Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning
Main Track

Separating Data Examples by Description Logic Concepts with Restricted Signatures

Jean Christoph Jung[1] , **Carsten Lutz**[2] , **Hadrien Pulcini**[3] , **Frank Wolter**[3]
[1]Institute of Computer Science, University of Hildesheim, Germany
[2]Department of Computer Science, University of Bremen, Germany
[3]Department of Computer Science, University of Liverpool, UK
jungj@uni-hildesheim.de, clu@uni-bremen.de, {H.Pulcini,wolter}@liverpool.ac.uk

Abstract

We study the separation of positive and negative data examples in terms of description logic concepts in the presence of an ontology. In contrast to previous work, we add a signature that specifies a subset of the symbols that can be used for separation, and we admit individual names in that signature. We consider weak and strong versions of the resulting problem that differ in how the negative examples are treated and we distinguish between separation with and without helper symbols. Within this framework, we compare the separating power of different languages and investigate the complexity of deciding separability. While weak separability is shown to be closely related to conservative extensions, strongly separating concepts coincide with Craig interpolants, for suitably defined encodings of the data and ontology. This enables us to transfer known results from those fields to separability. Conversely, we obtain original results on separability that can be transferred backward. For example, rather surprisingly, conservative extensions and weak separability in $\mathcal{ALCO}$ are both 3ExpTime-complete.

1 Introduction

There are several applications that fall under the broad term of supervised learning and seek to compute a logical expression that separates positive from negative examples given in the form of labeled data items in a knowledge base (KB). A prominent example is concept learning for description logics (DLs) where inductive logic programming methods are applied to construct separating concepts that can then be used, for instance, in ontology engineering (Lehmann and Hitzler 2010). Another example is reverse engineering of database queries (or query by example, QBE) (Martins 2019) which has also been studied in the presence of a DL ontology (Gutiérrez-Basulto, Jung, and Sabellek 2018; Ortiz 2019). A closed world semantics is adopted for QBE in databases while an open world semantics is required in the presence of ontologies; the latter is the case also in reverse engineering of SPARQL queries (Arenas, Diaz, and Kostylev 2016). Further applications are entity comparison in RDF graphs, where one aims to find meaningful descriptions that separate one entity from another (Petrova et al. 2017; Petrova et al. 2019) and generating referring expressions (GRE) where the aim is to describe a single data item by a logical expression such as a DL concept, separating it from all other data items (Krahmer and van Deemter 2012; Borgida, Toman, and Weddell 2016).

A fundamental problem common to all these applications is to decide whether a separating formula exists at all. There are several degrees of freedom in defining this problem. The first concerns the negative examples: is it enough that they do not entail the separating formula (*weak separability*) or are they required to entail its negation (*strong separability*)? Another one concerns the question whether additional helper symbols are admitted in the separating formula (*projective separability*) or not (*non-projective separability*). The emerging family of problems has recently been investigated in (Funk et al. 2019; Jung et al. 2020), concentrating on the case where the separating expression is a DL concept or a formula from a fragment of first-order logic (FO) such as the guarded fragment (GF) and unions of conjunctive queries (UCQs).

In this paper, we add a signature Σ (set of concept, role, and individual names) that is given as an additional input and require separating expressions to be formulated in Σ. This makes it possible to 'direct' separation towards expressions based on desired features and accordingly to exclude features that are not supposed to be used for separation. For example, consider an online book store where a user has labeled some books with *likes* (positive examples) or *dislikes* (negative examples). A "good" separating expression might include relevant features of books like genre or language, but exclude information about the author's age or gender.

The aim of this paper is to investigate the effect of adding a signature to the framework, and in particular to compare the separating power of different languages and determine the computational complexity of deciding separability. We focus on the case in which both the knowledge base and the separating expressions are formulated in DLs between $\mathcal{ALC}$ and its extension $\mathcal{ALCIO}$ with inverse roles and nominals. DLs with nominals are of particular interest to us as separating expressions formulated in such DLs may refer to individual names in the signature Σ. Returning to the book store example, one can use the standard DL representation of specific authors ('Hemingway') and languages ('English') as individuals in separating expressions. To understand the robustness of our results, we also discuss in how far they extend to the guarded fragment (GF) and the two-variable fragment (FO^2) of FO.

We start with weak projective separability. We first observe that helper symbols, which must be 'fresh' in that they do not occur in the given knowledge base, increase the ability to separate and lead to more succinct separating expressions. We concentrate on the case where helper symbols are concept names because admitting individual names leads to undecidability of the separability problem while admitting roles names either does not make a difference (in $\mathcal{ALC}$ and $\mathcal{ALCI}$) or makes a difference but is polynomial time reducible to separation without role names as helper symbols (in $\mathcal{ALCO}$ and $\mathcal{ALCIO}$). To investigate further the relationship between non-projective and projective weak separability, we then introduce the extension of UCQs in which compound DL concepts are admitted in atoms and show that in some important cases, non-projective weak separability in that language coincides with projective weak separability in the original description logic. In thise sense, helper concept names are thus 'captured' by UCQs.

We next investigate the complexity of projective weak separability with signature for the DLs above. A fundamental observation is that, due to the presence of the signature, the problem to decide *projective conservative extensions* at the ontology level is polynomial time reducible to the complement of projective weak separability. Here, 'projective' refers to the fact that conservativity is also required for expressions using fresh concept names. Conservative extensions have been studied in detail in the context of modular ontologies (Grau et al. 2008; Botoeva et al. 2016). The projective version is motivated by the requirement of *robustness under vocabulary extensions* in applications with frequent changes in the ontology (Konev et al. 2009). It coincides with the non-projective one for DLs with the Craig Interpolation property (CIP) such as $\mathcal{ALC}$ and $\mathcal{ALCI}$ (Jung et al. 2017), but not for DLs with nominals, such as $\mathcal{ALCO}$.

The reduction from conservative extensions yields a 2ExpTime lower bound for weak projective separability in $\mathcal{ALC}$ and $\mathcal{ALCI}$ (Ghilardi, Lutz, and Wolter 2006; Lutz, Walther, and Wolter 2007). We prove a matching upper bound by providing a bisimulation-based characterization of weak projective separability which is then decided by a reduction to the emptiness problem of suitable tree automata.

For $\mathcal{ALCO}$, we show the unexpected result that both projective conservative extensions and projective weak separability are 3ExpTime-complete. The lower bound is a substantial extension of the 2ExpTime-lower bound for conservative extensions in $\mathcal{ALC}$ from (Ghilardi, Lutz, and Wolter 2006), and it holds for non-projective conservative extensions and non-projective weak separability as well. The upper bound is again by an encoding into tree automata.

We then turn to strong separability where we observe that the projective and non-projective case coincide. We further observe that separating expressions are identical to *Craig interpolants* between formulas that encode the KBs with the positive and negative examples, respectively. Since FO enjoys the CIP, the existence of FO separating formulas is equivalent to the entailment between the encoding formulas. This entailment question is ExpTime-complete if the KBs are given in a DL between $\mathcal{ALC}$ and $\mathcal{ALCIO}$. Moreover, any FO-theorem prover that computes interpolants can be used to compute separating expressions (Hoder et al. 2012).

Interestingly, while DL concepts alone have a strictly weaker separating power than FO, a version of the aforementioned extension of UCQs with DL concepts is expressive enough to capture the separating power of FO. Regarding the decision problem, we use recent results on the complexity of Craig interpolant existence (Artale et al. 2021) to show that for any DL between $\mathcal{ALC}$ and $\mathcal{ALCIO}$, strong separability is 2ExpTime-complete if one separates using concepts from the same DL.

We finally consider weak and strong inseparability in the case where both the ontology and the separating formulas are in GF or FO^2. For GF, we prove that weak (projective) separability with signature is undecidable which is in contrast to decidability of weak separability when no signature restriction can be imposed on the separating formula (Jung et al. 2020). For FO^2, already weak separability without signature is undecidable (Jung et al. 2020). In the case of strong separability, the link between Craig interpolants and strongly separating formulas generalizes to GF and FO^2. Both logics fail to have the CIP (Hoogland and Marx 2002; Comer 1969; Pigozzi 1971), but recent results on the existence of Craig interpolants can be used to prove that strong separability in GF is 3ExpTime-complete and in FO^2 is in N2ExpTime and 2ExpTime-hard (Jung and Wolter 2021).

An appendix with full proofs is available at https://arxiv.org/abs/2107.05285.

2 Preliminaries

Let $\mathsf{N_C}$, $\mathsf{N_R}$, and $\mathsf{N_I}$ be countably infinite sets of *concept*, *role*, and *individual names*. A *role* is a role name r or an *inverse role* r^-, with r a role name and $(r^-)^- = r$. A *nominal* takes the form $\{c\}$ with $c \in \mathsf{N_I}$. An *$\mathcal{ALCIO}$-concept* is defined according to the syntax rule

$$C, D ::= \top \mid \bot \mid A \mid \{c\} \mid \neg C \mid C \sqcap D \mid \exists R.C$$

where A ranges over concept names, c over individual names, and R over roles. We use $C \sqcup D$ as abbreviation for $\neg(\neg C \sqcap \neg D)$, $\forall R.C$ for $\neg\exists R.(\neg C)$, and $C \to D$ for $\neg C \sqcup D$. An *$\mathcal{ALCI}$-concept* is an $\mathcal{ALCIO}$-concept without nominals, an *$\mathcal{ALCO}$-concept* an $\mathcal{ALCIO}$-concept without inverse roles, and an *$\mathcal{ALC}$-concept* is an $\mathcal{ALCO}$-concept without nominals. Let $\mathsf{DL_{ni}}$ denote the set of languages just introduced, where ni stands for nominals and inverses. For $\mathcal{L} \in \mathsf{DL_{ni}}$, an *$\mathcal{L}$-ontology* is a finite set of *concept inclusion (CIs)* $C \sqsubseteq D$ with C and D $\mathcal{L}$-concepts.

A *database* $\mathcal{D}$ is a finite set of *facts* of the form $A(a)$ or $r(a, b)$ where $A \in \mathsf{N_C}$, $r \in \mathsf{N_R}$, and $a, b \in \mathsf{N_I}$. An *$\mathcal{L}$-knowledge base ($\mathcal{L}$-KB)* takes the form $\mathcal{K} = (\mathcal{O}, \mathcal{D})$, where $\mathcal{O}$ is an $\mathcal{L}$-ontology and $\mathcal{D}$ a database. We assume w.l.o.g. that any nominal used in $\mathcal{O}$ also occurs in $\mathcal{D}$.

A *signature* Σ is a set of concept, role, and individual names, uniformly referred to as *symbols*. Σ is called *relational* if it does not contain individual names. We use $\text{sig}(X)$ to denote the set of symbols used in any syntactic object X such as a concept or an ontology. For a database $\mathcal{D}$ we denote by $\text{ind}(\mathcal{D})$ the set of individual names in $\mathcal{D}$.

Description logics are interpreted in *structures*

$$\mathfrak{A} = (\text{dom}(\mathfrak{A}), (A^{\mathfrak{A}})_{A \in \mathsf{N_C}}, (r^{\mathfrak{A}})_{r \in \mathsf{N_R}}, (c^{\mathfrak{A}})_{c \in \mathsf{N_I}})$$

with $A^{\mathfrak{A}} \subseteq \mathrm{dom}(\mathfrak{A})$, $r^{\mathfrak{A}} \subseteq \mathrm{dom}(\mathfrak{A})^2$, and $c^{\mathfrak{A}} \in \mathrm{dom}(\mathfrak{A})$. The *extension* $C^{\mathfrak{A}}$ of $\mathcal{ALCIO}$-concepts C is then defined as usual (Baader et al. 2017). For $D \subseteq \mathrm{dom}(\mathfrak{A})$, we use $\mathfrak{A}|_D$ to denote the restriction of $\mathfrak{A}$ to D. A *pointed structure* takes the form $\mathfrak{A}, a$ with $\mathfrak{A}$ a structure and $a \in \mathrm{dom}(\mathfrak{A})$. A structure $\mathfrak{A}$ *satisfies* CI $C \sqsubseteq D$ if $C^{\mathfrak{A}} \subseteq D^{\mathfrak{A}}$, fact $A(a)$ if $a^{\mathfrak{A}} \in A^{\mathfrak{A}}$, and fact $r(a, b)$ if $(a^{\mathfrak{A}}, b^{\mathfrak{A}}) \in r^{\mathfrak{A}}$. $\mathfrak{A}$ is a *model* of an ontology, database, or KB if it satisfies all CIs and facts in it. A KB is *satisfiable* if it has a model, and a concept C is *satisfiable w.r.t. a KB* $\mathcal{K}$ if $\mathcal{K}$ has a model $\mathfrak{A}$ with $C^{\mathfrak{A}} \neq \emptyset$.

We use standard notation for first-order logic (FO), and consider formulas constructed using concept names as unary relation symbols, role names as binary relation symbols, and individual names as constants. Equality is admitted. It is well-known that every DL concept C is equivalent to an FO-formula $\varphi_C(x)$ with a single free variable x. For a KB $\mathcal{K}$, an FO-formula $\varphi(x)$ with a single free variable x, and a constant a, we write $\mathcal{K} \models \varphi(a)$ if $\mathfrak{A} \models \varphi(a)$ in all models $\mathfrak{A}$ of $\mathcal{K}$.

We associate with every structure $\mathfrak{A}$ a directed graph $G^d_{\mathfrak{A}} = (\mathrm{dom}(\mathfrak{A}), \bigcup_{r \in \mathsf{N_R}} r^{\mathfrak{A}})$. Let $G^u_{\mathfrak{A}} = (\mathrm{dom}(\mathfrak{A}), E')$ be the undirected version of $G^d_{\mathfrak{A}}$ obtained by forgetting edge directions. We can thus apply graph theoretic terminology to structures. The directed graph $G^d_{\mathfrak{A}}$ is relevant for the DLs $\mathcal{ALC}$ and $\mathcal{ALCO}$ that do not support inverse roles while the undirected graph $G^u_{\mathfrak{A}}$ is relevant for $\mathcal{ALCI}$ and $\mathcal{ALCIO}$. To simplify notation, we often prefix a property of structures with the language for which it is relevant. For example, if $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCO}\}$ then we say that $\mathfrak{A}$ has *finite $\mathcal{L}$-outdegree* if $G^d_{\mathfrak{A}}$ has and we call $\mathfrak{A}$ *$\mathcal{L}$-rooted in a* if every node in $\mathfrak{A}$ is reachable from a in $G^d_{\mathfrak{A}}$. For $\mathcal{L} \in \{\mathcal{ALCI}, \mathcal{ALCIO}\}$, the two notions are defined in the same way, but based on $G^u_{\mathfrak{A}}$ in place of $G^d_{\mathfrak{A}}$. For $\mathcal{L} \in \{\mathcal{ALCI}, \mathcal{ALCIO}\}$, $\mathfrak{A}$ is an *$\mathcal{L}$-tree* if $G^u_{\mathfrak{A}}$ is acyclic (also excluding self loops) and there are no multi-edges in the sense that $R_1^{\mathfrak{A}}$ and $R_2^{\mathfrak{A}}$ are disjoint for all distinct roles R_1, R_2. For $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCO}\}$, $\mathfrak{A}$ is an *$\mathcal{L}$-tree* if, in addition, every node in $G^d_{\mathfrak{A}}$ has at most one incoming edge.

Let $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$. A model $\mathfrak{A}$ of an $\mathcal{L}$-KB $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ is an *$\mathcal{L}$-forest model* of $\mathcal{K}$ if $\mathfrak{A}$ with all $r(a^{\mathfrak{A}}, b^{\mathfrak{A}})$, $a, b \in \mathrm{ind}(\mathcal{D})$, removed is the disjoint union of $\mathcal{L}$-trees rooted at $a^{\mathfrak{A}}$, $a \in \mathrm{ind}(\mathcal{D})$. $\mathfrak{A}$ is an *$\mathcal{ALCO}$-forest model* of an $\mathcal{ALCO}$-KB $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ if $\mathfrak{A}$ with all $r(a, b^{\mathfrak{A}})$, $a \in \mathrm{dom}(\mathfrak{A})$, $b \in \mathrm{ind}(\mathcal{D})$, removed is the disjoint union of $\mathcal{ALCO}$-trees rooted at $a^{\mathfrak{A}}$, $a \in \mathrm{ind}(\mathcal{D})$. The following completeness result is well-known (Baader et al. 2017).

Lemma 1 *Let $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}, \mathcal{ALCO}\}$, $\mathcal{K}$ an $\mathcal{L}$-KB, and C an $\mathcal{L}$-concept. If $\mathcal{K} \not\models C(a)$, then there exists an $\mathcal{L}$-forest model $\mathfrak{A}$ for $\mathcal{K}$ of finite $\mathcal{L}$-outdegree with $a \notin C^{\mathfrak{A}}$.*

Note that Lemma 1 does not hold for $\mathcal{ALCIO}$, a counterexample is given in the appendix.

Besides DL-concepts, we use FO-formulas with a single free variable as separating formulas. Of particular importance are the following FO-fragments which combine the expressive power of (unions of) conjunctive queries with DLs. Let $\mathcal{L} \in \mathsf{DL_{ni}}$. Then $\mathrm{CQ}^{\mathcal{L}}$ denotes the language of all FO-formulas $\varphi(x) = \exists \vec{y}\, \psi$ where ψ is a conjunction of atoms $C(t)$, C an $\mathcal{L}$-concept, or $r(t_1, t_2)$ with t, t_1, t_2 variables or constants, and x is the single free variable of $\varphi(x)$. $\mathrm{UCQ}^{\mathcal{L}}$ contains all formulas $\varphi(x) = \varphi_1(x) \vee \cdots \vee \varphi_n(x)$ with $\varphi_i(x) \in \mathrm{CQ}^{\mathcal{L}}$. Clearly, $\mathrm{CQ}^{\mathcal{L}}$ and $\mathrm{UCQ}^{\mathcal{L}}$ contain all unary conjunctive queries (CQ) and unions of unary conjunctive queries (UCQ), respectively. Note that $\mathrm{UCQ}^{\mathcal{ALCI}}$ is a fragment of the unary negation fragment (UNFO), a decidable fragment of FO that generalizes many modal and description logics (Segoufin and ten Cate 2013). We next define rooted versions of these languages. We may view formulas in $\mathrm{CQ}^{\mathcal{L}}$ as structures, in the obvious way by ignoring atoms $C(t)$. For $\mathcal{L} \in \mathsf{DL_{ni}}$, $\mathrm{CQ}^{\mathcal{L}}_r$ denotes the formulas $\varphi(x)$ in $\mathrm{CQ}^{\mathcal{L}}$ that are $\mathcal{L}$-rooted in x and similar for $\mathrm{UCQ}^{\mathcal{L}}_r$. Finally note that, although the languages $\mathrm{UCQ}^{\mathcal{L}}_r$ and $\mathrm{UCQ}^{\mathcal{L}}$ are not syntactically closed under conjunction, every conjunction is again equivalent to a formula in the respective language.

For any $\mathcal{L} \in \mathsf{DL_{ni}}$ and signature Σ the definition of an $\mathcal{L}(\Sigma)$-bisimulation S between structures $\mathfrak{A}$ and $\mathfrak{B}$ is standard, for details we refer to (Lutz, Piro, and Wolter 2011; Goranko and Otto 2007). We write $\mathfrak{A}, d \sim_{\mathcal{L},\Sigma} \mathfrak{B}, e$ and call pointed structures $\mathfrak{A}, d$ and $\mathfrak{B}, e$ $\mathcal{L}(\Sigma)$-*bisimilar* if there exists an $\mathcal{L}(\Sigma)$-bisimulation S such that $(d, e) \in S$. We say that $\mathfrak{A}, d$ and $\mathfrak{B}, e$ are $\mathcal{L}(\Sigma)$-*equivalent*, in symbols $\mathfrak{A}, d \equiv_{\mathcal{ALCI},\Sigma} \mathfrak{B}, e$ if $d \in C^{\mathfrak{A}}$ iff $e \in C^{\mathfrak{B}}$ for all $C \in \mathcal{L}(\Sigma)$; ω-saturated structures are defined and discussed in (Chang and Keisler 1998).

Lemma 2 *Let $\mathcal{L} \in \mathsf{DL_{ni}}$. Let $\mathfrak{A}, d$ and $\mathfrak{B}, e$ be pointed structures of finite $\mathcal{L}$-outdegree or ω-saturated and Σ a signature. Then*

$$\mathfrak{A}, d \equiv_{\mathcal{L},\Sigma} \mathfrak{B}, e \quad \textit{iff} \quad \mathfrak{A}, d \sim_{\mathcal{L},\Sigma} \mathfrak{B}, e.$$

For the "if" direction, the condition "finite outdegree or ω-saturated" can be dropped.

The definition of a Σ-homomorphism h from a structure $\mathfrak{A}$ to a structure $\mathfrak{B}$ is standard. Every database $\mathcal{D}$ gives rise to a finite structure $\mathfrak{A}_{\mathcal{D}}$ in the obvious way. A Σ-homomorphism from database $\mathcal{D}$ to structure $\mathfrak{A}$ is a Σ-homomorphism from $\mathfrak{A}_{\mathcal{D}}$ to $\mathfrak{A}$.

We combine homomorphisms and bisimulations to characterize the languages $\mathrm{CQ}^{\mathcal{L}}$ and $\mathrm{CQ}^{\mathcal{L}}_r$. Consider pointed structures $\mathfrak{A}, d$ and $\mathfrak{B}, e$, and a subset D of $\mathrm{dom}(\mathfrak{A})$ such that $d \in D \subseteq \mathrm{dom}(\mathfrak{A})$. Let $\mathcal{L} \in \mathsf{DL_{ni}}$ and Σ a signature. Then a *$\mathrm{CQ}^{\mathcal{L}}(\Sigma)$-homomorphism* with domain D between $\mathfrak{A}, d$ and $\mathfrak{B}, e$ is a Σ-homomorphism $h : \mathfrak{A}_{|D} \to \mathfrak{B}$ such that $h(d) = e$ and $\mathfrak{A}, c \sim_{\mathcal{L},\Sigma} \mathfrak{B}, h(c)$ for all $c \in D$. In this case we write $\mathfrak{A}, d \to_{D,\mathcal{L},\Sigma} \mathfrak{B}, e$.

We write $\mathfrak{A}, d \Rightarrow_{\mathrm{CQ}^{\mathcal{L}}_r,\Sigma} \mathfrak{B}, e$ if $\mathfrak{A} \models \varphi(a)$ implies $\mathfrak{B} \models \varphi(b)$ for all $\varphi(x)$ in $\mathrm{CQ}^{\mathcal{L}}_r(\Sigma)$, and we write $\mathfrak{A}, d \Rightarrow^{\mathrm{mod}}_{\mathrm{CQ}^{\mathcal{L}}_r,\Sigma} \mathfrak{B}, e$ if for all finite $D \subseteq \mathrm{dom}(\mathfrak{A})$ such that the Σ-reduct of $\mathfrak{A}_{|D}$ is $\mathcal{L}$-rooted in d, we have $\mathfrak{A}, d \to_{D,\mathcal{L},\Sigma} \mathfrak{B}, e$. The definitions for $\mathrm{CQ}^{\mathcal{L}}$ are analogous except that the Σ-reduct of $\mathfrak{A}_{|D}$ need not be $\mathcal{L}$-rooted in d.

Lemma 3 *Let $\mathcal{L} \in \mathsf{DL_{ni}}$ and let $\mathfrak{A}, d$ and $\mathfrak{B}, e$ be pointed structures of finite $\mathcal{L}$-outdegree or ω-saturated, and Σ a signature. Then*

$$\mathfrak{A}, d \Rightarrow_{\mathrm{CQ}^{\mathcal{L}}_r,\Sigma} \mathfrak{B}, e \quad \textit{iff} \quad \mathfrak{A}, d \Rightarrow^{\mathit{mod}}_{\mathrm{CQ}^{\mathcal{L}}_r,\Sigma} \mathfrak{B}, e.$$

This equivalence holds for $\mathbf{CQ}^{\mathcal{L}}$ if $\mathfrak{A}$ and $\mathfrak{B}$ are ω-saturated. In both cases, for the "if"-direction, the condition "finite outdegree or ω-saturated" can be dropped.

3 Weak Separability with Signature

We start with introducing the problem of (weak) separability with signature, in its projective and non-projective version. Let $\mathcal{L} \in \mathsf{DL}_{\mathsf{ni}}$. A *labeled $\mathcal{L}$-KB* takes the form $(\mathcal{K}, P, N)$ with $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ an $\mathcal{L}$-KB and $P, N \subseteq \text{ind}(\mathcal{D})$ non-empty sets of *positive* and *negative examples*.

Definition 1 *Let $\mathcal{L} \in \mathsf{DL}_{\mathsf{ni}}$, $(\mathcal{K}, P, N)$ be a labeled $\mathcal{L}$-KB, and let $\Sigma \subseteq sig(\mathcal{K})$ be a signature. An FO-formula $\varphi(x)$* Σ-separates *$(\mathcal{K}, P, N)$ if $sig(\varphi) \subseteq \Sigma \cup \Sigma_{help}$ for some set Σ_{help} of concept names disjoint from $sig(\mathcal{K})$ and*

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

Let $\mathcal{L}_S$ be a fragment of FO. We say that $(\mathcal{K}, P, N)$ is projectively $\mathcal{L}_S(\Sigma)$-separable *if there is an $\mathcal{L}_S$-formula $\varphi(x)$ that Σ-separates $(\mathcal{K}, P, N)$ and* non-projectively $\mathcal{L}_S(\Sigma)$-separable *if there is such a $\varphi(x)$ with $sig(\varphi) \subseteq \Sigma$.*[1]

In Σ-separating formulas, concept names from Σ_{help} should be thought of as helper symbols. Their availability sometimes makes inseparable KBs separable, examples are provided below where we also discuss the effect of admitting role or individual names as helper symbols. We only consider FO-fragments $\mathcal{L}_S$ that are closed under conjunction. In this case, a labeled KB $(\mathcal{K}, P, N)$ is $\mathcal{L}_S(\Sigma)$-separable if and only if all $(\mathcal{K}, P, \{b\})$, $b \in N$, are $\mathcal{L}_S(\Sigma)$-separable, and likewise for projective $\mathcal{L}_S(\Sigma)$-separability (Jung et al. 2020). In what follows, we thus mostly consider labeled KBs with singleton sets N of negative examples.

Each choice of an ontology language $\mathcal{L}$ and a separation language $\mathcal{L}_S$ give rise to a projective and to a non-projective separability problem.

PROBLEM:	(Projective) $(\mathcal{L}, \mathcal{L}_S)$-separability w. signature
INPUT:	A labeled $\mathcal{L}$-KB $(\mathcal{K}, P, N)$ and signature $\Sigma \subseteq \text{sig}(\mathcal{K})$
QUESTION:	Is $(\mathcal{K}, P, N)$ (projectively) $\mathcal{L}_S(\Sigma)$-separable?

If $\mathcal{L} = \mathcal{L}_S$, then we simply speak of (projective) $\mathcal{L}$-separability. We study the complexity of $\mathcal{L}$-separability with signature where the KB $\mathcal{K}$ and sets of examples P and N are all taken to be part of the input. All lower bounds proved in this paper still hold if P and N are singleton sets.

We next provide an example that illustrates the importance of the distinction between projective and non-projective separability.

Example 1 *Let $\mathcal{D}$ contain $r(a_1, a_2), \dots, r(a_{n-1}, a_n)$, $r(a_n, a_1)$ and $r(b, b_1)$, where $n > 1$. Thus, the individual a_1 is part of an r-cycle of length n but b is not. Let $\mathcal{O} = \{\top \sqsubseteq \exists r.\top \sqcap \exists r^-.\top\}$, $\mathcal{K} = (\mathcal{O}, \mathcal{D})$, $P = \{a_1\}$, $N = \{b\}$, and $\Sigma = \{r\}$. Then $(\mathcal{K}, P, N)$ is non-projectively $\mathbf{CQ}(\Sigma)$-separable (take the $\mathbf{CQ}$ that states that x participates in a cycle of length n), but $(\mathcal{K}, P, N)$ is not non-projectively $\mathcal{ALCI}(\Sigma)$-separable because for any $\mathcal{ALCI}(\Sigma)$-concept C either $\mathcal{O} \models \top \sqsubseteq C$ or $\mathcal{O} \models C \sqsubseteq \bot$. If, however, a helper symbol A is allowed, then $A \to \exists r^n.A$ Σ-separates $(\mathcal{K}, P, N)$.*

We discuss the effect of also admitting individual names as helper symbols. Then already for $\mathcal{ALC}$-KBs, projective inseparability becomes undecidable. The proof is inspired by reductions of undecidable tiling problems in the context of conservative extensions and modularity (Lutz, Walther, and Wolter 2007; Grau et al. 2008).

Theorem 1 *Projective $(\mathcal{ALC}, \mathcal{ALCO})$-separability with signature becomes undecidable when additionally individual names are admitted as helper symbols.*

Admitting role names as helper symbols has a less dramatic impact. For $\mathcal{ALC}$ and $\mathcal{ALCI}$-separability they do not make any difference at all and for $\mathcal{ALCO}$ and $\mathcal{ALCIO}$ their effect can be captured by a single additional role name which enables a straightforward polynomial reduction to separability without role names as helper symbols.

Theorem 2 *(1) Let $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$. Then projective $\mathcal{L}$-separability coincides with projective $\mathcal{L}$-separability with concept and role names as helper symbols.*

(2) Let $\mathcal{L} \in \{\mathcal{ALCO}, \mathcal{ALCIO}\}$ and $(\mathcal{K}, P, N)$ be a labeled $\mathcal{L}$-KB and $\Sigma \subseteq sig(\mathcal{K})$ a signature. Let r_I be a fresh role name and let $\mathcal{K}'$ be the extension of $\mathcal{K}$ by the 'dummy' inclusion $\exists r_I.\top \sqsubseteq \exists r_I.\top$. Then the following conditions are equivalent:

- *$(\mathcal{K}, P, N)$ is projectively $\mathcal{L}(\Sigma)$-separable with concept and role names as helper symbols;*
- *$(\mathcal{K}', P, N)$ is projectively $\mathcal{L}(\Sigma \cup \{r_I\})$-separable.*

The proof uses the model-theoretic characterization of separability given in Theorem 4 below. The next example illustrates the use of a helper role name in $\mathcal{ALCO}$.

Example 2 *Let $\mathcal{K} = (\mathcal{O}, \mathcal{D})$, where $\mathcal{O} = \{A_0 \sqcap \exists r.\top \sqsubseteq \bot, B \sqsubseteq \forall r.A\}$ and $\mathcal{D} = \{r(c, a), A_0(a), A_0(b)\}$. Let $\Sigma = \{c, B, A\}$. Then $(\mathcal{K}, \{a\}, \{b\})$ is not projectively $\mathcal{ALCIO}(\Sigma)$-separable, but the $\mathcal{ALCO}(\Sigma)$-concept $\exists r_I.(\{c\} \sqcap B) \to A$ separates $(\mathcal{K}, \{a\}, \{b\})$ using the helper symbol r_I.*

We next make first observations regarding the separating power of several relevant separating languages. In (Funk et al. 2019; Jung et al. 2020), projective and non-projective separability are studied without signature restrictions, that is, all symbols used in the KB except individual names can appear in separating formulas. We call this the *full relational signature*. Surprisingly, it turned out that in this case many different separation languages have exactly the same separating power. In particular, a labeled $\mathcal{ALCI}$-KB is FO-separable iff it is UCQ-separable, and projective and non-projective separability coincide. No such result can be expected for separability with signature restrictions, as illustrated by the next example.

[1] It is worth clarifying the interplay between nominals in the separating language and individual names in Σ: If Σ does not contain individual names, then $\mathcal{ALCO}(\Sigma)$-separability coincides with $\mathcal{ALC}(\Sigma)$-separability; conversely, if $\mathcal{L}_S$ does not allow for nominals, $\mathcal{L}_S(\Sigma)$-separability coincides with $\mathcal{L}_S(\Sigma \setminus \mathsf{N_I})$-separability.

Example 3 *Let $\mathcal{K} = (\mathcal{O}, \mathcal{D})$, where $\mathcal{O} = \{A \sqsubseteq \exists r.B \sqcap \exists r.\neg B\}$ and $\mathcal{D} = \{A(a), r(b, c)\}$. Let $P = \{a\}$, $N = \{b\}$, and $\Sigma = \{r\}$. Clearly, the formula*

$$\exists y \exists y' (r(x, y) \wedge r(x, y') \wedge \neg(y = y'))$$

Σ-separates $(\mathcal{K}, P, N)$, but $(\mathcal{K}, P, N)$ is not UCQ(Σ)-separable.

It is also shown in (Funk et al. 2019; Jung et al. 2020) that for labeled $\mathcal{ALCI}$-KBs and with the full relational signature, UCQ-separability (projectively or not) coincides with projective $\mathcal{ALCI}$-separability. The next example shows that with restricted signatures, it is not even true that non-projective $\mathcal{ALCI}$-separability implies UCQ-separability.

Example 4 *Let $\mathcal{O} = \{A \sqsubseteq \forall r.B\}$ and $\mathcal{D} = \{A(a), C(b)\}$. Let $P = \{a\}$, $N = \{b\}$, and $\Sigma = \{r, B\}$. Clearly, the $\mathcal{ALC}$-concept $\forall r.B$ Σ-separates $(\mathcal{O}, \mathcal{D}, P, N)$, but $(\mathcal{O}, \mathcal{D}, P, N)$ is not UCQ(Σ)-separable.*

Conversely, it follows from Example 1 that UCQ-separability does not imply non-projective $\mathcal{ALCI}$-separability, even with the full relational signature. Interestingly, in the projective case, this implication holds even with restricted signatures: every UCQ(Σ)-separable labeled $\mathcal{ALCI}$-KB is also projectively $\mathcal{ALCI}(\Sigma)$-separable. This follows from more powerful equivalences proved below (Theorem 5).

In this paper, we mainly focus on projective separability. In fact, it emerges from (Funk et al. 2019; Jung et al. 2020) that insisting on non-projective separability is a source of significant technical difficulties while not always delivering more natural separating concepts. As our main aim is to study the impact of signature restrictions on separability, which is another source of significant technical challenges, we prefer to leave out the first such source and stick with projective separability.

We close this introduction with the observation that in contrast to the case of full relational signatures, FO-separability with signature is undecidable for labeled $\mathcal{ALC}$-KBs. We prove this using the same technique as for Theorem 1. Undecidability applies even when one separates in the decidable extension $\mathcal{ALCFIO}$ of $\mathcal{ALCIO}$ with unqualified number restrictions of the form $(\leq 1\ r)$.

Theorem 3 *$(\mathcal{ALC}, \mathcal{L}_S)$-separability with signature is undecidable for any fragment $\mathcal{L}_S$ of FO that contains $\mathcal{ALCFIO}$, both in the projective and non-projective case.*

4 Model-Theoretic Criteria and Equivalence Results

We provide powerful model-theoretic criteria that underly the decision procedures given later on. Moreover, we use these criteria to establish equivalences between projective separability and non-projective separability in more expressive languages that shed light on the role of helper symbols.

We start with the model-theoretic criteria using *functional* bisimulations. For $\mathcal{L} \in \mathsf{DL_{ni}}$ we write $\mathfrak{A}, a \sim^f_{\mathcal{L},\Sigma} \mathfrak{B}, b$ if there exists an $\mathcal{L}(\Sigma)$-bisimulation S between $\mathfrak{A}$ and $\mathfrak{B}$ that contains (a, b) and is *functional*, that is, $(d, d_1), (d, d_2) \in S$ implies $d_1 = d_2$. Note that $\mathfrak{A}, a \sim^f_{\mathcal{L},\Sigma} \mathfrak{B}, b$ implies that there is a homomorphism from $\mathfrak{A}, a$ to $\mathfrak{B}, b$ if $\mathfrak{A}$ is connected and $\mathcal{L} = \mathcal{ALCI}$, but not otherwise.

Theorem 4 *Let $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}, \mathcal{ALCO}\}$. Assume that $(\mathcal{K}, P, \{b\})$ is a labeled $\mathcal{L}$-KB with $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ and $\Sigma \subseteq sig(\mathcal{K})$. Then the following conditions are equivalent:*

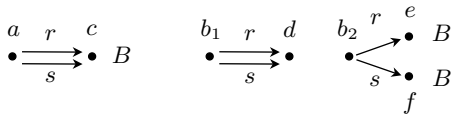
1. *$(\mathcal{K}, P, \{b\})$ is projectively $\mathcal{L}(\Sigma)$-separable.*
2. *there exists an $\mathcal{L}$-forest model $\mathfrak{A}$ of $\mathcal{K}$ of finite $\mathcal{L}$-outdegree and a set Σ_{help} of concept names disjoint from $sig(\mathcal{K})$ such that for all models $\mathfrak{B}$ of $\mathcal{K}$ and all $a \in P$: $\mathfrak{B}, a^{\mathfrak{B}} \not\sim_{\mathcal{L},\Sigma\cup\Sigma_{help}} \mathfrak{A}, b^{\mathfrak{A}}$.*
3. *there exists an $\mathcal{L}$-forest model $\mathfrak{A}$ of $\mathcal{K}$ of finite $\mathcal{L}$-outdegree such that for all models $\mathfrak{B}$ of $\mathcal{K}$ and all $a \in P$: $\mathfrak{B}, a^{\mathfrak{B}} \not\sim^f_{\mathcal{L},\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$.*

The equivalence between Points 1 and 2 of Theorem 4 is a direct consequence of the following characterization in the non-projective case (which can be proved using Lemmas 1 and 2): a labeled $\mathcal{L}$-KB $(\mathcal{K}, P, \{b\})$ is non-projectively $\mathcal{L}(\Sigma)$-separable iff there exists an $\mathcal{L}$-forest model $\mathfrak{A}$ of $\mathcal{K}$ of finite $\mathcal{L}$-outdegree such that for all models $\mathfrak{B}$ of $\mathcal{K}$ and all $a \in P$: $\mathfrak{B}, a^{\mathfrak{A}} \not\sim_{\mathcal{L},\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$. Due to cycles in the databases the general bisimulations used in this criterion and in Point 2 of Theorem 4 are hard to encode in an automata based decision procedure. Moreover, in Point 2 one has to "guess" the number of helper symbols needed. The criterion given in Point 3, in contrast, is much better suited for this purpose and does not speak about helper symbols.

The equivalence of 2. and 3. is surprisingly straightforward to show as one can work with the same model $\mathfrak{A}$. As Lemma 1 fails to hold for $\mathcal{L} = \mathcal{ALCIO}$, Theorem 4 also does not hold for this choice of $\mathcal{L}$. An example that illustrates the situation is given in the appendix.

As a first important application of Theorem 4, we show that projective $\mathcal{ALCI}$-separability is equivalent to non-projective separability in $\text{UCQ}_r^{\mathcal{ALCI}}$ and that projective $(\mathcal{ALC}, \mathcal{ALCO})$-separability is equivalent to non-projective $(\mathcal{ALC}, \text{UCQ}_r^{\mathcal{ALCO}})$-separability. The following example illustrates why the languages $\text{UCQ}_r^{\mathcal{L}}$ can non-projectively separate labeled KBs that cannot be separated non-projectively in a natural way in languages from $\mathsf{DL_{ni}}$.

Example 5 *Let $\mathcal{K} = (\mathcal{O}, \mathcal{D})$, where $\mathcal{O} = \{B \sqsubseteq \forall t.A\}$ and $\mathcal{D}$ is depicted below:*



Let $P = \{a\}, N = \{b_1, b_2\}$, and $\Sigma = \{r, s, t, A\}$. Then

$$\exists y\, r(x, y) \wedge s(x, y) \wedge (\forall t.A)(y) \in \mathbf{CQ}_r^{\mathcal{ALC}}$$

Σ-separates $(\mathcal{K}, P, N)$. The 'simplest' $\mathcal{ALC}$-concept Σ-separating $(\mathcal{K}, P, N)$ is $(\exists r.\forall t.A) \sqcap (\forall r.X \rightarrow \exists s.X)$, where X is fresh.

We next state the announced equivalences. Informally spoken, they show that admitting helper concept names corresponds to 'adding rooted UCQs'.

Theorem 5 *Let $(\mathcal{L}, \mathcal{L}_S)$ be either $(\mathcal{ALCI}, \mathcal{ALCI})$ or $(\mathcal{ALC}, \mathcal{ALCO})$ and let $(\mathcal{K}, P, \{b\})$ be a labeled $\mathcal{L}$-KB and $\Sigma \subseteq sig(\mathcal{K})$ a signature. Then the following conditions are equivalent:*

1. *$(\mathcal{K}, P, \{b\})$ is projectively $\mathcal{L}_S(\Sigma)$-separable;*
2. *$(\mathcal{K}, P, \{b\})$ is non-projectively $UCQ_r^{\mathcal{L}_S}(\Sigma)$-separable.*

Proof. The proof has two main steps. First, using Lemma 3, one can characterize non-projective $\text{UCQ}_r^{\mathcal{L}_S}(\Sigma)$-separability in terms of $\text{CQ}^{\mathcal{L}_S}(\Sigma)$-homomorphisms. Namely, $(\mathcal{K}, P, \{b\})$ is non-projectively $\text{UCQ}_r^{\mathcal{L}_S}(\Sigma)$-separable iff there exist an $\mathcal{L}_S$-forest model $\mathfrak{A}$ of $\mathcal{K}$ of finite $\mathcal{L}_S$-outdegree and $n > 0$ such that for all models $\mathfrak{B}$ of $\mathcal{K}$ and all $a \in P$, $\mathfrak{B}, a^{\mathfrak{B}} \not\to_{D,\mathcal{L}_S,\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$, for some D with $|D| \leq n$ such that the Σ-reduct of $\mathfrak{B}_{|D}$ is $\mathcal{L}_S$-rooted in $a^{\mathfrak{B}}$. Secondly, one can prove that this characterization is equivalent to Condition 3 of Theorem 4. Observe, for example, that functional Σ-bisimulations give rise to the combination of Σ-homomorphisms and Σ-bisimulations given in the characterization above. ❑

We observe that the equivalences of Theorem 5 do not hold when the ontology contains nominals.

Example 6 *Let $\mathcal{K} = (\mathcal{O}, \mathcal{D})$, where $\mathcal{O} = \{\{a\} \sqsubseteq \forall r.\{a\}, \top \sqsubseteq \exists r.\top\}$, and $\mathcal{D} = \{A(a), r(b, b)\}$. Let $\Sigma = \{r\}$. Then $(\mathcal{K}, \{a\}, \{b\})$ is projectively separated by the $\mathcal{ALC}(\Sigma)$-concept $X \to \forall r.X$ with X a fresh concept name, but it is not non-projectively $UCQ_r^{\mathcal{ALCO}}(\Sigma)$-separable.*

It remains open whether there is any natural fragment of FO such that a labeled $\mathcal{ALCO}$-KBs is non-projectively separable in the fragment if and only if it is projectively separable in $\mathcal{ALCO}$.

5 The Complexity of Weak Separability

We study the decidability and computational complexity of projective $\mathcal{L}$-separability for $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}, \mathcal{ALCO}\}$. The results established in this section are closely related to conservative extensions of ontologies and we also observe new results for that problem. For $\mathcal{L}$-ontologies $\mathcal{O}$ and $\mathcal{O}'$, we say that $\mathcal{O} \cup \mathcal{O}'$ is a *conservative extension of $\mathcal{O}$ in $\mathcal{L}$* if, for all concept inclusions $C \sqsubseteq D$ with C, D $\mathcal{L}$-concepts that use only symbols from $\text{sig}(\mathcal{O})$: if $\mathcal{O} \cup \mathcal{O}'$ entails $C \sqsubseteq D$ then already $\mathcal{O}$ entails $C \sqsubseteq D$. *Projective conservative extensions in $\mathcal{L}$* are defined in the same way except that C and D may additionally use fresh concept names, that is, concept names that are not in $\text{sig}(\mathcal{O} \cup \mathcal{O}')$. If $\mathcal{O} \cup \mathcal{O}'$ is not a conservative extension of $\mathcal{O}$ in $\mathcal{L}$, then there exists an $\mathcal{L}$-concept C that uses only symbols from $\text{sig}(\mathcal{O})$ and is satisfiable w.r.t. $\mathcal{O}$, but not w.r.t. $\mathcal{O} \cup \mathcal{O}'$. We call such a concept C a *witness concept* for $\mathcal{O}$ and $\mathcal{O}'$.

Lemma 4 *Let $\mathcal{L} \in \mathsf{DL}_{\mathsf{ni}}$. Then deciding conservative extensions in $\mathcal{L}$ can be reduced in polynomial time to the complement of $\mathcal{L}$-separability, both in the projective and non-projective case.*

Proof. The proof uses relativizations. Intuitively, given $\mathcal{O}$ and $\mathcal{O}'$ one computes a new ontology $\mathcal{O}_1$ which contains $\mathcal{O}$ and the relativization of $\mathcal{O}'$ to a fresh concept name A. Then, a concept C is a witness concept for $\mathcal{O}$ and $\mathcal{O}'$ iff $\neg C$ separates (w.r.t. $\mathcal{O}_1$) an individual that satisfies A from an individual that does not satisfy A. If $\mathcal{L}$ contains nominals the proof is slightly more involved. ❑

We start our analysis with the DLs $\mathcal{ALC}$ and $\mathcal{ALCI}$.

Theorem 6 *Projective $\mathcal{L}$-separability with signature is* 2ExpTime*-complete, for $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$.*

The lower bound follows from Lemma 4 and also holds for non-projective separability. In fact, it is known that deciding (non-projective) conservative extensions in $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$ is 2ExpTime-hard (Ghilardi, Lutz, and Wolter 2006; Lutz, Walther, and Wolter 2007) and that conservative extensions and projective conservative extensions coincide in logics that enjoy Craig interpolation (Jung et al. 2017), which $\mathcal{ALC}$ and $\mathcal{ALCI}$ do.

For the upper bound, we concentrate on $\mathcal{ALCI}$; the case of $\mathcal{ALC}$ is very similar, but simpler. The idea is to use two-way alternating tree automata (2ATA) (Vardi 1998) to decide Condition 3 of Theorem 4. More precisely, given $(\mathcal{K}, P, \{b\}), \Sigma$ with $\mathcal{K} = (\mathcal{O}, \mathcal{D})$, we construct a 2ATA $\mathcal{A}$ such that the language recognized by $\mathcal{A}$ is non-empty if and only if there is a forest model $\mathfrak{A}$ of $\mathcal{K}$ as described in Condition 3 of Theorem 4. The use of tree automata is enabled by the fact that Condition 3 refers to *forest models* of $\mathcal{K}$. Indeed, forest structures can be encoded in labeled trees using an appropriate alphabet. Intuitively, each node in the tree corresponds to an element in the forest structure and the label contains its type, the connection to its predecessor, and connections to individuals from $\mathcal{D}$.

It is not difficult to devise a 2ATA $\mathcal{B}$ (of polynomial size) that recognizes the finite outdegree forest models of $\mathcal{K}$, see e.g. (Jung et al. 2017). Observe next that it suffices to construct, for each $a \in P$, a 2ATA $\mathcal{A}_a$ such that $\mathcal{A}_a$ accepts $\mathfrak{A}$ iff

$(*_a)$ there is a model $\mathfrak{B}$ of $\mathcal{K}$ with $\mathfrak{B}, a^{\mathfrak{B}} \sim^f_{\mathcal{ALCI},\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$.

Indeed, a 2ATA that recognizes the following language is as required:

$$L(\mathcal{B}) \cap \bigcap_{a \in P} \overline{L(\mathcal{A}_a)}$$

where $\overline{L}$ denotes the complement of L. As complementation and intersection of 2ATAs involve only a polynomial blowup, we obtain the desired 2ATA $\mathcal{A}$ from $\mathcal{B}$ and the $\mathcal{A}_a$.

In principle, the existence of a (not necessarily functionally) bisimilar model $\mathfrak{B}$ can be checked using alternating automata as follows. We assume w.l.o.g. that the model $\mathfrak{B}$ is a forest model, because we can always consider an appropriate unraveling. Then, the alternating automaton 'virtually' traverses $\mathfrak{B}$ element-by-element, storing at each moment only the type of the current element in its state and visiting a bisimilar element in $\mathfrak{A}$. Alternation is crucial as the automaton has to extend the bisimulation *for all* successors of the current element in $\mathfrak{B}$ and symmetrically *for all* successors of the currently visited element in $\mathfrak{A}$. Functionality of the bisimulation poses a challenge: different parts of the run of the automaton can visit the same individual from $\mathcal{D}$ in $\mathfrak{B}$, and functionality requires that the automaton visits the same element in $\mathfrak{A}$. In order to solve that (and get tight bounds),

we replace $(*_a)$ with an equivalent condition in which functional bisimulations are carefully 'compiled away'.

We introduce some additional notation. An *extended database* is a database that additionally may contain 'atoms' of the form $C(a)$ with C an $\mathcal{ALCI}$-concept. The semantics of extended databases is defined in the expected way. Let $\text{sub}(\mathcal{K})$ denote the set of concepts that occur in $\mathcal{K}$, closed under single negation and under subconcepts. The *$\mathcal{K}$-type realized in a pointed structure* $\mathfrak{A}, a$ is defined as

$$\text{tp}_{\mathcal{K}}(\mathfrak{A}, a) = \{C \in \text{sub}(\mathcal{K}) \mid a \in C^{\mathfrak{A}}\}.$$

A *$\mathcal{K}$-type* is any set $t \subseteq \text{sub}(\mathcal{K})$ of the form $\text{tp}_{\mathcal{K}}(\mathfrak{A}, a)$. For a pointed database $\mathcal{D}, a$, we write $\mathcal{D}_{\text{con}(a)}, a \to_c^{\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$ if there is a Σ-homomorphism h from the maximal connected component $\mathcal{D}_{\text{con}(a)}$ of a in $\mathcal{D}$ to $\mathfrak{A}$ such that $h(a) = b^{\mathfrak{A}}$ and there is a $\mathcal{K}$-type t_d for each $d \in \text{ind}(\mathcal{D}_{\text{con}(a)})$ such that:

(i) there exists a model $\mathfrak{B}_d$ of $\mathcal{O}$ with $\text{tp}_{\mathcal{K}}(\mathfrak{B}_d, d) = t_d$ and $\mathfrak{B}_d, d \sim_{\mathcal{ALCI},\Sigma} \mathfrak{A}, h(d)$;

(ii) $(\mathcal{O}, \mathcal{D}')$ is satisfiable, for the extended database $\mathcal{D}' = \mathcal{D} \cup \{C(d) \mid C \in t_d,\ d \in \text{ind}(\mathcal{D}_{\text{con}(a)})\}$.

Lemma 5 *For all forest models $\mathfrak{A}$ of $\mathcal{K}$ and all $a \in P$, Condition $(*_a)$ is equivalent to $\mathcal{D}_{con(a)}, a \to_c^{\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$.*

Intuitively, the homomorphism h fixes the image of the bisimulation of the individuals from $\mathcal{D}$, and a 2ATA can decide $\mathcal{D}_{\text{con}(a)}, a \to_c^{\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$ as follows. It first nondeterministically guesses types t_d, $d \in \text{ind}(\mathcal{D}_{\text{con}(a)})$ that satisfy Item (ii) above and stores them in its states. Then it gradually guesses a Σ-homomorphism from $\mathcal{D}_{\text{con}(a)}$ to $\mathfrak{A}$. Whenever, it guesses a new image $h(d)$ for some d, it verifies the bisimulation condition in Item (i) as described above. Overall, $\mathcal{A}_a$ (and thus $\mathcal{A}$) uses exponentially many states. The 2ExpTime upper bound follows as non-emptiness can be decided in exponential time (Vardi 1998).

For $\mathcal{ALCO}$, we show the surprising result that separability becomes harder than in $\mathcal{ALC}$ and $\mathcal{ALCI}$, by one exponent. We establish the same result also for the more basic problem of conservative extensions.

Theorem 7 *Projective $\mathcal{ALCO}$-separability with signature and projective conservative extensions in $\mathcal{ALCO}$ are* 3ExpTime*-complete.*

We show in the appendix that the lower bound also applies to non-projective conservative extensions and, by Lemma 5, to non-projective $\mathcal{ALCO}$-separability with signature. An upper bound for that case remains open. The upper bound easily extends to the variant of projective conservative extensions where we are interested only in the entailment of concept inclusions $C \sqsubseteq D$ formulated in a given subsignature $\Sigma \subseteq \text{sig}(\mathcal{O})$, c.f. (Ghilardi, Lutz, and Wolter 2006).

By Lemma 4, it suffices to show the lower bound in Theorem 7 for conservative extensions and the upper bound for separability. We start with the former, which is proved by reduction of the word problem of double exponentially space bounded ATMs. The reduction strategy follows and extends the one used in (Ghilardi, Lutz, and Wolter 2006) to prove that deciding conservative extensions of $\mathcal{ALC}$-ontologies is 2ExpTime-complete. The reduction proceeds in two steps. First, for every $n \geq 1$ one crafts ontologies $\mathcal{O}_n$ and $\mathcal{O}'_n$ of size polynomial in n such that $\mathcal{O}_n \cup \mathcal{O}'_n$ is not a conservative extension of $\mathcal{O}_n$, but all witness concepts for $\mathcal{O}_n$ and $\mathcal{O}'_n$ are of size quadruple exponential in n. More precisely, $\mathcal{O}_n$ and $\mathcal{O}'_n$ implement a binary counter that is able to count the length of role paths up to $2^{2^{2^n}}$ and witness concepts need to enforce a binary tree of that depth. The triple exponential counter is implemented by building on a double exponential counter which in turn builds on a single exponential counter. The two latter counters are implemented exactly as in (Ghilardi, Lutz, and Wolter 2006) and the implementation of the triple exponential counter crucially uses a nominal. In fact, $\mathcal{O}_n$ does not use any nominals and a single nominal in $\mathcal{O}'_n$ suffices. The implementation of the counters is quite subtle. For the third counter, we independently send multiple $\mathcal{O}'_n$-types down a path in the binary tree generated by a witness concept and use the nominal to 're-synchronize' them again later. In the second step of the reduction, we simulate the computation of a fixed ATM on a given input in the binary trees of triple exponential depth generated by witness concepts for $\mathcal{O}_n$ and $\mathcal{O}'_n$.

For the upper bound in Theorem 7, we again pursue an automata-based approach. As for $\mathcal{ALCI}$, we encode forest structures as inputs to 2ATAs and the goal is to construct a 2ATA $\mathcal{A}_a$ that accepts an input $\mathfrak{A}$ if and only if Condition $(*_a)$ is true, with $\mathcal{ALCI}$ replaced by $\mathcal{ALCO}$. However, instead of going via an intermediate characterization such as Lemma 5, we directly use $(*_a)$ (at the cost of one exponent).

The problem of synchronizing different visits of the individuals in $\mathcal{D}$ during the (virtual) construction of $\mathfrak{B}$ is addressed as follows. We first construct a 2ATA $\mathcal{A}'_a$ over an extended alphabet. A labeled tree over that alphabet does not only contain the structure $\mathfrak{A}$, but also marks a *possible* choice of the elements in $\mathfrak{A}$ that are bisimilar to the individuals in $\mathcal{D}$. Now, when the automaton is in a state representing an individual $d \in \text{ind}(\mathcal{D})$ during the construction of $\mathfrak{B}$, it ensures that the currently visited element of $\mathfrak{A}$ is marked with d in the input. The desired automaton $\mathcal{A}_a$ is then obtained by projecting $\mathcal{A}'_a$ to the original input alphabet.

The 2ATA $\mathcal{A}'_a$ can be constructed in exponential time and has at most exponentially many states. Since projection of alternating automata involves an exponential blow-up, $\mathcal{A}_a$ is of double exponential size. Together with the exponential non-emptiness test, we obtain the 3ExpTime-upper bound.

6 Strong Separability with Signature

We discuss strong separability of labeled KBs. The crucial difference to weak separability is that the negation of the separating formula must be entailed at all negative examples.

Definition 2 *Let $\mathcal{L} \in \mathsf{DL}_{\mathsf{ni}}$, $(\mathcal{K}, P, N)$ be a labeled $\mathcal{L}$-KB, and let $\Sigma \subseteq \text{sig}(\mathcal{K})$ be a signature. An FO-formula $\varphi(x)$* strongly Σ-separates *$(\mathcal{K}, P, N)$ if $\text{sig}(\varphi) \subseteq \Sigma$ and*

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

Let $\mathcal{L}_S$ be a fragment of FO. We say that $(\mathcal{K}, P, N)$ is strongly $\mathcal{L}_S(\Sigma)$-separable *if there exists an $\mathcal{L}_S$-formula $\varphi(x)$ that strongly Σ-separates $(\mathcal{K}, P, N)$.*

In contrast to weak separability, we do not consider a projective version of strong separability as any formula φ that strongly Σ-separates a labeled KB $(\mathcal{K}, P, N)$ and uses helper symbols can easily be transformed into a strongly separating formula that uses only symbols from Σ: simply replace any occurrence of such a formula $A(x)$, $A \notin \Sigma$, by $x = x$ (or a concept name A by $\top$). Then, if φ strongly separates $(\mathcal{K}, P, N)$, so does the resulting formula φ'.

Note that for languages $\mathcal{L}_S$ closed under conjunction and disjunction a labeled KB $(\mathcal{K}, P, N)$ is strongly $\mathcal{L}_S(\Sigma)$-separable iff every $(\mathcal{K}, \{a\}, \{b\})$ with $a \in P$ and $b \in N$ is strongly $\mathcal{L}_S(\Sigma)$-separable. In fact, if $\varphi_{a,b}$ strongly separates $(\mathcal{K}, \{a\}, \{b\})$ for $a \in P$ and $b \in N$, then $\bigvee_{a \in P} \bigwedge_{b \in N} \varphi_{a,b}$ strongly separates $(\mathcal{K}, P, N)$. Without loss of generality, we may thus work with labeled KBs with singleton sets of positive and negative examples.

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:

PROBLEM :	Strong $(\mathcal{L}, \mathcal{L}_S)$ separability with signature
INPUT :	Labeled $\mathcal{L}$-KB $(\mathcal{K}, P, N)$ and signature $\Sigma \subseteq \text{sig}(\mathcal{K})$
QUESTION :	Is $(\mathcal{K}, P, N)$ strongly $\mathcal{L}_S(\Sigma)$-separable?

If $\mathcal{L} = \mathcal{L}_S$, then we simply speak of strong $\mathcal{L}$-separability. The study of strong separability is very closely linked to the study of interpolants and the Craig interpolation property (CIP). Given FO-formulas $\varphi(x), \psi(x)$ and a fragment $\mathcal{L}$ of FO, we say that an $\mathcal{L}$-formula $\chi(x)$ is an *$\mathcal{L}$-interpolant of φ, ψ* if $\varphi(x) \models \chi(x)$, $\chi(x) \models \psi(x)$, and $\text{sig}(\chi) \subseteq \text{sig}(\varphi) \cap \text{sig}(\psi)$. We say that $\mathcal{L}$ *has the CIP* if for all $\mathcal{L}$-formulas $\varphi(x), \psi(x)$ such that $\varphi(x) \models \psi(x)$, there exists an $\mathcal{L}$-interpolant of φ, ψ. FO and many of its fragments have the CIP (Craig 1957; ten Cate, Franconi, and Seylan 2013; Maksimova and Gabbay 2005). The link between interpolants and strongly separating formulas is easy to see: assume a labeled $\mathcal{L}$-KB $(\mathcal{K}, \{a\}, \{b\})$ and a signature $\Sigma \subseteq \text{sig}(\mathcal{K})$ are given. Obtain $\mathcal{K}_{\Sigma,a}$ (and $\mathcal{K}_{\Sigma,b}$) from $\mathcal{K}$ by taking the standard translation of $\mathcal{K}$ into FO and then

- replacing all concept and role names $X \notin \Sigma$ by fresh symbols X_a (X_b, respectively);
- replacing all individual names $c \notin \Sigma \cup \{a\}$ by fresh variables x_c (all $c \notin \Sigma \cup \{b\}$ by variables y_c, respectively);
- replacing a by x (and b by x, respectively) for a single fresh variable x;
- adding $x = a$ if $a \in \Sigma$ ($x = b$ if $b \in \Sigma$, respectively).

Let $\varphi_{\mathcal{K},\Sigma,a}(x) = \exists \vec{z}(\bigwedge \mathcal{K}_{\Sigma,a})$, where $\vec{z}$ is the sequence of free variables in $\mathcal{K}_{\Sigma,a}$ without the variable x and $(\bigwedge \mathcal{K}_{\Sigma,a})$ is the conjunction of all formulas in $\mathcal{K}_{\Sigma,a}$. $\varphi_{\mathcal{K},\Sigma,b}(x)$ is defined in the same way, with a replaced by b. The following lemma is a direct consequence of the construction.

Lemma 6 *Let $(\mathcal{K}, \{a\}, \{b\})$ be a labeled $\mathcal{L}$-KB, $\Sigma \subseteq \text{sig}(\mathcal{K})$ a signature, and $\mathcal{L}_S$ a fragment of FO. The following conditions are equivalent for any formula $\varphi(x)$ in $\mathcal{L}_S$:*

1. *φ strongly Σ-separates $(\mathcal{K}, \{a\}, \{b\})$;*
2. *φ is an $\mathcal{L}_S$-interpolant for $\varphi_{\mathcal{K},\Sigma,a}(x), \neg\varphi_{\mathcal{K},\Sigma,b}(x)$.*

Example 7 *To illustrate Lemma 6, let $\Sigma = \{r\}$ and $\mathcal{K} = (\mathcal{O}, \mathcal{D})$, with $\mathcal{O} = \{A \sqsubseteq \forall r.\neg A\}$ and $\mathcal{D} = \{A(a), r(b,b)\}$. Then, $\neg r(x,x)$ strongly Σ-separates $(\mathcal{K}, \{a\}, \{b\})$ and is an interpolant for $\varphi_{\mathcal{K},\Sigma,a}$, $\neg\varphi_{\mathcal{K},\Sigma,b}$ where $\varphi_{\mathcal{K},\Sigma,a}$, $\varphi_{\mathcal{K},\Sigma,b}$ are the following two formulas:*

$$\exists x_b\, r(x_b, x_b) \wedge A_a(x) \wedge \forall yz(r(y,z) \wedge A_a(y) \rightarrow \neg A_a(z)),$$
$$\exists y_a\, A_b(y_a) \wedge r(x,x) \wedge \forall yz\,(r(y,z) \wedge A_b(y) \rightarrow \neg A_b(z)).$$

Thus, the problem whether a labeled KB $(\mathcal{K}, P, N)$ is strongly $\mathcal{L}_S(\Sigma)$-separable and the computation of a strongly Σ-separating formula can be equivalently formulated as an interpolant existence problem. As FO has the CIP, we obtain the following characterization and complexity result for the existence of strongly FO(Σ)-separating formulas.

Theorem 8 *Let $\mathcal{L} \in \mathsf{DL_{ni}}$. The following conditions are equivalent for any $\mathcal{L}$-KB $(\mathcal{K}, \{a\}, \{b\})$ and signature $\Sigma \subseteq \text{sig}(\mathcal{K})$:*

1. *$(\mathcal{K}, \{a\}, \{b\})$ is strongly FO(Σ)-separable;*
2. *$\varphi_{\mathcal{K},\Sigma,a}(x) \models \neg\varphi_{\mathcal{K},\Sigma,b}(x)$.*

Strong $(\mathcal{L}, FO)$-separability with signature is ExpTime*-complete.*

The ExpTime upper bound follows from the fact that the complement of the problem to decide $\varphi_{\mathcal{K},\Sigma,a}(x) \models \neg\varphi_{\mathcal{K},\Sigma,b}(x)$ can be equivalently formulated as a concept satisfiability problem in the extension $\mathcal{ALCIO}^u$ of $\mathcal{ALCIO}$ with the universal role u. The lower bound can be proved by reduction of $\mathcal{ALC}$-KB satisfiability.

It follows from Theorem 8 that one can use FO theorem provers such as Vampire to compute strongly separating formulas (Hoder et al. 2012). FO is arguably too powerful, however, to serve as a useful separation language for labeled description logic KBs. Thus, two important questions arise: (1) which fragment of FO is needed to obtain a strongly separating formula in case that there is a strongly separating formula in FO? (2) What happens if the languages $\mathcal{L} \in \mathsf{DL_{ni}}$ are used as separation languages? For (1), one can show that none of the languages in UCQ$^{\mathcal{L}}$, $\mathcal{L} \in \mathsf{DL_{ni}}$, is sufficient to separate a and b in Example 7. We next show that the need for the negation of a CQ in that example is no accident. Indeed, by taking the closure BoCQ$^{\mathcal{ALCIO}}(\Sigma)$ of CQ$^{\mathcal{ALCIO}}(\Sigma)$ under negation, conjunction, and disjunction one obtains a sufficiently powerful language for (1), at least if the KB does not admit nominals.

Theorem 9 *The following conditions are equivalent for any labeled $\mathcal{ALCI}$-KB $(\mathcal{K}, P, N)$ and signature $\Sigma \subseteq sig(\mathcal{K})$.*

1. *$(\mathcal{K}, P, N)$ is strongly FO(Σ)-separable;*
2. *$(\mathcal{K}, P, N)$ is strongly BoCQ$^{\mathcal{ALCIO}}(\Sigma)$-separable.*

The proof of Theorem 9 uses the model-theoretic characterization given in Lemma 3 and techniques introduced in (Segoufin and ten Cate 2013). Problem (2) can be solved by using recent results about the complexity of deciding the existence of interpolants in DLs with nominals (Artale et al. 2021). Rather surprisingly, strong separability becomes one exponential harder than for FO. While the upper bounds are direct consequences of the results in (Artale et al. 2021), for the lower bounds one has to adapt the proofs.

Theorem 10 *Let $\mathcal{L} \in \mathsf{DL}_{\mathsf{ni}}$. Then strong $\mathcal{L}$-separability with signature is* 2EXPTIME*-complete.*

7 Separability with Signature in GF and FO^2

In the guarded fragment GF of FO quantification takes 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 $\alpha(\vec{x},\vec{y})$ is an atomic formula or an equality $x = y$ that contains all variables in $\vec{x},\vec{y}$ (Andréka, Németi, and van Benthem 1998; Hernich et al. 2020). The two-variable fragment, FO^2, is the fragment of FO with only two individual variables. For GF we admit relation symbols of arbitrary arity and equality, but no constant symbols. For FO^2 we make the same assumptions except that we admit relation symbols of arity one and two only. The definitions of weak projective and non-projective separability and of strong separability are the obvious extensions of the definitions given for description logics. Our results do not depend on whether one admits examples that are sets of tuples of constants of fixed but arbitary length or still only considers sets of constants.

Weak FO^2-separability is undecidable already with full relational signature, in both the projective and the non-projective case (Jung et al. 2020). For GF the situation is different: in both cases weak GF-separability is 2EXPTIME-complete, thus not harder than satisfiability. This result does not generalize to restricted signatures. In fact, by adapting the undecidability proof for conservative extensions given in (Jung et al. 2017), one can show the following.

Theorem 11 *Projective and non-projective $(\mathcal{L},\mathcal{L}_S)$-separability with signature are undecidable for all $(\mathcal{L},\mathcal{L}_S)$ such that $\mathcal{L}$ contains GF^3 and $\mathcal{L}_S$ contains $\mathcal{ALC}$.*

We now consider strong separability. For both FO^2 and GF the complexity of deciding strong separability with full relational signature is the same as validity, thus CONEXPTIME-complete and, respectively, 2EXPTIME-complete (Jung et al. 2020). With restricted signatures, the situation is different, and can again be analyzed in terms of interpolant existence. The formula $\varphi_{\mathcal{K},\Sigma,a}(x)$ constructed in Section 6 is not guaranteed to be in GF or FO^2 even if $\mathcal{K}$ is a GF or, respectively, FO^2-KB. It is, however, straightforward to construct formulas in the respective fragments that can serve the same purpose (either by using constants or by introducing a fresh relation symbol as a guard for $\mathcal{D}$ (for GF) and re-using variables (for FO^2)). Thus, strong separability in GF and FO^2-KBs is again equivalent to interpolant existence. Points 1 and 2 of the following theorem then follow from the CIP of FO and the complexity of GF and FO^2 (Grädel 1999; Grädel, Kolaitis, and Vardi 1997). Neither FO^2 nor GF have the CIP (Comer 1969; Pigozzi 1971; Hoogland and Marx 2002), thus separating in FO^2 and GF is less powerful than separating using FO. The complexity of interpolant existence for GF and FO^2 has recently been studied in (Jung and Wolter 2021) and the upper bounds in Points 3 and 4 follow directly from the complexity upper bounds for interpolant existence. The lower bounds are obtained by adapting the proofs.

Theorem 12 *1. Strong (GF,FO)-separability with signature is* 2EXPTIME*-complete;*

2. *Strong (FO^2, FO)-separability with signature is* CONEXPTIME*-complete;*
3. *Strong GF-separability is* 3EXPTIME*-complete, for relational signatures;*
4. *Strong FO^2-separability with signature is in* CON2EXPTIME *and* 2EXPTIME*-hard, for relational signatures.*

8 Discussion

We have started investigating separability of data examples under signature restrictions. Our main contributions are an analysis of the separating power of several important languages and the computational complexity of deciding separability. The following table gives an overview of the complexity of separability for expressive fragments of FO with and without signature restrictions. For Horn-DLs we refer the reader to (Funk et al. 2019; Jung, Lutz, and Wolter 2020). The results in the gray columns (weak, projective, with signature restriction and strong with signature restriction, respectively) are shown here, the results of the first (weak, projective, full signature), second (weak, non-projective, full signature), and fourth (strong and full signature) column are from (Funk et al. 2019; Jung et al. 2020).

	Weak Separability			Strong Separability	
$\mathcal{L}$	prj+full	full	prj+rstr	full	rstr
$\mathcal{ALC}$	NEXP	?	2EXP	EXP	2EXP
$\mathcal{ALCI}$	NEXP	NEXP	2EXP	EXP	2EXP
$\mathcal{ALCO}$	?	?	3EXP	?	2EXP
GF	2EXP	2EXP	Undec	2EXP	3EXP
FO^2	Undec	Undec	Undec	NEXP	$\leq$CON2EXP $\geq$2EXP

The missing entries for $\mathcal{ALCO}$ are due to the fact that nominals are considered for the first time in this article in the context of separability. We conjecture that the complexity is the same as for $\mathcal{ALC}$; note, however, that one has to be careful when defining separability problems in $\mathcal{ALCO}$ under the full signature as the individuals in the positive and negative examples should not be allowed in separating concepts.

Further interesting theoretical problems include: what is the complexity of weak projective separability with signature for $\mathcal{ALCIO}$, where the bisimulation characterization given in Theorem 4 does not hold? What is the complexity of non-projective weak separability with signature (and conservative extensions) for the DLs in $\mathsf{DL}_{\mathsf{ni}}$? From a practical viewpoint, it would be of interest to investigate systematically the size of separating concepts and to develop algorithms for computing them, if they exist. Recall that such an algorithm is already provided (by the relation of separating formulas to Craig interpolants) in the case of strong separability and it would be of interest to evaluate empirically the shape and size of Craig interpolants in FO in that case.

Acknowledgements

Carsten Lutz was supported by DFG CRC 1320 Ease. Frank Wolter was supported 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. Philos. Log.* 27(3):217–274.

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

Artale, A.; Jung, J. C.; Mazzullo, A.; Ozaki, A.; and Wolter, F. 2021. Living without Beth and Craig: Definitions and interpolants in description logics with nominals and role inclusions. In *Proc. of AAAI*.

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

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.

Botoeva, E.; Konev, B.; Lutz, C.; Ryzhikov, V.; Wolter, F.; and Zakharyaschev, M. 2016. Inseparability and conservative extensions of description logic ontologies: A survey. In *Proc. of Reasoning Web*, 27–89. Springer.

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

Comer, S. D. 1969. Classes without the amalgamation property. *Pacific J. Math.* 28(2):309–318.

Craig, W. 1957. Three uses of the herbrand-gentzen theorem in relating model theory and proof theory. *J. Symb. Log.* 22(3):269–285.

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.

Ghilardi, S.; Lutz, C.; and Wolter, F. 2006. Did I damage my ontology? A case for conservative extensions in description logics. In *Proc. of KR*, 187–197. AAAI Press.

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

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

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

Grau, B. C.; Horrocks, I.; Kazakov, Y.; and Sattler, U. 2008. Modular reuse of ontologies: Theory and practice. *J. Artif. Intell. Res.* 31:273–318.

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*.

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):20:1–20:47.

Hoder, K.; Holzer, A.; Kovács, L.; and Voronkov, A. 2012. Vinter: A vampire-based tool for interpolation. In Jhala, R., and Igarashi, A., eds., *Proc. of APLAS*, 148–156. Springer.

Hoogland, E., and Marx, M. 2002. Interpolation and definability in guarded fragments. *Stud. Log.* 70(3):373–409.

Jung, J. C., and Wolter, F. 2021. Living without Beth and Craig: Definitions and interpolants in the guarded and two-variable fragments. In *Proc. of LICS*.

Jung, J.; Lutz, C.; Martel, M.; Schneider, T.; and Wolter, F. 2017. Conservative extensions in guarded and two-variable fragments. In *Proc. of ICALP*, 108:1–108:14. Schloss Dagstuhl – LZI.

Jung, J. C.; Lutz, C.; Pulcini, H.; and Wolter, F. 2020. Logical separability of incomplete data under ontologies. In *Proc. of KR*.

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.

Konev, B.; Lutz, C.; Walther, D.; and Wolter, F. 2009. Formal properties of modularisation. In *Modular Ontologies*, volume 5445 of *Lecture Notes in Computer Science*. Springer. 25–66.

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

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

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

Lutz, C.; Walther, D.; and Wolter, F. 2007. Conservative extensions in expressive description logics. In *Proc. of IJCAI*, 453–458.

Maksimova, L., and Gabbay, D. 2005. *Interpolation and Definability in Modal and Intuitionistic Logics*. Clarendon Press.

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

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.

Pigozzi, D. 1971. Amalgamation, congruence-extension, and interpolation properties in algebras. *Algebra Univers.* (1):269–349.

Segoufin, L., and ten Cate, B. 2013. Unary negation. *Log. Methods Comput. Sci.* 9(3).

ten Cate, B.; Franconi, E.; and Seylan, I. 2013. Beth definability in expressive description logics. *J. Artif. Intell. Res.* 48:347–414.

Vardi, M. Y. 1998. Reasoning about the past with two-way automata. In *Proc. of ICALP*, 628–641.