We conjecture that the problems in Theorem 2 are 2ExpTime-complete also in data complexity, see Section 5.2 for further discussion in the context of GF.

We briefly mention the case of FO-separability of labeled KBs in which the ontology is empty. From the connection to rooted UCQ evaluation, it is immediate that this problem is coNP-complete. This is in contrast to GI-completeness of the FO-definability problem on closed world structures (Arenas and Diaz 2016).

5 Results on Separability

We study $(\mathcal{L}, \mathcal{L})$-separability for $\mathcal{L} \in \{\mathcal{ALCI}, \text{GF}, \text{FO}^2\}$. None of these fragments $\mathcal{L}$ contains UCQ, and thus we cannot use Theorem 1 in the same way as for GNFO above. All our results, in particular the lower bounds, also apply to the special case of GRE where the set P of positive examples is a singleton and P, N is a partition of $\text{cons}(\mathcal{D})$. The same is true for the special case of entity comparison where both P and N are singletons.

5.1 Separability of $\mathcal{ALCI}$-KBs

We are interested in separating labeled $\mathcal{ALCI}$-KBs $(\mathcal{K}, P, N)$ in terms of $\mathcal{ALCI}$-concepts which is relevant for concept learning, for generating referring expressions, and for entity comparison. Note that since $\mathcal{ALCI}$-concepts are FO-formulas with one free variable, positive and negative examples are single constants rather than proper tuples. Projective $(\mathcal{ALCI}, \mathcal{ALCI})$-separability has already been studied in (Funk et al. 2019) and thus we concentrate mainly on the non-projective case.

We start, however, with two observations on projective separability. It is shown in (Funk et al. 2019) that a labeled $\mathcal{ALCI}$-KB $(\mathcal{K}, P, N)$ is projectively $\mathcal{ALCI}$-separable

[1] In fact, it even contains only relation symbols that occur in $\mathcal{D}$ while symbols that only occur in $\mathcal{O}$ are not used.

iff Condition 4 from Theorem 1 holds. We thus obtain the following.[2]

Corollary 2 *Projective $(\mathcal{ALCI}, \mathcal{ALCI})$-separability coincides with $(\mathcal{ALCI}, \mathcal{L}_S)$-separability for all FO-fragments $\mathcal{L}_S \supseteq$ UCQ.*

It is proved in (Funk et al. 2019) that the separability problem from Corollary 2 is NExpTime-complete in combined complexity. It is also stated that it is Π_2^p-complete in data complexity, and that the same is the case for $(\mathcal{ALC}, \mathcal{ALC})$-separability. Unfortunately, though, the results on data complexity are incorrect. We start with correcting them.

Theorem 3 *Projective $(\mathcal{ALCI}, \mathcal{ALCI})$-separability is* NExpTime*-complete in data complexity and projective $(\mathcal{ALC}, \mathcal{ALC})$-separability is* PSpace*-complete in data complexity.*

The lower bounds are proved using reductions from a tiling problem and QBF validity, respectively. The upper bounds are by reduction to rooted UCQ-entailment on $\mathcal{ALC}(\mathcal{I})$-KBs with a fixed ontology.

We now turn to the main topic of this section, non-projective separability. We first observe that projective and non-projective separability are indeed different.

Example 3 *Let $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ be the $\mathcal{ALCI}$-KB where*

$$\begin{aligned} \mathcal{O} &= \{\top \sqsubseteq \exists R.\top \sqcap \exists R^-.\top\} \\ \mathcal{D} &= \{R(a,a), R(b,c)\}. \end{aligned}$$

Further let $P = \{a\}$ and $N = \{b\}$. Then the $\mathcal{ALCI}$-concept $A \rightarrow \exists R.A$ separates $(\mathcal{K}, P, N)$, using the concept name A as a helper symbol, and thus $(\mathcal{K}, P, N)$ is projectively $\mathcal{ALCI}$-separable.

In contrast, $(\mathcal{K}, P, N)$ is not non-projectively $\mathcal{ALCI}$-separable. In fact, every $\mathcal{ALCI}$-concept C with $\mathsf{sig}(C) = \{R\}$ is equivalent to $\top$ or to $\bot$ w.r.t. $\mathcal{O}$. Thus if $\mathcal{K} \models C(a)$, then $\mathcal{O} \models C \equiv \top$, and so $\mathcal{K} \models C(b)$.

Of course, Example 3 implies that an analogue of Corollary 2 fails for non-projective separability. In fact, it is easy to see that the labeled $\mathcal{ALCI}$-KB in Example 3, which is not $\mathcal{ALCI}$-separable, is separated by the CQ $R(x,x)$.

We next aim to characterize $(\mathcal{ALCI}, \mathcal{ALCI})$-separability in the style of Point 3 of Theorem 1. We start with noting that the ontology $\mathcal{O}$ used in Example 3 is very strong and enforces that all elements of all models of $\mathcal{O}$ are $\mathsf{sig}(\mathcal{K})$-bisimilar to each other. For ontologies that make such strong statements, symbols from outside of $\mathsf{sig}(\mathcal{K})$ might be required to construct a separating concept. It turns out that this is the only effect that distinguishes non-projective from projective separability. We next make this precise.

We use bisimulations between pointed structures, defined in the standard way but restricted to a signature Σ, see e.g. (Lutz, Piro, and Wolter 2011; Goranko and Otto 2007) for details. With $\mathfrak{A}, a \sim_{\mathcal{ALCI},\Sigma} \mathfrak{B}, b$, we indicate that there is a Σ-bisimulation between $\mathfrak{A}$ and $\mathfrak{B}$ that contains (a, b). For a KB $\mathcal{K}$, we use $\mathsf{cl}(\mathcal{K})$ to denote the set of concepts in $\mathcal{K}$ and the concepts $\exists R.\top$ and $\exists R^-.\top$ for all role names $R \in \mathsf{sig}(\mathcal{K})$, closed under subconcepts and single negation. A *$\mathcal{K}$-type* is a set $t \subseteq \mathsf{cl}(\mathcal{K})$ such that there exists a model $\mathfrak{A}$ of $\mathcal{K}$ and an $a \in \mathrm{dom}(\mathfrak{A})$ with $\mathrm{tp}_{\mathcal{K}}(\mathfrak{A}, a) = t$ where

$$\mathrm{tp}_{\mathcal{K}}(\mathfrak{A}, a) = \{C \in \mathsf{cl}(\mathcal{K}) \mid a \in C^{\mathfrak{A}}\}$$

is the *$\mathcal{K}$-type of a in* $\mathfrak{A}$. We say that a $\mathcal{K}$-type t is *connected* if $\exists R.\top \in t$ for some role R.

Definition 2 *A $\mathcal{K}$-type t is $\mathcal{ALCI}$-*complete *if for any two pointed models $\mathfrak{A}_1, b_1$ and $\mathfrak{A}_2, b_2$ of $\mathcal{K}$, $t = tp_{\mathcal{K}}(\mathfrak{A}_1, b_1) = tp_{\mathcal{K}}(\mathfrak{A}_2, b_2)$ implies $\mathfrak{A}_1, b_1 \sim_{\mathcal{ALCI},\mathsf{sig}(\mathcal{K})} \mathfrak{A}_2, b_2$.*

This is similar in spirit to the notion of a complete theory in classical logic (Chang and Keisler 1998). A type t is *realizable in $\mathcal{K}, b$*, where $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ and $b \in \mathrm{cons}(\mathcal{D})$, if there exists a model $\mathfrak{A}$ of $\mathcal{K}$ such that $\mathrm{tp}_{\mathcal{K}}(\mathfrak{A}, b^{\mathfrak{A}}) = t$.

Example 4 *(1) In Example 3, there is only a single $\mathcal{K}$-type and this type is $\mathcal{ALCI}$-complete.*

(2) Let $\mathcal{D}$ be a database and $\mathcal{O}_{\mathcal{D}}$ the ontology that contains all CIs that only use symbols from $\mathsf{sig}(\mathcal{D})$ and are true in the structure $\mathfrak{A}_{\mathcal{D}}$. This ontology is infinite, but easily seen to be equivalent to a finite ontology. Let $\mathcal{K} = (\mathcal{O}_{\mathcal{D}}, \mathcal{D})$. Then every $\mathcal{K}$-type is $\mathcal{ALCI}$-complete.

We are now in the position to formulate the characterization of non-projective $(\mathcal{ALCI}, \mathcal{ALCI})$-separability.

Theorem 4 *A labeled $\mathcal{ALCI}$-KB $(\mathcal{K}, P, \{b\})$ is non-projectively $\mathcal{ALCI}$-separable iff there exists a model $\mathfrak{A}$ of $\mathcal{K}$ such that for all $a \in P$:*

1. *$\mathcal{D}_{con(a)}, a \not\to \mathfrak{A}, b^{\mathfrak{A}}$ and*
2. *if $tp_{\mathcal{K}}(\mathfrak{A}, b^{\mathfrak{A}})$ is connected and $\mathcal{ALCI}$-complete, then $tp_{\mathcal{K}}(\mathfrak{A}, b^{\mathfrak{A}})$ is not realizable in $\mathcal{K}, a$.*

Proof. (idea) It is not difficult to show that $(\mathcal{K}, P, \{b\})$ is non-projectively $\mathcal{ALCI}$-separable iff there is a model $\mathfrak{A}$ of $\mathcal{K}$ such that for all models $\mathfrak{B}$ of $\mathcal{K}$ and all $a \in P$: $\mathfrak{B}, a^{\mathfrak{B}} \not\sim_{\mathcal{ALCI},\mathsf{sig}(\mathcal{K})} \mathfrak{A}, b^{\mathfrak{A}}$. One then proves that non-existence of a bisimilar $\mathfrak{B}, a^{\mathfrak{B}}$ can be equivalently replaced by non-existence of a homomorphism from $\mathcal{D}_{\mathrm{con}(a)}, a$ if $\mathrm{tp}_{\mathcal{K}}(\mathfrak{A}, b^{\mathfrak{A}})$ is not connected or not $\mathcal{ALCI}$-complete. ❑

Note that Point 1 of Theorem 4 is identical to Point 3 of Theorem 1 and that the characterization of projective $(\mathcal{ALCI}, \mathcal{ALCI})$-separability in (Funk et al. 2019) is as in Theorem 4 with Point 2 dropped.

In practice, one would expect that KBs $\mathcal{K}$ are such that no connected $\mathcal{K}$-type is $\mathcal{ALCI}$-complete (while every non-connected $\mathcal{K}$-type is necessarily $\mathcal{ALCI}$-complete). It thus makes sense to consider the following special case. A labeled $\mathcal{ALCI}$-KB $(\mathcal{K}, P, N)$ is *strongly incomplete* if no connected $\mathcal{K}$-type that is realizable in some $\mathcal{K}, b$, with $b \in N$, is $\mathcal{ALCI}$-complete. For $\mathcal{ALCI}$-KBs that are strongly incomplete, we can drop Point 2 from Theorem 4 and obtain the following from Theorem 1 and Corollary 2.

Corollary 3 *For labeled $\mathcal{ALCI}$-KBs that are strongly incomplete, non-projective $\mathcal{ALCI}$-separability coincides with non-projective and projective $\mathcal{L}_S$-separability for all FO-fragments $\mathcal{L}_S \supseteq$ UCQ.*

[2] The UNA is made in (Funk et al. 2019), but not in the current paper. This is inessential for $(\mathcal{ALCI}, \mathcal{ALCI})$-separability since $\mathcal{K} \models C(a)$ with UNA iff $\mathcal{K} \models C(a)$ without UNA if $\mathcal{K}$ is an $\mathcal{ALCI}$-KB and C an $\mathcal{ALCI}$-concept.

It follows from Theorem 4 that we can reduce projective $(\mathcal{ALCI},\mathcal{ALCI})$-separability to non-projective $(\mathcal{ALCI},\mathcal{ALCI})$-separability in polynomial time. Let $(\mathcal{K},P,\{b\})$, $\mathcal{K}=(\mathcal{O},\mathcal{D})$, be a labeled $\mathcal{ALCI}$-KB. Then $\mathcal{K}$ is projectively $\mathcal{ALCI}$-separable if and only if $(\mathcal{K}',P,\{b\})$ is non-projectively $\mathcal{ALCI}$-separable where $\mathcal{K}'=(\mathcal{O}',\mathcal{D})$ and $\mathcal{O}'=\mathcal{O}\cup\{A\sqsubseteq A\}$, A a fresh concept name. In fact, $\mathcal{K}$ is clearly projectively $\mathcal{ALCI}$-separable iff $\mathcal{K}'$ is, and $\mathcal{K}'$ is projectively $\mathcal{ALCI}$-separable iff it is non-projectively $\mathcal{ALCI}$-separable because no connected $\mathcal{K}'$-type is $\mathcal{ALCI}$-complete and thus Point 2 of Theorem 4 is vacuously true for $\mathcal{K}'$. This also implies that whenever a labeled $\mathcal{ALCI}$-KB is projectively separable, then a single fresh concept name suffices for separation.

We now have everything in place to clarify the complexity of non-projective $(\mathcal{ALCI},\mathcal{ALCI})$-separability.

Theorem 5 *Non-projective $(\mathcal{ALCI},\mathcal{ALCI})$-separability is* NExpTime-*complete in combined complexity and in data complexity.*

Proof. (sketch) The lower bound is a consequence of Theorem 3 and the mentioned reduction of projective separability to non-projective separability. For the upper bound, we first observe in the full version that it is ExpTime-complete to decide whether a given $\mathcal{K}$-type t is $\mathcal{ALCI}$-complete. Let $(\mathcal{K},P,\{b\})$ be a labeled $\mathcal{ALCI}$-KB. For any $\mathcal{K}$-type t, let $\mathcal{K}_t=(\mathcal{O}_t,\mathcal{D}_t)$ where $\mathcal{O}_t=\mathcal{O}\cup\{A\sqsubseteq\bigsqcap_{C\in t}C\}$ and $\mathcal{D}_t=\mathcal{D}\cup\{A(b)\}$ for a fresh concept name A. By Theorem 4, $(\mathcal{K},P,\{b\})$ is $\mathcal{ALCI}$-separable iff there exists a $\mathcal{K}$-type t that is realizable in $\mathcal{K},b$ such that (i) $\mathcal{K}_t\not\models\bigvee_{a\in P}\varphi_{\mathcal{D}_{\mathsf{con}(a)},a}(b)$ and (ii) if t is connected and $\mathcal{ALCI}$-complete, then t is not realizable in $\mathcal{K},a$ for any $a\in P$. The NExpTime upper bound now follows from the fact that rooted UCQ evaluation on $\mathcal{ALCI}$-KBs is in coNExpTime (complement of (i)) and that $\mathcal{ALCI}$-completeness of t and realizability of t in $\mathcal{K},a$ can be checked in ExpTime. ❑

When the ontology in $\mathcal{K}$ is empty, then no connected $\mathcal{K}$-type is $\mathcal{ALCI}$-complete and thus Point 2 of Theorem 4 is vacuously true. It follows that non-projective (and projective) $\mathcal{ALCI}$-separability of KBs $(\emptyset,\mathcal{D})$ coincides with FO-separability and is coNP-complete.

5.2 Separability of GF-KBs

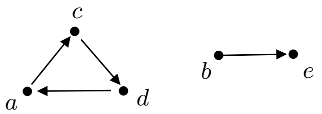
We study projective and non-projective $(\mathrm{GF},\mathrm{GF})$-separability which turns out to behave similarly to the $\mathcal{ALCI}$ case in many ways. The results are, however, significantly more difficult to establish.

We start with an example which shows that projective and non-projective $(\mathrm{GF},\mathrm{GF})$-separability do not coincide. Note that Example 3 does not serve this purpose since the labeled KB given there is separable by the GF-formula $R(x,x)$. We use the more succinct $\mathcal{ALCI}$-syntax for GF-formulas and ontologies whenever possible.

Example 5 *Define $\mathcal{K}=(\mathcal{O},\mathcal{D})$ where*

$$\mathcal{O}=\{\top\sqsubseteq\exists R.\top\sqcap\exists R^-.\top,\ \forall x\forall y(R(x,y)\rightarrow\neg R(y,x))\}$$
$$\mathcal{D}=\{R(a,c),R(c,d),R(d,a),R(b,e)\}$$

That is, $\mathcal{D}$ looks as follows:



The labeled GF-KB $(\mathcal{K},\{a\},\{b\})$ is separated by the $\mathcal{ALCI}$-concept $C=A\rightarrow\exists R.\exists R.\exists R.A$ that uses the concept name A as a helper symbol. In contrast, the KB is not non-projectively GF-separable since every GF-formula $\varphi(x)$ with $\mathsf{sig}(\varphi)=\{R\}$ is equivalent to $x=x$ or $\neg(x=x)$ w.r.t. $\mathcal{O}$.

To illustrate the role of the second sentence in $\mathcal{O}$, let $\mathcal{O}^-$ be $\mathcal{O}$ without this sentence. Then $\mathcal{K}^-=(\mathcal{O}^-,\mathcal{D})$ is separated by the GF-sentence obtained from the separating $\mathcal{ALCI}$-concept C above by replacing each occurrence of $A(x)$ in $C^\dagger$ by $\exists y(R(x,y)\wedge x\neq y\wedge R(y,y))$. We thus use a non-atomic formula in place of a helper symbol.

Let *openGF* be the fragment of GF that consists of all open formulas in GF whose subformulas are all open and in which equality is not used as a guard. OpenGF was first considered in (Hernich et al. 2020) where it is also observed that a GF formula is equivalent to an openGF formula if and only if it is invariant under disjoint unions. Informally, openGF relates to GF in the same way as $\mathcal{ALCI}$ relates to the extension of $\mathcal{ALCI}$ with the universal role (Baader et al. 2017). We start our investigation with observing the following.

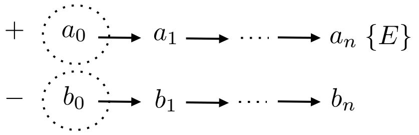
Theorem 6 *(GF,GF)-separability coincides with $(GF,openGF)$-separability, both in the projective and in the non-projective case.*

The proof of Theorem 6 uses guarded bisimulations between pointed structures, defined in the standard way (Grädel and Otto 2014), and openGF bisimulations as defined in (Hernich et al. 2020). With $\mathfrak{A},\vec{a}\sim_{\mathrm{openGF},\Sigma}\mathfrak{B},\vec{b}$, we indicate that there is a Σ-openGF-bisimulation between $\mathfrak{A}$ and $\mathfrak{B}$ that contains $(\vec{a},\vec{b})$. Arguably, openGF formulas are more natural for separation purposes than unrestricted GF formulas as they use only 'local' quantifiers and thus speak only about the neighbourhood of the examples. The next example shows that this is at the expense of larger separating formulas (a slightly modified example shows the same behaviour for $\mathcal{ALCI}$ and its extension with the universal role).

Example 6 *Let*

$$\mathcal{O}=\{A\sqsubseteq\forall R.A,\ \forall xy(R(x,y)\rightarrow\neg R(y,x))\}$$

and let $\mathcal{D}$ contain two R-paths of length n, $a_0Ra_1R\dots Ra_n$ and $b_0Rb_1R\dots Rb_n$ with a_n labeled with E:



Consider the labeled GF-KB $(\mathcal{K},\{a_0\},\{b_0\})$ with $\mathcal{K}=(\mathcal{O},\mathcal{D})$. Then the GF-formula $A(x)\rightarrow\exists y(A(y)\wedge E(y))$ separates $(\mathcal{K},\{a_0\},\{b_0\})$, but we show in the full version that the shortest separating openGF-formula has guarded quantifier rank n.

Let $\mathcal{K}=(\mathcal{O},\mathcal{D})$ be a GF-KB. For each $n\geq 1$, fix a tuple of distinct variables $\vec{x}_n$ of length n. We use $\mathsf{cl}(\mathcal{K})$ to denote the smallest set of GF-formulas that is closed under subformulas and single negation and contains: all formulas from $\mathcal{O}$; $x = y$ for distinct variables x, y; for all $R \in \mathsf{sig}(\mathcal{K})$ of arity n and all distinct $x, y \in [\vec{x}_n]$, the formulas $R(\vec{x}_n)$, $\exists \vec{y}_1\, (R(\vec{x}_n) \wedge x \neq y)$ where $\vec{y}_1$ is $\vec{x}_n$ without x, and $\exists \vec{y}_2\, R(\vec{x}_n)$ for all $\vec{y}_2$ with $[\vec{y}_2] \subseteq [\vec{x}_n] \setminus \{x, y\}$. Let $\mathfrak{A}$ be a model of $\mathcal{K}$ and $\vec{a}$ a tuple in $\mathfrak{A}$. The *$\mathcal{K}$-type of $\vec{a}$ in* $\mathfrak{A}$ is defined as

$$\mathrm{tp}_{\mathcal{K}}(\mathfrak{A}, \vec{a}) = \{\varphi \mid \mathfrak{A} \models \varphi(\vec{a}), \varphi \in \mathsf{cl}(\mathcal{K})[\vec{x}]\},$$

where $\mathsf{cl}(\mathcal{K})[\vec{x}]$ is obtained from $\mathsf{cl}(\mathcal{K})$ by substituting in any formula $\varphi \in \mathsf{cl}(\mathcal{K})$ the free variables of φ by variables in $\vec{x}$ in all possible ways, $\vec{x}$ a tuple of distinct variables of the same length as $\vec{a}$. Any such $\mathcal{K}$-type of some $\vec{a}$ in a model $\mathfrak{A}$ of $\mathcal{K}$ is called a *$\mathcal{K}$-type* and denoted $\Phi(\vec{x})$. A $\mathcal{K}$-type $\Phi(\vec{x})$ is *connected* if it contains a formula of the form $\exists \vec{y}_1\, (R(\vec{x}) \wedge x_i \neq x_j)$. It is *realizable in* $\mathcal{K}, \vec{b}$ if there exists a model $\mathfrak{A}$ of $\mathcal{K}$ with $\mathrm{tp}_{\mathcal{K}}(\mathfrak{A}, \vec{b}) = \Phi(\vec{x})$.

Definition 3 *Let $\mathcal{K}$ be a GF-KB. A $\mathcal{K}$-type $\Phi(\vec{x})$ is* openGF-complete *if for any two pointed models $\mathfrak{A}_1, \vec{b}_1$ and $\mathfrak{A}_2, \vec{b}_2$ of $\mathcal{K}$, $\Phi(\vec{x}) = tp_{\mathcal{K}}(\mathfrak{A}_1, \vec{b}_1) = tp_{\mathcal{K}}(\mathfrak{A}_2, \vec{b}_2)$ implies $\mathfrak{A}_1, \vec{b}_1 \sim_{openGF, \Sigma} \mathfrak{A}_2, \vec{b}_2$.*

In the labeled KB $\mathcal{K}$ from Example 5, there is only a single $\mathcal{K}$-type $\Phi_1(x)$ with free variable x and only a single $\mathcal{K}$-type $\Phi_2(x, y)$ with free variables x, y, and both of them are openGF-complete. In the KB $\mathcal{K}^-$ from the same example, there are multiple types of each kind and no connected type is openGF-complete.

We could now characterize non-projective (GF, GF)-separability in a way that is completely analogous to Theorem 4, replacing $\mathcal{ALCI}$-completeness of types with openGF-completeness. However, this works only for labeled KBs $(\mathcal{K}, P, \{\vec{b}\})$, $\mathcal{K} = (\mathcal{O}, \mathcal{D})$, such that all constants in $[\vec{b}]$ can reach one another in the Gaifman graph of $\mathcal{D}$. To formulate a condition for the general case, for a tuple $\vec{a} = (a_1, \ldots, a_n)$ and $I \subseteq \{1, \ldots, n\}$ let $\vec{a}_I = (a_i \mid i \in I)$.

Theorem 7 *A labeled GF-KB $(\mathcal{K}, P, \{\vec{b}\})$ with $\vec{b} = (b_1, \ldots, b_n)$ is non-projectively GF-separable iff there exists a model $\mathfrak{A}$ of $\mathcal{K}$ such that for all $\vec{a} \in P$:*

1. *$\mathcal{D}_{con(\vec{a})}, \vec{a} \not\to \mathfrak{A}, \vec{b}^{\mathfrak{A}}$ and*
2. *if the set I of all i such that $tp_{\mathcal{K}}(\mathfrak{A}, b_i^{\mathfrak{A}})$ is connected and openGF-complete is not empty, then*
   - *(a) $J = \{1, \ldots, n\} \setminus I \neq \emptyset$ and $\mathcal{D}_{con(\vec{a}_J)}, \vec{a}_J \not\to \mathfrak{A}, \vec{b}_J^{\mathfrak{A}}$ or*
   - *(b) $tp_{\mathcal{K}}(\mathfrak{A}, \vec{b}^{\mathfrak{A}})$ is not realizable in $\mathcal{K}, \vec{a}$.*

For projective GF-separability, Point 2 must be dropped.

In contrast to the case of $\mathcal{ALCI}$, the proof requires the careful use of bounded bisimulation and crucially relies on the fact that evaluating rooted UCQs on GF-KBs is finitely controllable (Bárány, Gottlob, and Otto 2014), a subject that is picked up again in the subsequent section.

Paralleling the case of $\mathcal{ALCI}$, we could now define a notion of strongly incomplete GF-KBs and observe a counterpart of Corollary 3. We refrain from giving the details. Also as for $\mathcal{ALCI}$, we can reduce projective (GF, GF)-separability to non-projective (GF, GF)-separability in polynomial time and show that a single unary helper symbol always suffices to separate a GF-KB that is projectively GF-separable. The following is an immediate consequence of Theorems 1 and 7.

Corollary 4 *Projective (GF, GF)-separability coincides with projective $(GF, \mathcal{L}_S)$-separability for all FO-fragments $\mathcal{L}_S \supseteq UCQ$.*

We obtain the following in a similar way as Theorem 5.

Theorem 8 *Projective and non-projective (GF, GF)-separability are* 2ExpTime*-complete in combined complexity.*

The lower bounds in Theorem 8 are by reduction from satisfiability in GF. We conjecture that the problems in Theorem 8 are 2ExpTime-complete also in data complexity. In fact, it seems possible but laborious to strengthen the proof from (Lutz 2008) that UCQ evaluation on $\mathcal{ALCI}$-KBs is 2ExpTime-hard so that it uses a fixed TBox; this would use similar ideas as the proof of Theorem 3. Moreover, it is not hard to reduce UCQ evaluation on $\mathcal{ALCI}$-KBs to rooted UCQ evaluation on GF-KBs in polynomial time. This would yield the conjectured result.

In the special case where the ontology is empty, Point 2 of Theorem 7 is vacuously true and thus projective and non-projective GF-separability coincide with FO-separability.

5.3 Separability of FO^2-KBs

We show that $(\mathrm{FO}^2, \mathrm{FO}^2)$- and $(\mathrm{FO}^2, \mathrm{FO})$-separability are undecidable both in the projective and in the non-projective case. We also show that these separation problems do not coincide even in the projective case, in contrast to our results on $\mathcal{ALCI}$ and GF in the previous sections. This in fact applies to all fragments of FO that have the finite model property, but for which UCQ evaluation is not finitely controllable. In the context of FO^2, we generally assume that examples are tuples of length one or two.

UCQ evaluation on FO^2-KBs is undecidable (Rosati 2007) and the proof easily adapts to rooted UCQs. Together with Theorem 1, we obtain undecidability of $(\mathrm{FO}^2, \mathrm{FO})$-separability both in the projective and non-projective case (which coincide, due to that theorem). The proof can further be adapted to projective and non-projective $(\mathrm{FO}^2, \mathrm{FO}^2)$-separability. It uses only a single positive example.

Theorem 9 *For $\mathcal{L} \in \{FO, FO^2\}$, projective and non-projective $(FO^2, \mathcal{L})$-separability is undecidable, even for labeled KBs with a single positive example.*

Example 5 shows that $(\mathrm{FO}^2, \mathrm{FO})$-separability and $(\mathrm{FO}^2, \mathrm{FO}^2)$-separability do not coincide in the non-projective case, since every FO^2-formula $\varphi(x)$ with $\mathsf{sig}(\varphi) = \{R\}$ is equivalent to $x = x$ or to $\neg(x = x)$ w.r.t. the ontology $\mathcal{O}$ used there. The example also yields that projective and non-projective $(\mathrm{FO}^2, \mathrm{FO}^2)$-separability do not coincide. We next show that $(\mathrm{FO}^2, \mathrm{FO})$-separability and $(\mathrm{FO}^2, \mathrm{FO}^2)$-separability do not coincide also in the projective case, in a more general setting.

Let $\mathcal{L}$ be a fragment of FO. Evaluating queries from a query language $Q \subseteq$ FO is *finitely controllable* on $\mathcal{L}$-KBs if for every $\mathcal{L}$-ontology $\mathcal{O}$, database $\mathcal{D}$, $\mathcal{L}$-formula $\varphi(\vec{x})$, tuple of constants $\vec{c}$, and model $\mathfrak{A}$ of $\mathcal{O}$ and $\mathcal{D}$ that satisfies $\mathfrak{A} \not\models \varphi(\vec{c})$, there is also a finite such model $\mathfrak{A}$. We further say that $\mathcal{L}$ has the *finite model property (FMP)* if evaluating queries from $\mathcal{L}$ is finitely controllable on $\mathcal{L}$-KBs. Finally, $\mathcal{L}$ has the *relativization property* (Chang and Keisler 1998) if for every $\mathcal{L}$-sentence φ and unary relation symbol $A \notin \mathsf{sig}(\varphi)$, there exists a sentence φ' such that for every structure $\mathfrak{A}$, $\mathfrak{A} \models \varphi'$ iff $\mathfrak{A}_{|A} \models \varphi$ where $\mathfrak{A}_{|A}$ is the $A^{\mathfrak{A}}$-reduct of $\mathfrak{A}$, that is, the restriction of $\mathfrak{A}$ to domain $A^{\mathfrak{A}}$.

FO^2 has the FMP and the relativization property, but evaluating rooted UCQs on FO^2 is not finitely controllable (Rosati 2007). The following theorem thus implies that projective (FO^2, FO)-separability does not coincide with projective $(\text{FO}^2, \text{FO}^2)$-separability.

Theorem 10 *Let $\mathcal{L}$ be a fragment of FO that has the relativization property and the FMP and such that projective $(\mathcal{L}, FO)$-separability coincides with projective $(\mathcal{L}, \mathcal{L})$-separability. Then evaluating rooted UCQs on $\mathcal{L}$-KBs is finitely controllable.*

When the ontology is empty, projective and non-projective FO^2-separability coincide with FO-separability.

6 Strong Separability

We introduce strong separability and give a characterization of strong (FO, FO)-separability that, in contrast to Theorem 1, establishes a link to KB unsatisfiability rather than to the evaluation of rooted UCQs. We also observe that strong projective separability and strong non-projective separability coincide in all relevant cases. We also settle the complexity of deciding strong separability in GNFO.

Definition 4 *An FO-formula $\varphi(\vec{x})$* strongly separates *a labeled FO-KB $(\mathcal{K}, P, N)$ if*

1. *$\mathcal{K} \models \varphi(\vec{a})$ for all $\vec{a} \in P$ and*
2. *$\mathcal{K} \models \neg\varphi(\vec{a})$ for all $\vec{a} \in N$.*

Let $\mathcal{L}_S$ be a fragment of FO. We say that $(\mathcal{K}, P, N)$ is strongly projectively $\mathcal{L}_S$-separable *if there is an $\mathcal{L}_S$-formula $\varphi(\vec{x})$ that strongly separates $(\mathcal{K}, P, N)$ and* strongly (non-projectively) $\mathcal{L}_S$-separable *if there is such a $\varphi(\vec{x})$ with* $\mathsf{sig}(\varphi) \subseteq \mathsf{sig}(\mathcal{K})$.

By definition, (projective) strong separability implies (projective) weak separability, but the converse is false.

Example 7 *Let $\mathcal{K}_1 = (\emptyset, \mathcal{D})$ with*

$$\mathcal{D} = \{\mathsf{votes}(a, c_1), \mathsf{votes}(b, c_2), \mathsf{Left}(c_1), \mathsf{Right}(c_2)\}.$$

Then $(\mathcal{K}_1, \{a\}, \{b\})$ is weakly separated by the $\mathcal{ALCI}$-concept $\exists\mathsf{votes}.\mathsf{Left}$*, but it is not strongly FO-separable.*

Now let $\mathcal{K}_2 = (\mathcal{O}, \mathcal{D})$ with

$$\mathcal{O} = \{\exists\mathsf{votes}.\mathsf{Left} \sqsubseteq \neg\exists\mathsf{votes}.\mathsf{Right}\}.$$

Then $\exists\mathsf{votes}.\mathsf{Left}$ *strongly separates $(\mathcal{K}_2, \{a\}, \{b\})$.*

As illustrated by Example 7, 'negative information' introduced by the ontology is crucial for strong separability because of the open world semantics and since the database cannot contain negative information. In fact, labeled KBs with an empty ontology are never strongly separable. In a sense, weak separability tends to be too credulous if the data is incomplete regarding positive information, see Example 1, while strong separability tends to be too sceptical if the data is incomplete regarding negative information as shown by Example 7.

For FO-fragments $\mathcal{L}_S$ closed under conjunction and disjunction, a labeled KB $(\mathcal{K}, P, N)$ is strongly (projectively) $\mathcal{L}_S$-separable iff every KB $(\mathcal{K}, \{\vec{a}\}, \{\vec{b}\})$ is, $\vec{a} \in P$ and $\vec{b} \in N$. In fact, if $\varphi_{\vec{a},\vec{b}}$ separates $(\mathcal{K}, \{a\}, \{b\})$ for all $\vec{a} \in P$ and $\vec{b} \in N$, then $\bigvee_{\vec{a} \in P} \bigwedge_{\vec{b} \in N} \varphi_{\vec{a},\vec{b}}$ separates $(\mathcal{K}, P, N)$. Note that this is the setup of entity comparison.

In contrast to weak separability, projective and non-projective separability coincide in all cases of strong separability that are relevant to this paper. From now on, we thus omit these qualifications.

Proposition 1 *Let $(\mathcal{K}, P, N)$ be an FO-KB and let $\mathcal{L}_S \in \{UCQ, \mathcal{ALCI}, GF, openGF, GNFO, FO^2, FO\}$. Then $(\mathcal{K}, P, N)$ is strongly projectively $\mathcal{L}_S$-separable iff it is strongly non-projectively $\mathcal{L}_S$-separable.*

The main observation behind Propositon 1 is that if a formula φ strongly separates a labeled KB $(\mathcal{K}, P, N)$ using some $R \notin \mathsf{sig}(\mathcal{K})$, then the formula φ' obtained from φ by replacing R by some $R' \in \mathsf{sig}(\mathcal{K})$ of the same arity also strongly separates $(\mathcal{K}, P, N)$.

Each choice of an ontology language $\mathcal{L}$ and a separation language $\mathcal{L}_S$ thus gives rise to a (single) strong separability problem that we refer to as *strong $(\mathcal{L}, \mathcal{L}_S)$-separability*, defined in the expected way. We next characterize strong (FO, FO)-separability in terms of KB unsatisfiability and show that strong (FO, FO)-separability coincides with strong (FO, UCQ)-separability. Let $\mathcal{D}$ be a database and let $\vec{a} = (a_1, \dots, a_n)$ and $\vec{b} = (b_1, \dots, b_n)$ be tuples of constants in $\mathcal{D}$. We write $\mathcal{D}_{\vec{a}=\vec{b}}$ to denote the database obtained by taking $\mathcal{D} \cup \mathcal{D}'$, $\mathcal{D}'$ a disjoint copy of $\mathcal{D}$, and then identifying a_i and b'_i for $1 \leq i \leq n$.

Theorem 11 *Let $(\mathcal{K}, P, N)$ be a labeled FO-KB, $\mathcal{K} = (\mathcal{O}, \mathcal{D})$. Then the following conditions are equivalent:*

1. *$(\mathcal{K}, P, N)$ is strongly UCQ-separable;*
2. *$(\mathcal{K}, P, N)$ is strongly FO-separable;*
3. *for all $\vec{a} \in P$ and $\vec{b} \in N$, the KB $(\mathcal{O}, \mathcal{D}_{\vec{a}=\vec{b}})$ is unsatisfiable;*
4. *the UCQ $\bigvee_{\vec{a} \in P} \varphi_{\mathcal{D}_{con(\vec{a}),\vec{a}}}$ strongly separates $(\mathcal{K}, P, N)$.*

Proof. "1 ⇒ 2", "2 ⇒ 3", and "4 ⇒ 1" are straightforward. It remains to prove "3 ⇒ 4". Thus assume that $\bigvee_{\vec{a} \in P} \varphi_{\mathcal{D}_{\text{con}(\vec{a}),\vec{a}}}$ does not strongly separate $(\mathcal{K}, P, N)$. Then there are a model $\mathfrak{A}$ of $\mathcal{K}$, $\vec{a} \in P$, and $\vec{b} \in N$ such that $\mathfrak{A} \models \varphi_{\mathcal{D}_{\text{con}(\vec{a}),\vec{a}}}(\vec{b}^{\mathfrak{A}})$. One can easily interpret the constants of $\mathcal{D}_{\vec{a}=\vec{b}}$ in such a way that $\mathfrak{A}$ becomes a model of $\mathcal{D}_{\vec{a}=\vec{b}}$. Thus the KB $(\mathcal{O}, \mathcal{D}_{\vec{a}=\vec{b}})$ is satisfiable. ❏

Note that the UCQ in Point 4 of Theorem 11 is a concrete separating formula of polynomial size, and that it is identical to the UCQ in Point 4 of Theorem 1. Point 3 provides the announced link to KB unsatisfiability. Such a connection was first observed in (Funk et al. 2019). Satisfiability of GNFO-KBs is 2ExpTime-complete in combined complexity and NP-complete in data complexity (Bárány, ten Cate, and Segoufin 2015; Bárány, ten Cate, and Otto 2012). This can be used to show the following.

Theorem 12 *Strong $(GNFO, GNFO)$-separability coincides with strong $(GNFO, \mathcal{L}_S)$-separability for all FO-fragments $\mathcal{L}_S \supseteq$ UCQ. It is* 2ExpTime*-complete in combined complexity and* coNP*-complete in data complexity.*

A slightly careful argument is needed to obtain the coNP lower bound for data complexity in the special case of GRE. For example, one can adapt the coNP-hardness proof from (Schaerf 1993) in a suitable way. The same is true for Theorems 14, 16, and 17 below.

7 Results on Strong Separability

We study strong $(\mathcal{L}, \mathcal{L})$-separability for $\mathcal{L} \in \{\mathcal{ALCI}, \text{GF}, \text{FO}^2\}$. For all these cases, strong $(\mathcal{L}, \mathcal{L})$-separability coincides with strong $(\mathcal{L}, \text{FO})$-separability and thus we can use the link to KB unsatisfiability provided by Theorem 11 to obtain decidability and tight complexity bounds. As in the case of weak separability, all results also apply to the special cases of GRE and of entity comparison.

7.1 Strong Separability of $\mathcal{ALCI}$-KBs

It has been shown in (Funk et al. 2019) that strong $(\mathcal{ALCI}, \mathcal{ALCI})$-separability is ExpTime-complete in combined complexity and coNP-complete in data complexity. Here, we add that strong $(\mathcal{ALCI}, \mathcal{ALCI})$-separability coincides with strong $(\mathcal{ALCI}, \text{FO})$-separability. With $\mathcal{K}$-types, we mean the types introduced for $\mathcal{ALCI}$ in Section 5.1. We identify a type with the conjunction of concepts in it.

Theorem 13 *For every labeled $\mathcal{ALCI}$-KB $(\mathcal{K}, P, N)$, the following conditions are equivalent:*

1. *$(\mathcal{K}, P, N)$ is strongly $\mathcal{ALCI}$-separable;*
2. *$(\mathcal{K}, P, N)$ is strongly FO-separable;*
3. *For all $a \in P$ and $b \in N$, there do not exist models $\mathfrak{A}$ and $\mathfrak{B}$ of $\mathcal{K}$ such that $a^{\mathfrak{A}}$ and $b^{\mathfrak{B}}$ realize the same $\mathcal{K}$-type;*
4. *The $\mathcal{ALCI}$-concept $t_1 \sqcup \cdots \sqcup t_n$ strongly separates $(\mathcal{K}, P, N)$, $t_1, \ldots, t_n$ the $\mathcal{K}$-types realizable in $\mathcal{K}, a$.*

Note that Point 4 of Theorem 13 provides concrete separating concepts. These are not illuminating, but of size at most $2^{p(||\mathcal{O}||)}$, p a polynomial. In contrast to the case of weak separability, the length of separating concepts is thus independent of $\mathcal{D}$.

Theorem 14 *Strong $(\mathcal{ALCI}, \mathcal{ALCI})$-separability coincides with strong $(\mathcal{ALCI}, \mathcal{L}_S)$-separability for all FO-fragments $\mathcal{L}_S \supseteq$ UCQ.*

7.2 Strong Separability of GF-KBs

We start with observing a counterpart of Theorem 6.

Theorem 15 *Strong (GF, GF)-separability coincides with strong $(GF, openGF)$-separability.*

The proof is based on bisimulations. We can next prove an analogue of Theorem 13, using $\mathcal{K}$-types for GF as defined in Section 5.2 in place of $\mathcal{K}$-types for $\mathcal{ALCI}$. An explicit formulation can be found in the full version. It follows that the size of strongly separating GF-formulas is at most $2^{2^{p(||\mathcal{O}||)}}$, p a polynomial, and thus does not depend on the database. Interestingly, we can use a variation of Example 6 to show that this is not the case for separating openGF-formulas. Details are given in the full version. Satisfiability of GF-KBs is 2ExpTime-complete in combined complexity and NP-complete in data complexity (Grädel 1999; Bárány, ten Cate, and Otto 2012). We obtain the following.

Theorem 16 *Strong (GF, GF)-separability coincides with strong $(GF, \mathcal{L}_S)$-separability for all FO-fragments $\mathcal{L}_S \supseteq$ UCQ. It is* 2ExpTime*-complete in combined complexity and* coNP*-complete in data complexity.*

7.3 Strong Separability of FO^2-KBs

We show that in contrast to weak separability, strong $(\text{FO}^2, \text{FO}^2)$-separability is decidable. The proof strategy is the same as for $\mathcal{ALCI}$ and GF and thus we first need a suitable notion of type for FO^2-KBs. Existing such notions, such as the types defined in (Grädel, Kolaitis, and Vardi 1997), are not strong enough for our purposes. For readers familiar with the model theory of FO^2, we remark that they do not record sufficient information about certain special elements in models sometimes referred to as *kings*. Fortunately, it is possible to define a sufficiently strong notion of type. We can then once more establish a theorem that parallels Theorem 13. As in the GF case, strongly separating formulas are of size at most $2^{2^{p(||\mathcal{O}||)}}$, p a polynomial. Since satisfiability of FO^2-KBs is NExpTime-complete in combined complexity and NP-complete in data complexity (Pratt-Hartmann 2009). We obtain the following.

Theorem 17 *Strong (FO^2, FO^2)-separability coincides with strong $(FO^2, \mathcal{L}_S)$-separability for all FO-fragments $\mathcal{L}_S \supseteq$ UCQ. It is* NExpTime*-complete in combined complexity and* coNP*-complete in data complexity.*

8 Conclusion

In this article and in (Funk et al. 2019), we have started an investigation of the separability problem for labeled KBs. Numerous questions remain to be addressed, including the following. What is the exact role of the UNA? What happens if (some) constants are admitted in the ontology or separating language? What happens if some symbols of the KB are not admitted in separating formulas? What is the size of separating formulas? What happens if one restricts the shape or size of separating formulas?

Acknowledgements

Carsten Lutz was supported by the DFG CRC Ease and Frank Wolter by EPSRC grant EP/S032207/1.

References

Andréka, H.; Németi, I.; and van Benthem, J. 1998. Modal languages and bounded fragments of predicate logic. *J. Philosophical Logic* 27(3):217–274.

Areces, C.; Koller, A.; and Striegnitz, K. 2008. Referring expressions as formulas of description logic. In *Proc. of INLG*. The Association for Computer Linguistics.

Arenas, M., and Diaz, G. I. 2016. The exact complexity of the first-order logic definability problem. *ACM Trans. Database Syst.* 41(2):13:1–13:14.

Arenas, M.; Diaz, G. I.; and Kostylev, E. V. 2016. Reverse engineering SPARQL queries. In *Proc. of WWW*, 239–249.

Baader, F.; Deborah; Calvanese, D.; McGuiness, D. L.; Nardi, D.; and Patel-Schneider, P. F., eds. 2003. *The Description Logic Handbook*. Cambridge University Press.

Baader, F.; Horrocks, I.; Lutz, C.; and Sattler, U. 2017. *An Introduction to Description Logics*. Cambride University Press.

Baader, F.; Küsters, R.; and Molitor, R. 1999. Computing least common subsumers in description logics with existential restrictions. In *Proc. of IJCAI*, 96–103.

Badea, L., and Nienhuys-Cheng, S. 2000. A refinement operator for description logics. In *Proc. of ILP*, 40–59.

Bárány, V.; Gottlob, G.; and Otto, M. 2014. Querying the guarded fragment. *Logical Methods in Computer Science* 10(2).

Bárány, V.; ten Cate, B.; and Otto, M. 2012. Queries with guarded negation. *PVLDB* 5(11):1328–1339.

Bárány, V.; ten Cate, B.; and Segoufin, L. 2015. Guarded negation. *J. ACM* 62(3):22:1–22:26.

Barceló, P., and Romero, M. 2017. The complexity of reverse engineering problems for conjunctive queries. In *Proc. of ICDT*, 7:1–7:17.

Borgida, A.; Toman, D.; and Weddell, G. E. 2016. On referring expressions in query answering over first order knowledge bases. In *Proc. of KR*, 319–328.

Bühmann, L.; Lehmann, J.; Westphal, P.; and Bin, S. 2018. DL-learner - structured machine learning on semantic web data. In *Proc. of WWW*, 467–471.

Bühmann, L.; Lehmann, J.; and Westphal, P. 2016. DL-learner - A framework for inductive learning on the semantic web. *J. Web Sem.* 39:15–24.

Chang, C., and Keisler, H. J. 1998. *Model Theory*. Elsevier.

Cohen, W. W.; Borgida, A.; and Hirsh, H. 1992. Computing least common subsumers in description logics. In *Proc. of AAAI*, 754–760.

Deutch, D., and Gilad, A. 2019. Reverse-engineering conjunctive queries from provenance examples. In *Proc. of EDBT*, 277–288.

Divroodi, A. R.; Ha, Q.; Nguyen, L. A.; and Nguyen, H. S. 2018. On the possibility of correct concept learning in description logics. *Vietnam J. Computer Science* 5(1):3–14.

Fanizzi, N.; Rizzo, G.; d'Amato, C.; and Esposito, F. 2018. DLFoil: Class expression learning revisited. In *Proc. of EKAW*, 98–113.

Fanizzi, N.; d'Amato, C.; and Esposito, F. 2008. DL-FOIL concept learning in description logics. In *Proc. of ILP*, 107–121.

Funk, M.; Jung, J. C.; Lutz, C.; Pulcini, H.; and Wolter, F. 2019. Learning description logic concepts: When can positive and negative examples be separated? In *Proc. of IJCAI*, 1682–1688.

Goranko, V., and Otto, M. 2007. Model theory of modal logic. In *Handbook of Modal Logic*. Elsevier. 249–329.

Grädel, E., and Otto, M. 2014. The freedoms of (guarded) bisimulation. In *Johan van Benthem on Logic and Information Dynamics*. Springer International Publishing. 3–31.

Grädel, E.; Kolaitis, P. G.; and Vardi, M. Y. 1997. On the decision problem for two-variable first-order logic. *Bulletin of Symbolic Logic* 3(1):53–69.

Grädel, E. 1999. On the restraining power of guards. *J. Symb. Log.* 64(4):1719–1742.

Gutiérrez-Basulto, V.; Jung, J. C.; and Sabellek, L. 2018. Reverse engineering queries in ontology-enriched systems: The case of expressive Horn description logic ontologies. In *Proc. of IJCAI-ECAI*.

Ha, Q.; Hoang, T.; Nguyen, L. A.; Nguyen, H. S.; Szalas, A.; and Tran, T. 2012. A bisimulation-based method of concept learning for knowledge bases in description logics. In *Proc. of SoICT*, 241–249. ACM.

Hernich, A.; Lutz, C.; Papacchini, F.; and Wolter, F. 2020. Dichotomies in ontology-mediated querying with the guarded fragment. *ACM Trans. Comput. Log.* 21(3):1–47.

Iannone, L.; Palmisano, I.; and Fanizzi, N. 2007. An algorithm based on counterfactuals for concept learning in the semantic web. *Appl. Intell.* 26(2):139–159.

Jung, J. C.; Lutz, C.; and Wolter, F. 2020. Least general generalizations in description logic: Verification and existence. In *Proc. of AAAI*, 2854–2861. AAAI Press.

Kalashnikov, D. V.; Lakshmanan, L. V.; and Srivastava, D. 2018. Fastqre: Fast query reverse engineering. In *Proc. of SIGMOD*, 337–350.

Kimelfeld, B., and Ré, C. 2018. A relational framework for classifier engineering. *ACM Trans. Database Syst.* 43(3):11:1–11:36.

Krahmer, E., and van Deemter, K. 2012. Computational generation of referring expressions: A survey. *Computational Linguistics* 38(1):173–218.

Lehmann, J., and Haase, C. 2009. Ideal downward refinement in the $\mathcal{EL}$ description logic. In *Proc. of ILP*, 73–87.

Lehmann, J., and Hitzler, P. 2010. Concept learning in description logics using refinement operators. *Machine Learning* 78:203–250.

Lisi, F. A., and Straccia, U. 2015. Learning in description logics with fuzzy concrete domains. *Fundamenta Informaticae* 140(3-4):373–391.

Lisi, F. A. 2012. A formal characterization of concept learning in description logics. In *Proc. of DL*.

Lisi, F. A. 2016. A model+solver approach to concept learning. In Adorni, G.; Cagnoni, S.; Gori, M.; and Maratea, M., eds., *Proc. of AI*IA 2016*, volume 10037 of *LNCS*, 266–279. Springer.

Lutz, C.; Piro, R.; and Wolter, F. 2011. Description logic TBoxes: Model-theoretic characterizations and rewritability. In *Proc. of IJCAI*.

Lutz, C. 2008. The complexity of conjunctive query answering in expressive description logics. In *Proc. of IJCAR*, 179–193.

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

Nebel, B. 1990. *Reasoning and Revision in Hybrid Representation Systems*. Springer.

Ortiz, M. 2019. Ontology-mediated queries from examples: a glimpse at the DL-Lite case. In *Proc. of GCAI*, 1–14.

Petrova, A.; Sherkhonov, E.; Grau, B. C.; and Horrocks, I. 2017. Entity comparison in RDF graphs. In *Proc. of ISWC*, 526–541.

Petrova, A.; Kostylev, E. V.; Grau, B. C.; and Horrocks, I. 2019. Query-based entity comparison in knowledge graphs revisited. In *Proc. of ISWC*, 558–575. Springer.

Pratt-Hartmann, I. 2009. Data-complexity of the two-variable fragment with counting quantifiers. *Inf. Comput.* 207(8):867–888.

Rosati, R. 2007. The limits of querying ontologies. In *Proc. of ICDT*, 164–178.

Sarker, M. K., and Hitzler, P. 2019. Efficient concept induction for description logics. In *Proc. of AAAI*, 3036–3043.

Schaerf, A. 1993. On the complexity of the instance checking problem in concept languages with existential quantification. *J. of Intel. Inf. Systems* 2:265–278.

Staworko, S., and Wieczorek, P. 2012. Learning twig and path queries. In *Proc. of ICDT*, 140–154. ACM.

Straccia, U., and Mucci, M. 2015. pFOIL-DL: Learning (fuzzy) $\mathcal{EL}$ concept descriptions from crisp OWL data using a probabilistic ensemble estimation. In *Proc. of SAC*, 345–352. ACM.

Toman, D., and Weddell, G. E. 2019. Finding ALL answers to OBDA queries using referring expressions. In *Proc. of AI*, 117–129. Springer.

Tran, Q. T.; Chan, C.; and Parthasarathy, S. 2009. Query by output. In *Proc. of PODS*, 535–548. ACM.

Tran, T.; Nguyen, L. A.; and Hoang, T. 2015. Bisimulation-based concept learning for information systems in description logics. *Vietnam J. Computer Science* 2(3):149–167.

Weiss, Y. Y., and Cohen, S. 2017. Reverse engineering spj-queries from examples. In *Proc. of PODS*, 151–166. ACM.

Zarrieß, B., and Turhan, A. 2013. Most specific generalizations w.r.t. general $\mathcal{EL}$-TBoxes. In *Proc. of IJCAI*, 1191–1197.

Zhang, M.; Elmeleegy, H.; Procopiuc, C. M.; and Srivastava, D. 2013. Reverse engineering complex join queries. In *Proc. of SIGMOD*, 809–820. ACM.