# 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 $\Sigma$ (set of concept, role, and individual names) that is given as an additional input and require separating expressions to be formulated in $\Sigma$. 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 $\Sigma$. 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 ($\text{FO}^2$) 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 and then deciding the characterization 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 $\text{FO}^2$. 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 $\text{FO}^2$, 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 both GF and $\text{FO}^2$. Both languages 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 $\text{FO}^2$ is in N2ExpTime and 2ExpTime-hard (Jung and Wolter 2021).

## 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* $\Sigma$ is a set of concept, role, and individual names, uniformly referred to as *symbols*. $\Sigma$ 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 $A^{\mathfrak{A}} \subseteq \text{dom}(\mathfrak{A})$, $r^{\mathfrak{A}} \subseteq \text{dom}(\mathfrak{A})^2$, and $c^{\mathfrak{A}} \in \text{dom}(\mathfrak{A})$. The *extension* $C^{\mathfrak{A}}$ of $\mathcal{ALCIO}$-concepts $C$ is then defined as usual (Baader et al. 2017). For $D \subseteq \text{dom}(\mathfrak{A})$, 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 $a \in \text{dom}(\mathfrak{A})$. 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 $G^d_{\mathfrak{A}} = (\text{dom}(\mathfrak{A}), \bigcup_{r \in \mathsf{N_R}} r^{\mathfrak{A}})$. Let $G^u_{\mathfrak{A}} = (\text{dom}(\mathfrak{A}), E')$ 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 $R^{\mathfrak{A}}_1$ and $R^{\mathfrak{A}}_2$ 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 $r(a^{\mathfrak{A}}, b^{\mathfrak{A}})$, $a, b \in \text{ind}(\mathcal{D})$, removed is the disjoint union of $\mathcal{L}$-trees rooted at $a^{\mathfrak{A}}$, $a \in \text{ind}(\mathcal{D})$. $\mathfrak{A}$ is an *$\mathcal{ALCO}$-forest model* of an $\mathcal{ALCO}$-KB $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ if $\mathfrak{A}$ with all $r(a, b^{\mathfrak{A}})$, $a \in \text{dom}(\mathfrak{A})$, $b \in \text{ind}(\mathcal{D})$, removed is the disjoint union of $\mathcal{ALCO}$-trees rooted at $a^{\mathfrak{A}}$, $a \in \text{ind}(\mathcal{D})$. 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 $\text{CQ}^{\mathcal{L}}$ denotes the language of all FO-formulas $\varphi(x) = \exists \vec{y}\, \psi$ where $\psi$ 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 $\varphi(x)$. $\text{UCQ}^{\mathcal{L}}$ contains all formulas $\varphi(x) = \varphi_1(x) \vee \cdots \vee \varphi_n(x)$ with $\varphi_i(x) \in \text{CQ}^{\mathcal{L}}$. Clearly, $\text{CQ}^{\mathcal{L}}$ and $\text{UCQ}^{\mathcal{L}}$ contain all unary conjunctive queries (CQ) and unions of unary conjunctive queries (UCQ), respectively. Note that $\text{UCQ}^{\mathcal{ALCI}}$ 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 $\text{CQ}^{\mathcal{L}}$ as structures, in the obvious way by ignoring atoms $C(t)$. For $\mathcal{L} \in \mathsf{DL_{ni}}$, $\text{CQ}^{\mathcal{L}}_r$ denotes the formulas $\varphi(x)$ in $\text{CQ}^{\mathcal{L}}$ that are $\mathcal{L}$-rooted in $x$ and similar for $\text{UCQ}^{\mathcal{L}}_r$. Finally note that, although the languages $\text{UCQ}^{\mathcal{L}}_r$ and $\text{UCQ}^{\mathcal{L}}$ 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 $\Sigma$ 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 $C \in \mathcal{L}(\Sigma)$; $\omega$-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 $\omega$-saturated and $\Sigma$ 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 $\omega$-saturated" can be dropped.*

The definition of a $\Sigma$-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 $\Sigma$-homomorphism from database $\mathcal{D}$ to structure $\mathfrak{A}$ is a $\Sigma$-homomorphism from $\mathfrak{A}_{\mathcal{D}}$ to $\mathfrak{A}$.

We combine homomorphisms and bisimulations to characterize the languages $\text{CQ}^{\mathcal{L}}$ and $\text{CQ}^{\mathcal{L}}_r$. Consider pointed structures $\mathfrak{A}, d$ and $\mathfrak{B}, e$, and a subset $D$ of $\text{dom}(\mathfrak{A})$ such that $d \in D \subseteq \text{dom}(\mathfrak{A})$. Let $\mathcal{L} \in \mathsf{DL_{ni}}$ and $\Sigma$ a signature. Then a *$\text{CQ}^{\mathcal{L}}(\Sigma)$-homomorphism* with domain $D$ between $\mathfrak{A}, d$ and $\mathfrak{B}, e$ is a $\Sigma$-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 $\mathfrak{A}, d \Rightarrow_{\text{CQ}^{\mathcal{L}}_r,\Sigma} \mathfrak{B}, e$ if $\mathfrak{A} \models \varphi(a)$ implies $\mathfrak{B} \models \varphi(b)$ for all $\varphi(x)$ in $\text{CQ}^{\mathcal{L}}_r(\Sigma)$, and we write $\mathfrak{A}, d \Rightarrow^{\text{mod}}_{\text{CQ}^{\mathcal{L}}_r,\Sigma} \mathfrak{B}, e$ if for all finite $D \subseteq \text{dom}(\mathfrak{A})$ such that the $\Sigma$-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 $\text{CQ}^{\mathcal{L}}$ are analogous except that the $\Sigma$-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 $\omega$-saturated, and $\Sigma$ a signature. Then*

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

*This equivalence holds for $\mathbf{CQ}^{\mathcal{L}}$ if $\mathfrak{A}$ and $\mathfrak{B}$ are $\omega$-saturated. In both cases, for the "if"-direction, the condition "finite outdegree or $\omega$-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 $\varphi(x)$* $\Sigma$-separates *$(\mathcal{K}, P, N)$ if $sig(\varphi) \subseteq \Sigma \cup \Sigma_{help}$ for some set $\Sigma_{help}$ 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 $\Sigma$-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 $\Sigma$-separating formulas, concept names from $\Sigma_{\text{help}}$ 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 $r(a_1, a_2), \ldots, r(a_{n-1}, a_n)$, $r(a_n, a_1)$ 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 $A \to \exists r^n.A$ $\Sigma$-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 $\Sigma$: If $\Sigma$ 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*

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

*$\Sigma$-separates $(\mathcal{K}, P, N)$, but $(\mathcal{K}, P, N)$ is not UCQ$(\Sigma)$-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 $\forall r.B$ $\Sigma$-separates $(\mathcal{O}, \mathcal{D}, P, N)$, but $(\mathcal{O}, \mathcal{D}, P, N)$ is not UCQ$(\Sigma)$-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$(\Sigma)$-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:*

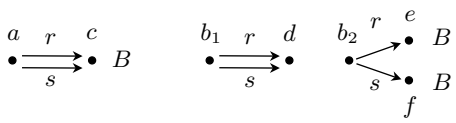
1. *$(\mathcal{K}, P, \{b\})$ is projectively $\mathcal{L}(\Sigma)$-separable.*
2. *there exists an $\mathcal{L}$-forest model $\mathfrak{A}$ of $\mathcal{K}$ of finite $\mathcal{L}$-outdegree and a set $\Sigma_{help}$ of concept names disjoint from $sig(\mathcal{K})$ such that for all models $\mathfrak{B}$ 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*

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

*$\Sigma$-separates $(\mathcal{K}, P, N)$. The 'simplest' $\mathcal{ALC}$-concept $\Sigma$-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 $\Sigma$-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 $\Sigma$-bisimulations give rise to the combination of $\Sigma$-homomorphisms and $\Sigma$-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 $\Sigma$-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 $\Sigma$-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 $\Sigma$-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 $\Sigma$-separates $(\mathcal{K}, P, N)$.*

In contrast to weak separability, we do not consider a projective version of strong separability as any formula $\varphi$ that strongly $\Sigma$-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 $\Sigma$: 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 $\varphi$ strongly separates $(\mathcal{K}, P, N)$, so does the resulting formula $\varphi'$.

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 $\varphi, \psi$* 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 $\varphi, \psi$. 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 fresh 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. Then the following conditions are equivalent for any formula $\varphi(x)$ in $\mathcal{L}_S$:*

1. *$\varphi$ strongly $\Sigma$-separates $(\mathcal{K}, \{a\}, \{b\})$;*
2. *$\varphi$ 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 $(\Sigma)$-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 $\Sigma$-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$(\Sigma)$-separating formulas.

**Theorem 8** *Let $\mathcal{L} \in \mathsf{DL}_{\mathsf{ni}}$. 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$(\Sigma)$-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 $\mathcal{L} \in \mathsf{DL}_{\mathsf{ni}}$ are used as separation languages? For (1), one can show that none of the languages in UCQ$^{\mathcal{L}}$, $\mathcal{L} \in \mathsf{DL}_{\mathsf{ni}}$, 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 $\Sigma \subseteq \text{sig}(\mathcal{K})$.*

1. *$(\mathcal{K}, P, N)$ is strongly FO$(\Sigma)$-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 comprehensively 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 in this article, the results of the first (weak, projective, and full signature), second (weak, non-projective, and full signature), and fourth (strong and full signature) column are shown in (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<br>$\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 that provide positive and negative counterexamples should be disallowed from 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.

Chandra, A. K.; Kozen, D. C.; and Stockmeyer, L. J. 1981. Alternation. *J. ACM* 28:114–133.

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.

Jung, J. C.; Lutz, C.; and Zeume, T. 2020. On the decidability of expressive description logics with transitive closure and regular role expressions. In *Proc. of KR*.

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.

# A Further Preliminaries

We remind the reader of different kinds of bisimulations that characterize the expressive power of the languages in $\mathsf{DL}_{\mathsf{ni}}$ (Lutz, Piro, and Wolter 2011; Goranko and Otto 2007). Let $\mathfrak{A}$ and $\mathfrak{B}$ be structures and $\Sigma$ a signature. A relation $S \subseteq \mathrm{dom}(\mathfrak{A}) \times \mathrm{dom}(\mathfrak{B})$ is an *$\mathcal{ALCO}(\Sigma)$-bisimulation between $\mathfrak{A}$ and $\mathfrak{B}$* if the following conditions hold:

1. if $(d, e) \in S$ and $A \in \Sigma$, then $d \in A^{\mathfrak{A}}$ iff $e \in A^{\mathfrak{B}}$;
2. if $(d, e) \in S$ and $c \in \Sigma$, then $d = c^{\mathfrak{A}}$ iff $e = c^{\mathfrak{B}}$;
3. if $(d, e) \in S$, $r \in \Sigma$, and $(d, d') \in r^{\mathfrak{A}}$, then there is an $e'$ with $(e, e') \in r^{\mathfrak{B}}$ and $(d', e') \in S$;
4. if $(d, e) \in S$, $r \in \Sigma$, $(e, e') \in r^{\mathfrak{B}}$, then there is a $d'$ with $(d, d') \in r^{\mathfrak{A}}$ and $(d', e') \in S$,

$S$ is an *$\mathcal{ALCIO}(\Sigma)$-bisimulation between $\mathfrak{A}$ and $\mathfrak{B}$* if Points 3 and 4 also hold for inverse roles over $\Sigma$. If $\Sigma$ is relational then we speak about $\mathcal{ALC}(\Sigma)$ and $\mathcal{ALCI}(\Sigma)$-bisimulations, respectively.

Let $\Sigma$ be a signature. A *$\Sigma$-homomorphism* $h$ from a structure $\mathfrak{A}$ to a structure $\mathfrak{B}$ is a function $h : \mathrm{dom}(\mathfrak{A}) \to \mathrm{dom}(\mathfrak{B})$ such that $a \in A^{\mathfrak{A}}$ implies $h(a) \in A^{\mathfrak{B}}$ for all $A \in \Sigma$, $(a, b) \in r^{\mathfrak{A}}$ implies $(h(a), h(b)) \in r^{\mathfrak{B}}$ for all $r \in \Sigma$, and $h(c^{\mathfrak{A}}) = c^{\mathfrak{B}}$ for all $c \in \Sigma$.

# B Proofs for Section 2

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

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

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

**Proof.** Assume first that $\mathfrak{A}, a \Rightarrow^{\mathrm{mod}}_{\mathrm{CQ}^{\mathcal{L}}_r, \Sigma} \mathfrak{B}, b$ and let $\varphi(x)$ be a formula in $\mathrm{CQ}^{\mathcal{L}}_r(\Sigma)$ such that $\mathfrak{A} \models \varphi(a)$. Then there exists a mapping $h$ from the set $\mathrm{var}(\varphi)$ of variables in $\varphi(x)$ to $\mathfrak{A}$ such that $h(x) = a$ and

- If $r(y, z)$ is a conjunct of $\varphi(x)$, then $(h(y), h(z)) \in r^{\mathfrak{A}}$;
- If $C(y)$ is a conjunct of $\varphi(x)$, then $h(y) \in C^{\mathfrak{A}}$.

Let $D$ be the image of $\mathrm{var}(\varphi)$ under $h$. Then the $\Sigma$-reduct of $\mathfrak{A}_{|D}$ is $\mathcal{L}$-rooted in $a$ and, by definition of $\mathfrak{A}, a \Rightarrow^{\mathrm{mod}}_{\mathrm{CQ}^{\mathcal{L}}, \Sigma} \mathfrak{B}, b$, we have a $\Sigma$-homomorphism $h'$ from $\mathfrak{A}_{|D}$ to $\mathfrak{B}$ such that $h'(a) = b$ and $\mathfrak{A}, c \sim_{\mathcal{L}, \Sigma} \mathfrak{B}, h'(c)$ for all $c \in D$. Take the composition $h' \circ h$ and observe that by Lemma 2, $h' \circ h(y) \in C^{\mathfrak{B}}$ if $C(y)$ is a conjunct of $\varphi$. Thus, $\mathfrak{B} \models \varphi(b)$, as required. The proof for $\mathrm{CQ}^{\mathcal{L}}$ is the same except that the one does not need to observe that the $\Sigma$-reduct of $\mathfrak{A}_{|D}$ is $\mathcal{L}$-rooted in $a$.

Conversely, assume that $\mathfrak{A}, a \Rightarrow_{\mathrm{CQ}^{\mathcal{L}}_r, \Sigma} \mathfrak{B}, b$. To show that $\mathfrak{A}, a \Rightarrow^{\mathrm{mod}}_{\mathrm{CQ}^{\mathcal{L}}_r, \Sigma} \mathfrak{B}, b$, let $D$ be such that the $\Sigma$-reduct of $\mathfrak{A}_{|D}$ is $\mathcal{L}$-rooted at $a$. Consider the set for formulas $q^{\mathfrak{A}}_D$ that is obtained by regarding the nodes $d$ in $D$ as variables $x_d$ and taking $(x_{d_1}, x_{d_2})$ if $(d_1, d_2) \in r^{\mathfrak{A}}$, $r \in \Sigma$, and $C(x_d)$ if $d \in C^{\mathfrak{A}}$ for $C \in \mathcal{L}(\Sigma)$. If follows from $\mathfrak{A}, a \Rightarrow_{\mathrm{CQ}^{\mathcal{L}}_r, \Sigma} \mathfrak{B}, b$ that every finite subset of $q^{\mathfrak{A}}_D$ is satisfied in $\mathfrak{B}$ under an assignment mapping $x_a$ to $b$. If $\mathfrak{B}$ is $\omega$-saturated, then $q^{\mathfrak{A}}_D$ is satisfied in $\mathfrak{B}$ by definition of $\omega$-saturatedness (and also holds if the $\Sigma$-reduct of $\mathfrak{A}_{|D}$ is not rooted in $a$). If $\mathfrak{B}$ has finite outdegree then this can be shown directly using the condition that the $\Sigma$-reduct of $\mathfrak{A}_{|D}$ is rooted in $a$. Let $v$ be the satisfying assignment. Then $h : D \to \mathfrak{B}$ defined by setting $h(d) = v(x_d)$ is a $\Sigma$-homomorphism, $h(a) = b$, and $\mathfrak{A}, c \sim_{\mathcal{L}, \Sigma} \mathfrak{B}, h(c)$ for all $c \in D$, as required. The implication for $\mathrm{CQ}^{\mathcal{L}}$ follows using the comment above. ❑

We slightly extend Lemma 1 as required later. The proof is by a standard selective unraveling procedure.

**Lemma 7** *Let $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}, \mathcal{ALCO}\}$ and let $\mathcal{K}$ be 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 $\mathfrak{A}$ of $\mathcal{K}$ of finite $\mathcal{L}$-outdegree with $a \notin C^{\mathfrak{A}}$.*

*For every model $\mathfrak{A}$ of $\mathcal{K}$ there exists an $\mathcal{L}$-forest model $\mathfrak{A}'$ of $\mathcal{K}$ such that $\mathfrak{A}', a^{\mathfrak{A}'} \sim^{f}_{\mathcal{L}} \mathfrak{A}, a^{\mathfrak{A}}$. If $\mathfrak{A}$ is finite, then there exists such a model of finite $\mathcal{L}$-outdegree.*

We now show that Lemma 1 does not hold for $\mathcal{ALCIO}$. Note that an $\mathcal{ALCIO}$-forest model of $\mathcal{K}$ is a model $\mathfrak{A}$ of $\mathcal{K}$ such that $\mathfrak{A}$ with all $R(a, b^{\mathfrak{A}})$, $R$ a role, $a \in \mathrm{dom}(\mathfrak{A})$, $b \in \mathrm{ind}(\mathcal{D})$ is a disjoint union of $\mathcal{ALCI}$-trees rooted at $a$, $a \in \mathrm{ind}(\mathcal{D})$. Then the concept

$$\{a\} \sqcap A \sqcap \exists s.\top \sqcap \forall s.(\neg A \sqcap \exists r.\exists s^{-}.\{a\})$$

is satisfiable in a model of $\mathcal{K} = (\emptyset, \{A(a)\})$, but not in any $\mathcal{ALCIO}$-forest model of $\mathcal{K}$ of finite $\mathcal{ALCIO}$-outdegree.

# C Proofs for Section 3

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

The proof is by a reduction of the following undecidable tiling problem.

**Definition 3** *A* tiling system $S = (\mathcal{T}, H, V, R, L, T, B)$ *consists of a finite set $\mathcal{T}$ of* tiles*, horizontal and vertical* matching relations $H, V \subseteq \mathcal{T} \times \mathcal{T}$*, and sets $R, L, T, B \subseteq \mathcal{T}$ of* right *tiles,* left *tiles,* top *tiles, and* bottom *tiles. A* solution *to $S$ is a triple $(n, m, \tau)$ where $n, m \geq 1$ and $\tau : \{0, \ldots, n\} \times \{0, \ldots, m\} \to \mathcal{T}$ such that the following hold:*

1. *$(\tau(i, j), \tau(i + 1, j)) \in H$, for all $i < n$ and $j \leq m$;*
2. *$(\tau(i, j), \tau(i, j + 1)) \in V$, for all $i \leq n$ and $j < m$;*
3. *$\tau(0, j) \in L$ and $\tau(n, j) \in R$, for all $0 \leq j \leq m$;*
4. *$\tau(i, 0) \in B$ and $\tau(i, m) \in T$, for all $0 \leq i \leq n$.*

We show how to convert a tiling system $S$ into a labeled $\mathcal{ALC}$-KB $(\mathcal{K}, P, N)$ and signature $\Sigma$ such that $S$ has a solution iff $(\mathcal{K}, P, N)$ is projectively $\mathcal{ALCO}(\Sigma)$-separable with individual names as additional helper symbols.

Let $S = (\mathcal{T}, H, V, R, L, T, B)$ be a tiling system. Define an ontology $\mathcal{O}$ containing the following inclusions.

- Every grid node is labeled with exactly one tile and the matching conditions are satisfied:

$$\top \sqsubseteq \bigsqcup_{t \in T} (t \sqcap \bigsqcap_{t' \in T,\, t' \neq t} \neg t')$$

$$\top \sqsubseteq \bigsqcap_{t \in T} (t \to (\bigsqcup_{(t,t') \in H} \forall r_x.t' \sqcap \bigsqcup_{(t,t') \in V} \forall r_y.t'))$$

- The concepts left, right, top, bottom mark the borders of the grid in the expected way:

$$\begin{aligned} \mathsf{bottom} &\sqsubseteq \neg\mathsf{top} \sqcap \forall r_x.\mathsf{bottom} \\ \mathsf{right} &\sqsubseteq \forall r_y.\mathsf{right} \\ \mathsf{left} &\sqsubseteq \neg\mathsf{right} \sqcap \forall r_y.\mathsf{left} \\ \mathsf{top} &\sqsubseteq \forall r_x.\mathsf{top} \\ \neg\mathsf{top} &\equiv \exists r_y.\top \\ \neg\mathsf{right} &\equiv \exists r_x.\top \end{aligned}$$

  and $\mathsf{bottom} \sqsubseteq \bigsqcup_{t\in B} t$, $\mathsf{right} \sqsubseteq \bigsqcup_{t\in R} t$, $\mathsf{left} \sqsubseteq \bigsqcup_{t\in L} t$, $\mathsf{top} \sqsubseteq \bigsqcup_{t\in T} t$.

- There is an infinite outgoing $r_x/r_y$-path starting at $Q$ or some grid cell does not close in the part of models reachable from $Q$:

$$Q \sqsubseteq \exists r_x.Q \sqcup \exists r_y.Q \sqcup (\exists r_x.\exists r_y.P \sqcap \exists r_y.\exists r_x.\neg P)$$

- $Q$ is triggered by $A_1 \sqcap D$:

$$A_1 \sqcap D \sqsubseteq Q$$

Now let $\mathcal{D} = \{A_1(a), Y(b), D(o), \mathsf{left}(o), \mathsf{bottom}(o)\}$, set

$$\Sigma = \{o, r_x, r_y, \mathsf{left}, \mathsf{right}, \mathsf{top}, \mathsf{bottom}\}$$

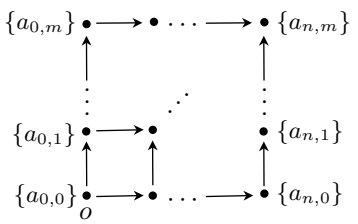
and consider the labeled KB $(\mathcal{K}, \{a\}, \{b\})$ where $\mathcal{K} = (\mathcal{O}, \mathcal{D})$.

**Lemma 8** *If $S$ has a solution, then there is an $\mathcal{ALCO}(\Sigma \cup \Sigma_{help})$-concept that separates $(\mathcal{K}, \{a\}, \{b\})$, where $\Sigma_{help}$ is a set of fresh individual names.*

**Proof.** Assume that $S$ has a solution consisting of a properly tiled $n \times m$ grid. We design an $\mathcal{ALCO}(\Sigma \cup \Sigma_{\text{help}})$-concept $G$ so that any model of $G$ and $\mathcal{K}$ includes a properly tiled $n \times m$-grid with lower left corner $o$, where $\Sigma_{\text{help}}$ is a set of fresh individual names. The individual names in $\Sigma_{\text{help}}$ are $a_{i,j}$, $0 \leq i \leq n$, $0 \leq j \leq m$. Then let $G$ be the $\mathcal{ALCO}(\Sigma \cup \Sigma_{\text{help}})$-concept that states that $G$ is true at the bottom left corner of a $r_x/r_y$ grid in which the nodes are given by the interpretation of the individual names $a_{i,j}$ and such that

- $a_{i+1,j}$ is the only $r_x$-successor of $a_{i,j}$;
- $a_{i,j+1}$ is the only $r_y$-successor of $a_{i,j}$;
- the borders of the grid satisfy the respective concepts left, right, top, bottom;
- $o = a_{0,0}$.

Thus, the grid can be depicted as follows:




We show that $\mathcal{K} \models \neg G(a)$ and $\mathcal{K} \not\models \neg G(b)$, thus $G$ separates $(\mathcal{K}, \{a, \{b\})$.

Assume first for a proof by contradiction that there is a model $\mathfrak{A}$ of $\mathcal{K}$ such that $\mathfrak{A} \models G(a)$. Then $a^{\mathfrak{A}} = o^{\mathfrak{A}}$ and so $a^{\mathfrak{A}} \in (A_1 \sqcap D)^{\mathfrak{A}}$. But then $a^{\mathfrak{A}} \in Q^{\mathfrak{A}}$. This contradicts the fact that $o^{\mathfrak{A}}$ is the origin of an $n \times m$-grid in $\mathfrak{A}$.

Now for $\mathcal{K} \not\models \neg G(b)$. We find a model $\mathfrak{A}$ of $\mathcal{K}$ with $b^{\mathfrak{A}} \in G^{\mathfrak{A}}$ since the concept name $Q$ is not triggered at $b$ as $A_1$ is not true for $b$. ❏

The following lemma implies that if $S$ has no solution, then $(\mathcal{K}, \{a\}, \{b\})$ is not projectively FO($\Sigma$)-separable.

**Lemma 9** *If $S$ has no solution, then for every model $\mathfrak{A}$ of $\mathcal{K}$, there is a model $\mathfrak{B}$ of $\mathcal{K}$ such that $(\mathfrak{A}, b^{\mathfrak{A}})$ is $\Sigma$-isomorphic to $(\mathfrak{B}, a^{\mathfrak{A}})$.*

**Proof.** (sketch) If $b^{\mathfrak{A}} \neq o^{\mathfrak{A}}$, then we can simply obtain $\mathfrak{B}$ from $\mathfrak{A}$ by switching $a^{\mathfrak{A}}$ and $b^{\mathfrak{A}}$, making $A_1$ true at $a^{\mathfrak{A}}$, and $D$ at exactly $o^{\mathfrak{A}}$. If $b^{\mathfrak{A}} = o^{\mathfrak{A}}$, then after switching we additionally have to re-interpret $Q$ and $P$ in a suitable way. But $S$ has no solution and thus when following $r_x/r_y$-paths from $o^{\mathfrak{A}}$ in $\mathfrak{A}$, we must either encounter an infinite such path or a non-closing grid cell as otherwise we can extract from $\mathfrak{A}$ a solution for $S$. Thus we can re-interpret $Q$ and $P$ as required. ❏

**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.*

**Proof.** First assume that $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}, \mathcal{ALCO}\}$. We employ the characterization of projective separability given below in Theorem 4. Observe that the following conditions are equivalent:

- there exists an $\mathcal{L}(\Sigma \cup (\mathsf{N_C} \cup \mathsf{N_R}) \setminus \text{sig}(\mathcal{K}))$-concept $C$ such that $\mathcal{K} \models C(a)$ for all $a \in P$ and $\mathcal{K} \not\models C(b)$;
- there exists an $\mathcal{L}$-forest model $\mathfrak{A}$ of $\mathcal{K}$ of finite $\mathcal{L}$-outdegree and a set $\Sigma'$ of concept and role names disjoint from sig($\mathcal{K}$) such that for all models $\mathfrak{B}$ of $\mathcal{K}$ and all $a \in P$: $\mathfrak{B}, a^{\mathfrak{B}} \not\sim_{\mathcal{L}, \Sigma \cup \Sigma'} \mathfrak{A}, b^{\mathfrak{A}}$.

Thus, for $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$ it suffices to show that the second condition is equivalent to the third condition of Theorem 4. For a proof by contradiction assume that there exists an $\mathcal{L}$-forest model $\mathfrak{A}$ of $\mathcal{K}$ satisfying Condition 2 above for $\Sigma'$ but not Condition 3 of Theorem 4. Take a model $\mathfrak{B}$ of $\mathcal{K}$ and a functional $\Sigma$-bisimulation $f$ witnessing $\mathfrak{B}, a^{\mathfrak{B}} \sim^f_{\mathcal{L},\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$ for some $a \in P$. We modify $\mathfrak{B}$ to obtain a model $\mathfrak{B}'$ of $\mathcal{K}$ such that $\mathfrak{B}', a^{\mathfrak{B}'} \sim^f_{\mathcal{L},\Sigma\cup\Sigma'} \mathfrak{A}, b^{\mathfrak{A}}$ and thus obtain a contradiction. $\mathfrak{B}'$ is obtained from $\mathfrak{B}$ by assuming first that $\mathfrak{B}$ does not interpret any symbol in $\Sigma'$ and then

- taking the disjoint union $\mathfrak{B} \cup \mathfrak{A}'$ of $\mathfrak{B}$ and a copy $\mathfrak{A}'$ of $\mathfrak{A}$ that does not interpret any individual names nor symbols in $\Sigma'$.
- observing that the function $g = f \cup \mathsf{id}$, where $\mathsf{id}$ maps every node in $\mathfrak{A}'$ to the node in $\mathfrak{A}$ of which it is a copy, is a functional and surjective $\mathcal{L}(\Sigma)$-bisimulation between $\mathfrak{B} \cup \mathfrak{A}'$ and $\mathfrak{A}$.
- setting $A^{\mathfrak{B}'} = g^{-1}(A^{\mathfrak{A}})$ for all concept names $A \in \Sigma'$ and $r^{\mathfrak{B}'} = g^{-1}(r^{\mathfrak{A}})$ for all role names $r \in \Sigma'$.

It is easy to see that $g$ is a functional $\mathcal{L}(\Sigma \cup \Sigma')$-bisimulation between $\mathfrak{B}'$ and $\mathfrak{A}$, as required.

Now assume that $\mathcal{L} = \mathcal{ALCO}$. As Condition 2 trivially implies Condition 1, it suffices to show that Condition 1 implies Condition 2. Assume that Condition 1 holds. Take an $\mathcal{L}$-forest model $\mathfrak{A}$ of $\mathcal{K}$ of finite $\mathcal{L}$-outdegree and a set $\Sigma'$ of concept and role names disjoint from $\mathsf{sig}(\mathcal{K})$ such that for all models $\mathfrak{B}$ of $\mathcal{K}$ and all $a \in P$: $\mathfrak{B}, a^{\mathfrak{B}} \not\sim_{\mathcal{L},\Sigma\cup\Sigma'} \mathfrak{A}, b^{\mathfrak{A}}$. Assume for a proof by contradiction that there does not exist any such model if $\Sigma'$ is replaced by $\Sigma \cup \{r_I\}$. Obtain $\mathfrak{A}'$ from $\mathfrak{A}$ dropping the interpretation of role names in $\Sigma'$ and instead setting

$$r_I^{\mathfrak{A}'} = \{(b^{\mathfrak{A}}, c^{\mathfrak{A}}) \mid c \in \mathsf{ind}(\mathcal{D})\} \cup \bigcup_{r \in \mathsf{N_R}} r^{\mathfrak{A}}.$$

Then $\mathfrak{A}'$ is an $\mathcal{L}$-forest model of $\mathcal{K}$ of finite $\mathcal{L}$-outdegree. Thus, by assumption there exists a model $\mathfrak{B}$ of $\mathcal{K}$ and $a \in P$ such that $\mathfrak{B}, a^{\mathfrak{B}} \sim^f_{\mathcal{L},\Sigma\cup\{r_I\}} \mathfrak{A}', b^{\mathfrak{A}'}$. Let $f$ be the functional $\mathcal{L}(\Sigma \cup \{r_I\})$-bisimulation witnessing this. Then $f$ is surjective. Now obtain $\mathfrak{B}'$ from $\mathfrak{B}$ by keeping the interpretation of symbols not in $\Sigma'$ and setting $A^{\mathfrak{B}'} = f^{-1}(A^{\mathfrak{A}})$ for all concept names $A \in \Sigma'$ and $r^{\mathfrak{B}'} = f^{-1}(r^{\mathfrak{A}})$ for all role names $r \in \Sigma'$. Then $f$ is a functional $\mathcal{L}(\Sigma \cup \Sigma')$-bisimulation between $\mathfrak{B}', a^{\mathfrak{B}'}$ and $\mathfrak{A}, b^{\mathfrak{A}}$ and we have derived a contradiction.

Finally, assume that $\mathcal{L} = \mathcal{ALCIO}$. Then we cannot use Theorem 4 as it does not hold for $\mathcal{ALCIO}$. However, if one replaces $\mathcal{L}$-forest models of finite $\mathcal{L}$-outdegree by $\omega$-saturated models, then Theorem 4 holds for $\mathcal{ALCIO}$. Now exactly the same proof can be done for $\mathcal{ALCIO}$ as for $\mathcal{ALCO}$ using $\omega$-saturated models instead of forest models. ❑

**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.*

The proof is by reduction of the same tiling problem as in the proof of Theorem 1. If fact, given a tiling system $S$, the labeled KB $(\mathcal{K}, \{a\}, \{b\})$ is exactly the same KB as in the proof of Theorem 1. The only difference is in Lemma 8 about the construction of a concept witnessing separability: this concept is now not a concept using individual names as helper symbols but a $\mathcal{ALCFIO}(\Sigma)$-concept without helper symbols.

**Lemma 10** *If $S$ has a solution, then there is an $\mathcal{ALCFIO}(\Sigma)$-concept that non-projectively separates $(\mathcal{K}, \{a\}, \{b\})$.*

**Proof.** Assume that $S$ has a solution consisting of a properly tiled $n \times m$ grid. We design an $\mathcal{ALCFIO}(\Sigma)$-concept $G$ so that any model of $G$ and $\mathcal{K}$ includes a properly tiled $n \times m$-grid with lower left corner $o$. Let $F$ be the obvious concept stating that $(\leqslant 1\ r)$ holds for $r \in \{r_x, r_x, r_y^-, r_x^-\}$ for all nodes reachable in no more than $2(n+m)$ steps along roles $r_x, r_x, r_y, r_x^-$. For every word $w \in \{r_x, r_y\}^*$, denote by $\overleftarrow{w}$ the word that is obtained by reversing $w$ and then adding $\cdot^-$ to each symbol. Let $|w|_r$ denote the number of occurrences of the symbol $r$ in $w$. Now let $G = F \sqcap E$, where $E$ is the conjunction of

$$\{o\} \sqcap \forall r_x^{n+1}.\bot \sqcap \forall r_x^{\leq n}.\mathsf{bottom} \sqcap \forall r_y^{m+1}.\bot \sqcap \forall r_y^{\leq m}.\mathsf{left}$$

and for every $w \in \{r_x, r_y\}^*$ such that $|w|_{r_x} < n$ and $|w|_{r_y} < m$, the concept

$$\exists(w \cdot r_x r_y r_x^- r_y^- \cdot \overleftarrow{w}).\{o\},$$

where $\exists w.F$ abbreviates $\exists r_1. \cdots \exists r_k.F$ if $w = r_1 \cdots r_k$. It is readily checked that $G$ indeed enforces a grid, as announced.

We show that $\mathcal{K} \models \neg G(a)$ and $\mathcal{K} \not\models \neg G(b)$, thus $G$ separates $(\mathcal{K}, \{a, \{b\})$.

Assume first for a proof by contradiction that there is a model $\mathfrak{A}$ of $\mathcal{K}$ such that $\mathfrak{A} \models G(a)$. Then $a^{\mathfrak{A}} = o^{\mathfrak{A}}$ and so $a^{\mathfrak{A}} \in (A_1 \sqcap D)^{\mathfrak{A}}$. But then $a^{\mathfrak{A}} \in Q^{\mathfrak{A}}$. This contradicts the fact that $o^{\mathfrak{A}}$ is the origin of an $n \times m$-grid in $\mathfrak{A}$.

Now for $\mathcal{K} \not\models \neg G(b)$. We find a model $\mathfrak{A}$ of $\mathcal{K}$ with $b^{\mathfrak{A}} \in G^{\mathfrak{A}}$ since the concept name $Q$ is not triggered at $b$ as $A_1$ is not true for $b$. ❑

# D Proofs for Section 4

**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 $\Sigma \subseteq \mathit{sig}(\mathcal{K})$. 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 $\Sigma_{\mathit{help}}$ of concept names disjoint from $\mathit{sig}(\mathcal{K})$ such that for all models $\mathfrak{B}$ of $\mathcal{K}$ and all $a \in P$: $\mathfrak{B}, a^{\mathfrak{B}} \not\sim_{\mathcal{L},\Sigma\cup\Sigma_{\mathit{help}}} \mathfrak{A}, b^{\mathfrak{A}}$.*
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 $a \in P$: $\mathfrak{B}, a^{\mathfrak{B}} \not\sim^f_{\mathcal{L},\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$.*

**Proof.** "1. $\Rightarrow$ 2". Assume that $(\mathcal{K}, P, \{b\})$ is projectively $\mathcal{L}(\Sigma)$-separable. Take an $\mathcal{L}$-concept $C$ that separates $(\mathcal{K}, P, \{b\})$ and uses symbols from $\Sigma \cup \Sigma_{\text{help}}$, where $\Sigma_{\text{help}}$ is a set of concept names disjoint from $\mathsf{sig}(\mathcal{K})$. By Lemma 1, there exists a $\mathcal{L}$-forest model $\mathfrak{A}$ of $\mathcal{K}$ of finite $\mathcal{L}$-outdegree such that $b^{\mathfrak{A}} \in (\neg C)^{\mathfrak{A}}$. Then $\mathfrak{A}$ is as required for Condition 2, by Lemma 2.

"2 $\Rightarrow$ 1". Assume Condition 2 holds for $\mathfrak{A}$ and $\Sigma_{\text{help}}$. Let

$$t_{\mathfrak{A}}(b) = \{C \in \mathcal{L}(\Sigma \cup \Sigma_{\text{help}}) \mid b^{\mathfrak{A}} \in C^{\mathfrak{A}}\}.$$

It follows from Lemma 2 that

$$\Gamma_a = \mathcal{K} \cup \{C(a) \mid C \in t_{\mathfrak{A}}(b)\}$$

is not satisfiable, for any $a \in P$. (Otherwise an $\omega$-saturated satisfying model would contradict Condition 2.) By compactness (and closure under conjunctions) we find for every $a \in P$ a concept $C_a \in t_{\mathfrak{A}}(b)$ such that $\mathcal{K} \models \neg C(b)$. Thus, the concept $\neg(\bigsqcap_{a \in P} C_a)$ separates $(\mathcal{K}, P, \{b\})$, as required.

"2 $\Rightarrow$ 3". Take an $\mathcal{L}$-forest model $\mathfrak{A}$ and $\Sigma_{\text{help}}$ such that Condition 2 holds. We show that Condition 3 holds for $\mathfrak{A}$ as well. Suppose for a proof by contradiction that there exists a model $\mathfrak{B}$ of $\mathcal{K}$ and an $a \in P$ and a functional $\Sigma$-bisimulation $f$ witnessing $\mathfrak{B}, a^{\mathfrak{B}} \sim^f_{\mathcal{L},\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$. We may assume that $\mathfrak{B}$ does not interpret any symbols in $\Sigma_{\text{help}}$ and define $\mathfrak{B}'$ by expanding $\mathfrak{B}$ as follows: for every concept name $A \in \Sigma_{\text{help}}$ and $d \in \text{dom}(f)$, let $d \in A^{\mathfrak{B}'}$ if $f(d) \in A^{\mathfrak{A}}$. It is easy to see that $f$ witnesses $\mathfrak{B}', a^{\mathfrak{B}'} \sim_{\mathcal{L},\Sigma \cup \Sigma_{\text{help}}} \mathfrak{A}, b^{\mathfrak{A}}$, and we have derived a contradiction.

"3 $\Rightarrow$ 2". Take an $\mathcal{L}$-forest model $\mathfrak{A}$ of $\mathcal{K}$ of finite $\mathcal{L}$-outdegree such that Condition 3 holds. We may assume that $\mathfrak{A}$ only interprets the symbols in $\text{sig}(\mathcal{K})$. Define $\mathfrak{A}'$ by expanding $\mathfrak{A}$ as follows. Take for any $d \in \text{dom}(\mathfrak{A})$ a fresh concept name $A_d$ and set $A_d^{\mathfrak{A}'} = \{d\}$. Then Condition 2 holds for $\mathfrak{A}'$ and $\Sigma_{\text{help}} = \{A_d \mid d \in \text{dom}(\mathfrak{A})\}$. ❑

We next show that Theorem 4 does not hold for $\mathcal{ALCIO}$. To this end we define a labeled $\mathcal{ALCI}$-KB $(\mathcal{K}, \{a\}, \{b\})$ and signature $\Sigma$ such that $(\mathcal{K}, \{a\}, \{b\})$ is weakly $\mathcal{ALCIO}(\Sigma)$-separable but there does not exist an $\mathcal{ALCIO}$-forest model $\mathfrak{A}$ of $\mathcal{K}$ of finite $\mathcal{ALCIO}$-outdegree such that for all models $\mathfrak{B}$ of $\mathcal{K}$: $\mathfrak{A}, b^{\mathfrak{A}} \not\sim^f_{\mathcal{ALCIO},\Sigma} \mathfrak{B}, a^{\mathfrak{B}}$.

Let $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ with

$$\begin{aligned}
\mathcal{D} &= \{A(a), B(b), C(c), r_0(b,c)\} \\
\mathcal{O} &= \{C \sqsubseteq \exists r_0^-.A \to A_0, \\
&\qquad C \sqsubseteq \exists s.\top \sqcap \forall s.(E \sqcap \exists r.(E \sqcap \exists s^-.\top)) \\
&\qquad A_0 \sqsubseteq \exists s.\exists s^-.\neg A_0 \sqcup \exists s.\exists r.\exists s^-.\neg A_0, \\
&\qquad B \sqcup A \sqsubseteq \neg C\}
\end{aligned}$$

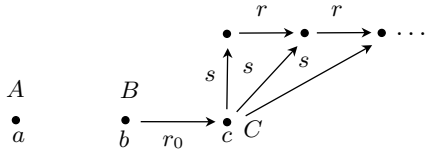
where $E$ stands for $\neg C \sqcap \neg A \sqcap \neg B$. Let $\Sigma = \{c, r_0, s, r\}$.

**Lemma 11** *The $\mathcal{ALCIO}(\Sigma)$-concept*

$$D = \neg\exists r_0(\{c\} \sqcap \forall s.(\forall s^-.\{c\} \sqcap \forall r.\forall s^-.\{c\}))$$

*weakly separates $(\mathcal{K}, \{a\}, \{b\})$.*

**Proof.** We first show that $\mathcal{K} \models D(a)$. Assume there is a model $\mathfrak{A}$ of $\mathcal{K}$ with $a^{\mathfrak{A}} \notin D^{\mathfrak{A}}$. Then $(a^{\mathfrak{A}}, c^{\mathfrak{A}}) \in r_0^{\mathfrak{A}}$. Then by definition of $\mathcal{K}$ we have $c^{\mathfrak{A}} \in A_0^{\mathfrak{A}}$ thus $c^{\mathfrak{A}} \in (\exists s.\exists s^-.\neg A_0 \sqcup \exists s.\exists r.\exists s^-.\neg A_0)^{\mathfrak{A}}$, contradicting $c^{\mathfrak{A}} \in (\forall s.(\forall s^-.\{c\} \sqcap \forall r.\forall s^-.\{c\}))^{\mathfrak{A}}$. On the other hand, the model depicted below




clearly satisfies $D(b)$. The fact that it is a model of $\mathcal{K}$ is also straightforward, as its extension of $A_0$ is empty.

❑

**Lemma 12** *For every $\mathcal{ALCIO}$-forest model $\mathfrak{A}$ of $\mathcal{K}$ of finite $\mathcal{ALCIO}$-outdegree there exists a model $\mathfrak{B}$ of $\mathcal{K}$ such that $\mathfrak{B}, a^{\mathfrak{B}} \sim^f_{\mathcal{ALCIO},\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$.*

**Proof.** Assume $\mathfrak{A}$ is given. We construct $\mathfrak{B}$ as follows. Let $\text{dom}(\mathfrak{B}) = \text{dom}(\mathfrak{A})$, $a^{\mathfrak{B}} = b^{\mathfrak{B}} = b^{\mathfrak{A}}$, $c^{\mathfrak{B}} = c^{\mathfrak{A}}$, $A^{\mathfrak{B}} = \{a^{\mathfrak{B}}\}$, $A_0^{\mathfrak{B}} = C^{\mathfrak{B}} = \{c^{\mathfrak{B}}\}$, $C^{\mathfrak{B}} = \{c^{\mathfrak{A}}\}$, and $B^{\mathfrak{B}} = B^{\mathfrak{A}}$. Let $\rho^{\mathfrak{B}} = \rho^{\mathfrak{A}}$ for all role names $\rho$. There is a $\Sigma$-isomorphism between $\mathfrak{A}, b^{\mathfrak{A}}$ and $\mathfrak{B}, a^{\mathfrak{B}}$, as $\mathfrak{B}$ only differs from $\mathfrak{A}$ with respect to symbols outside of $\Sigma$. It is clear that $\mathfrak{B}$ is a model of $\mathcal{D}$. We then check that $\mathfrak{B}$ satisfies each inclusion of $\mathcal{O}$. The first inclusion is clearly satisfied as $C^{\mathfrak{B}} = A_0^{\mathfrak{B}} = \{c^{\mathfrak{B}}\}$. The second and fourth inclusion are clearly satisfied by $\mathfrak{B}$ as they are by $\mathfrak{A}$. The third inclusion is satisfied: we have $A_0^{\mathfrak{B}} = \{c^{\mathfrak{B}}\}$. Assume the third inclusion is not satisfied. Then, by the second inclusion and $C(c) \in \mathcal{D}$, there is an infinite $r^{\mathfrak{A}}$-chain of nodes distinct from $c^{\mathfrak{A}}, a^{\mathfrak{A}}, b^{\mathfrak{A}}$ all of which are in relation $(s^-)^{\mathfrak{A}}$ to $c^{\mathfrak{A}}$. Then either $\mathfrak{A}$ is not an $\mathcal{ALCIO}$-forest model as it contains an $r^{\mathfrak{A}}$-cycle of nodes distinct from interpretations of individual names or it does not have finite $\mathcal{ALCIO}$-outdegree as the outdegree of $c^{\mathfrak{A}}$ is infinite. ❑

We now show that Theorem 4 cannot be repaired for $\mathcal{ALCIO}$ by admitting infinite outdegree forest models. To this end we define a labeled $\mathcal{ALCI}$-KB $(\mathcal{K}, \{a\}, \{b\})$ and signature $\Sigma$ such that $(\mathcal{K}, \{a\}, \{b\})$ is not projectively weakly $\mathcal{ALCIO}(\Sigma)$-separable but there exists an $\mathcal{ALCIO}$-forest model $\mathfrak{A}$ of $\mathcal{K}$ of such that for all models $\mathfrak{B}$ of $\mathcal{K}$: $\mathfrak{B}, a^{\mathfrak{B}} \not\sim^f_{\mathcal{ALCIO},\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$.

Let $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ with

$$\begin{aligned}
\mathcal{D} &= \{A(a), B(b), C(c), r_0(b,c)\} \\
\mathcal{O} &= \{C \sqsubseteq \exists r_0^-.A \to A_0, \\
&\qquad A_0 \sqsubseteq (\exists s.\top \sqcap \forall s.\exists r.\exists s^-.A_0) \to \exists s.B', \\
&\qquad B' \sqsubseteq \exists r^-.B' \\
&\qquad B \sqcup A \sqsubseteq \neg C\}
\end{aligned}$$

Let $\Sigma = \{r_0, s, r, c\}$.

**Lemma 13** *$(\mathcal{K}, \{a\}, \{b\})$ is not weakly projectively $\mathcal{ALCIO}(\Sigma)$-separable.*

**Proof.** Let $\mathfrak{A}$ be a model of $\mathcal{K}$. We show there exists a model $\mathfrak{B}$ of $\mathcal{K}$ such that $\mathfrak{B}, a^{\mathfrak{B}} \equiv_{\mathcal{ALCIO},\Sigma \cup \Sigma'} \mathfrak{A}, b^{\mathfrak{A}}$, where $\Sigma'$ is the set of all concept names that do not occur in $\mathcal{K}$. Let the model $\mathfrak{B}_0$ be defined in the same way as $\mathfrak{A}$ except that $a^{\mathfrak{B}_0} = b^{\mathfrak{A}}$, $A^{\mathfrak{B}_0} = \{a^{\mathfrak{B}_0}\}$ $A_0^{\mathfrak{B}_0} = C^{\mathfrak{B}_0} = \{c^{\mathfrak{B}_0}\}$, and $B^{\mathfrak{B}_0} = \{b^{\mathfrak{B}_0}\}$, and we define the extension of $B'$ according to a case distinction.

Case 1. $c^{\mathfrak{B}_0} \notin (\exists s.\top \sqcap \forall s.\exists r.\exists s^-.A_0)^{\mathfrak{B}_0}$. Then set $B'^{\mathfrak{B}_0} = \emptyset$. Then $\mathfrak{B}_0$ is a model of $\mathcal{K}$ and the identity is a $\Sigma$-isomorphism between $\mathfrak{B}_0$ and $\mathfrak{A}$ mapping $a^{\mathfrak{B}_0}$ to $b^{\mathfrak{A}}$ and we are done.

Case 2. Otherwise. As $A_0^{\mathfrak{B}_0} = \{c^{\mathfrak{B}_0}\}$, the set

$$t(x) = \{s(c,x)\} \cup \{r(y_1,x), r(y_2,y_1), r(y_3,y_2), \ldots\}$$

is finitely satisfiable in $\mathfrak{B}_0$, so it is realized in an elementary extension $\mathfrak{B}_1$ of $\mathfrak{B}_0$. That implies there exists an infinite $r^-$-chain $a_1', a_2', \ldots$ in $\mathfrak{B}_1$ with $(c^{\mathfrak{B}_1}, a_1') \in s^{\mathfrak{B}_1}$. Let $\mathfrak{B}$ be

obtained from $\mathfrak{B}_1$ by defining the extension of $B'$ as $\{a'_i : i \geq 1\}$. Then $\mathfrak{B}$ is a model of $\mathcal{K}$ and $\mathfrak{A}, b^{\mathfrak{A}} \equiv_{\mathcal{ALCI}\mathcal{O},\Sigma\cup\Sigma'} \mathfrak{B}, a^{\mathfrak{B}}$. ❑

**Lemma 14** *There exists a model $\mathfrak{A}$ of $\mathcal{K}$ such that $\mathfrak{B}, a^{\mathfrak{B}} \nsim_{\mathcal{ALCI}\mathcal{O},\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$ for all models $\mathfrak{B}$ of $\mathcal{K}$.*

**Proof.** Consider the model $\mathfrak{A}$ depicted above. An explicit definition is given by setting

$$\mathrm{dom}(\mathfrak{A}) = \{a, b, c\} \cup \{a_i : i \geq 0\}$$

and

$$\begin{array}{ll} A_0^{\mathfrak{A}} = \emptyset & (B')^{\mathfrak{A}} = \emptyset \\ A^{\mathfrak{A}} = \{a\} = a^{\mathfrak{A}} & r_0^{\mathfrak{A}} = \{b, c\} \\ B^{\mathfrak{A}} = \{b\} = b^{\mathfrak{A}} & r^{\mathfrak{A}} = \{(a_i, a_{i+1}) : i \geq 0\} \\ C^{\mathfrak{A}} = \{c\} = c^{\mathfrak{A}} & s^{\mathfrak{A}} = \{(c, a_i) : i \geq 0\} \end{array}$$

It is immediate that $\mathfrak{A}$ is a model of $\mathcal{K}$. If $\mathfrak{B}, a^{\mathfrak{B}} \sim_{\mathcal{ALCI}\mathcal{O},\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$, then $b^{\mathfrak{A}} \in (\exists r_0.(\{c\} \sqcap \exists s.\top \sqcap \forall s.\exists r.\exists s^-.\{c\}))^{\mathfrak{A}}$ implies $a^{\mathfrak{B}} \in (\exists r_0.(\{c\} \sqcap \exists s.\top \sqcap \forall s.\exists r.\exists s^-.\{c\}))^{\mathfrak{B}}$ as $c, s, r, r_0 \in \Sigma$. The latter implies $a^{\mathfrak{B}} \in (\exists r_0.(A_0 \sqcap \exists s.\top \sqcap \forall s.\exists r.\exists s^-.A_0))^{\mathfrak{B}}$ as $c^{\mathfrak{B}} \in A_0^{\mathfrak{B}}$ in virtue of $\{A(a), C(c), r_0(a, c)\} \subseteq \mathcal{D}$ and the first inclusion $C \sqsubseteq \exists r_0^-.A \to A_0$ of $\mathcal{O}$. Then, by the second inclusion we get that $a^{\mathfrak{B}} \in (\exists r_0.\exists s.B')^{\mathfrak{B}}$ while $b^{\mathfrak{A}} \notin (\exists r_0.\exists s.B')^{\mathfrak{A}}$. By the third inclusion, $a^{\mathfrak{B}}$ then has a $r_0$ successor with an $s$ successor from which starts an infinite $r^-$ chain, while $b^{\mathfrak{A}}$ does not. A $\Sigma$-bisimulation including $(a^{\mathfrak{B}}, b^{\mathfrak{A}})$ is then impossible, as $\{r_0, s, r\} \subseteq \Sigma$. ❑

The following lemma provides a model-theoretic characterization of $(\mathcal{L}, \mathrm{UCQ}_r^{\mathcal{L}_S})$-separability, for some pairs $(\mathcal{L}, \mathcal{L}_S)$.

**Lemma 15** *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 non-projectively $UCQ_r^{\mathcal{L}_S}(\Sigma)$-separable;*
2. *there exists an $\mathcal{L}_S$-forest model $\mathfrak{A}$ of $\mathcal{K}$ of finite $\mathcal{L}_S$-outdegree such that there exists an $n$ such that for all models $\mathfrak{B}$ of $a \in P$: there exist $D \subseteq dom(\mathfrak{B})$ of cardinality not exceeding $n$ such that the $\Sigma$-reduct of $\mathfrak{A}_{|D}$ is $\mathcal{L}_S$-rooted in $a^{\mathfrak{B}}$ and $\mathfrak{B}, a^{\mathfrak{B}} \not\to_{D,\mathcal{ALCO},\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$.*

**Proof.** "1 $\Rightarrow$ 2". Assume that Condition 1 holds and take a formula $\varphi(x)$ in $\mathrm{UCQ}_r^{\mathcal{L}_S}(\Sigma)$ such that $\mathcal{K} \models \varphi(a)$ for all $a \in P$ and $\mathcal{K} \not\models \varphi(b)$. Then there exists a model $\mathfrak{A}$ of $\mathcal{K}$ such that $\mathfrak{A} \not\models \varphi(b)$. By Lemma 7, we may assume that $\mathfrak{A}$ is an $\mathcal{L}_S$-forest model of $\mathcal{K}$ of finite $\mathcal{L}_S$-outdegree. Then Condition 2 follows for $n$ the number of variables in $\varphi$, using Lemma 3.

"2 $\Rightarrow$ 1". The proof is indirect. Assume that Condition 1 does not hold. Let $\mathfrak{A}$ be any $\mathcal{L}_S$-forest model of $\mathcal{K}$ of finite $\mathcal{L}_S$-outdegree and set

$$\Gamma = \mathcal{K} \cup \{\neg\varphi(x) \mid \varphi(x) \in \mathrm{UCQ}_r^{\mathcal{L}_S}(\Sigma), \mathfrak{A} \models \neg\varphi(b)\}.$$

Then, by compactness, $\Gamma$ is satisfiable with $x = a$ for some $a \in P$. To show this, assume that it is not the case. Then for any $a \in P$ there exists a finite subset $\Gamma'_a$ of $\Gamma$ such that $\Gamma'_a$ is not satisfiable with $x = a$. Then $\Gamma' = \bigcup_{a\in P} \Gamma'_a$ is not satisfiable with $x = a$, for any $a \in P$. We may assume that $\Gamma' = \mathcal{K} \cup \{\neg\varphi_1(x), \cdots, \neg\varphi_n(x)\}$. Then $\mathcal{K} \models \varphi_1 \vee \cdots \vee \varphi_n(a)$ for all $a \in P$. Observe that $\varphi_1 \vee \cdots \vee \varphi_n \in \mathrm{UCQ}_r^{\mathcal{L}_S}$. Thus, as we assume that Condition 1 does not hold, $\mathcal{K} \models \varphi_1 \vee \cdots \vee \varphi_n(b)$. Hence $\mathfrak{A} \models \varphi_1 \vee \cdots \vee \varphi_n(b)$ and so there exists $i$ such that $\mathfrak{A} \models \varphi_i(b)$. We have derived a contradiction.

Take an $\omega$-saturated model $\mathfrak{B}$ of $\mathcal{K}$ satisfying $\Gamma$ in some $a \in P$. We have by definition $\mathfrak{B}, a^{\mathfrak{B}} \Rightarrow_{\mathrm{CQ}_r^{\mathcal{L}_S},\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$. By Lemma 3, $\mathfrak{B}, a^{\mathfrak{B}} \Rightarrow^{\mathrm{mod}}_{\mathrm{CQ}_r^{\mathcal{L}_S},\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$. This contradicts Condition 2, as required. ❑

**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.** We first assume that $(\mathcal{L}, \mathcal{L}_S) = (\mathcal{ALCI}, \mathcal{ALCI})$. It suffices to show that Point 3 of Theorem 4 and Point 2 of Lemma 15 are equivalent. Assume first that Point 3 of Theorem 4 holds for $\mathfrak{A}$. For a model $\mathfrak{C}$ of $\mathcal{K}$ we denote by $\mathcal{D}_\Sigma^{\mathfrak{C},a}$ the maximal connected component of $a^{\mathfrak{C}}$ in $\mathfrak{C}_{|\mathcal{D}^{\mathfrak{C}}}$, where $\mathcal{D}^{\mathfrak{C}} = \{c^{\mathfrak{C}} \mid c \in \mathrm{ind}(\mathcal{D})\}$.

We show that for all models $\mathfrak{B}$ of $\mathcal{K}$ and all $a \in P$: $\mathfrak{B}, a^{\mathfrak{B}} \not\to_{\mathcal{D}_\Sigma^{\mathfrak{B},a},\mathcal{ALCI},\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$. Then Point 2 of Lemma 15 holds for $n = |\mathcal{D}|$. For a proof be contradiction assume that there is a model $\mathfrak{B}$ of $\mathcal{K}$ and an $a \in P$ such that $h : \mathfrak{B}, a^{\mathfrak{B}} \to_{\mathcal{D}_\Sigma^{\mathfrak{B},a},\mathcal{ALCI},\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$. We aim to convert $\mathfrak{B}$ and $h$ into a new model $\mathfrak{B}'$ of $\mathcal{K}$ and a functional bisimulation witnessing $\mathfrak{B}', a^{\mathfrak{B}'} \sim^f_{\mathcal{L},\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$ and thus derive a contradiction. To this end, we require the notion of a $k$-unfolding of a structure in which we do not only unfold into a tree-like structure but also take $k$ copies of every successor. In detail, we define the *$k$-unfolding* $\mathfrak{B}_d^{\leq k}$ of a structure $\mathfrak{B}$ at $d \in \mathrm{dom}(\mathfrak{B})$ as follows, for any $k > 0$. The domain of $\mathfrak{B}_d^{\leq k}$ is the set $W$ of all words $w = d_0 R_0(d_1, i_1) \cdots R_{n-1}(d_n, i_n)$ such that $d_0 = d$, $(d_i, d_{i+1}) \in R_i^{\mathfrak{B}}$ for all $i < n$, and $i_j \leq k$ for all $j \leq n$, where all $R_i$ are roles. Let $\mathrm{tail}(w) = d_n$. The interpretation of concept names and role names is as expected: we set $w \in A^{\mathfrak{B}_d^{\leq k}}$ if $\mathrm{tail}(w) \in A^{\mathfrak{B}}$ and we set for $w_1, w_2 \in W$, $(w_1, w_2) \in R^{\mathfrak{B}_d^{\leq k}}$ if $w_2$ is obtained from $w_1$ by concatenating $w_1$ and some $R_{n+1}(d_{n+1}, i_{n+1})$.

Now let $k$ be the maximum over the $\mathcal{ALCI}$-outdegrees of the nodes in $\mathfrak{A}$. We define a new model $\mathfrak{B}'$ of $\mathcal{K}$ by taking $\mathfrak{B}$, removing all nodes $d$ not in $\mathcal{D}^{\mathfrak{B}}$ from it, and instead attaching $\mathfrak{B}_d^{\leq k}$ to $d$, for any $d \in \mathcal{D}^{\mathfrak{B}}$. Now one can easily show that there is a functional $\mathcal{ALCI}(\Sigma)$-bisimulation $f$ between $\mathfrak{B}', a^{\mathfrak{B}'}$ and $\mathfrak{A}, b^{\mathfrak{A}}$: to define $f$ take the homomorphism $h$ and extend it with functional bisimulations witnessing $\mathfrak{B}_d^{\leq k}, d \sim^f_{\mathcal{ALCI},\Sigma} \mathfrak{A}, h(d)$ for every $d \in \mathcal{D}_\Sigma^{\mathfrak{B},a}$.

Assume now that Point 2 of Lemma 15 holds for $\mathfrak{A}$. We show that Point 3 of Theorem 4 holds for $\mathfrak{A}$. The

proof is indirect. Assume Point 3 does not hold for $\mathfrak{A}$. Thus, there exists a model $\mathfrak{B}$ of $\mathcal{K}$ and $a \in P$ such that $\mathfrak{B}, a^{\mathfrak{B}} \sim^{f}_{\mathcal{ALCI},\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$. Then we can regard the restriction of $f$ to any subset $D$ of $\mathrm{dom}(\mathfrak{B})$ as a $\Sigma$-homomorphism $h$ for which clearly $\mathfrak{B}, c \sim_{\mathcal{ALCI},\Sigma} \mathfrak{A}, h(c)$ for all $c \in D$. Thus, $\mathfrak{B}, a^{\mathfrak{B}} \to_{D,\mathcal{ALCI},\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$. But then Point 2 of Lemma 15 does not hold for $\mathfrak{A}$.

We now assume that $(\mathcal{L}, \mathcal{L}_S) = (\mathcal{ALC}, \mathcal{ALCO})$. Again it suffices to show that Point 3 of Theorem 4 and Point 2 of Lemma 15 are equivalent.

Assume first that Point 3 of Theorem 4 holds for $\mathfrak{A}$. We show that $\mathfrak{A}$ witnesses Point 2 of Lemma 15. The proof is indirect. Assume that for all $n > 0$ there exists a model $\mathfrak{B}$ of $\mathcal{K}$ and $a \in P$ such that for all $D \subseteq \mathrm{dom}(\mathfrak{B})$ of cardinality not exceeding $n$ such that the $\Sigma$-reduct of $\mathfrak{A}_{|D}$ is $\mathcal{ALCO}$-rooted in $a^{\mathfrak{B}}$ we have $\mathfrak{B}, a^{\mathfrak{B}} \to_{D,\mathcal{ALCO},\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$.

Let $R$ denote the set of individuals $c \in \mathrm{ind}(\mathcal{D}) \cap \Sigma$ such that there is an $\mathcal{ALC}(\Sigma)$-path from $b^{\mathfrak{A}}$ to $c^{\mathfrak{A}}$ in $\mathfrak{A}$. For any $c \in R$ let $n_c$ be the length of the shortest such path and let $m = \sum_{c \in R} n_c |\mathrm{ind}(\mathcal{D})|$. Let $\mathfrak{B}$ be a model of $\mathcal{K}$ and $a \in P$ such that for all $D \subseteq \mathrm{dom}(\mathfrak{B})$ of cardinality not exceeding $m$ and such that the $\Sigma$-reduct of $\mathfrak{B}_{|D}$ is $\mathcal{ALCO}$-rooted in $a^{\mathfrak{B}}$ we have $\mathfrak{B}, a^{\mathfrak{B}} \to_{D,\mathcal{ALCO},\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$.

Now choose $D_0 \subseteq \mathrm{dom}(\mathfrak{B})$ minimal such that the $\Sigma$-reduct of $\mathfrak{B}_{|D}$ is rooted in $a^{\mathfrak{B}}$ and $D_0$ contains all $c^{\mathfrak{B}}$ with $c \in \mathrm{ind}(\mathcal{D}) \cap \Sigma$ such that there is an $\mathcal{ALC}(\Sigma)$-path from $a^{\mathfrak{B}}$ to $c^{\mathfrak{B}}$. Note that as $\mathfrak{B}, a^{\mathfrak{B}} \sim_{\mathcal{ALCO},\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$ the individuals we obtain are exactly those in $R$ and the cardinality of $D_0$ does not exceed $\sum_{c \in R} n_c$. Obtain $D$ from $D_0$ by adding all nodes $c^{\mathfrak{B}}$ with $c \in \mathrm{ind}(\mathcal{D})$ such that there exists an $\mathcal{ALC}(\Sigma)$-path from a node in $D_0$ through $\mathcal{D}^{\mathfrak{B}}$ to $c^{\mathfrak{B}}$. Then the cardinality of $D$ does not exceed $m$. Thus, we have $\mathfrak{B}, a^{\mathfrak{B}} \to_{D,\mathcal{ALCO},\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$. Let $h$ be the $\Sigma$-homomorphism witnessing this.

We define the *directed $k$-unfolding omitting $\Sigma$-individuals*, $\mathfrak{B}_d^{d,\leq k}$, of a structure $\mathfrak{B}$ at $d \in \mathrm{dom}(\mathfrak{B})$ as follows, for any $k > 0$. The domain of $\mathfrak{B}_d$ is the set $W$ of all words $w = d_0 r_0(d_1, i_1) \cdots r_{n-1}(d_n, i_n)$ such that $d_0 = d$, $(d_i, d_{i+1}) \in r_i^{\mathfrak{B}}$ for all $i < n$, and $i_j \leq k$ for all $j \leq n$, where all $r_i$ are role names and $d_n$ does not interpret an individual name in $\Sigma$ if $r_0, \ldots, r_{n-1}$ are all in $\Sigma$. Let $\mathrm{tail}(w) = d_n$. The interpretation of concept names and role names is as expected: we set $w \in A^{\mathfrak{B}_d^{d,\leq k}}$ if $\mathrm{tail}(w) \in A^{\mathfrak{B}}$ and we set for $w_1, w_2 \in W$, $(w_1, w_2) \in r^{\mathfrak{B}_d^{d,\leq k}}$ if $w_2$ is obtained from $w_1$ by concatenating $w_1$ and some $r_{n+1}(d_{n+1}, i_{n+1})$.

Now let $k$ be the maximal $\mathcal{ALC}$-outdegree of a node in $\mathfrak{A}$ and define a model $\mathfrak{B}'$ of $\mathcal{K}$ by taking $\mathfrak{B}$, removing all nodes $d$ not in $\mathcal{D}^{\mathfrak{B}}$ from it, and instead attaching $\mathfrak{B}_d^{d,\leq k}$ to $d$ for any $d \in \mathcal{D}^{\mathfrak{B}}$. Moreover, add $(w, c^{\mathfrak{B}})$ to the interpretation of $r$ if $(\mathrm{tail}(w), c^{\mathfrak{B}}) \in r^{\mathfrak{B}}$ and $c \in \Sigma$.

Then one can show that there is a functional $\mathcal{ALCO}(\Sigma)$-bisimulation $f$ between $\mathfrak{B}', a^{\mathfrak{B}}$ and $\mathfrak{A}, b^{\mathfrak{A}}$ by taking the homomorphim $h$ and extend it with the functional bisimulations witnessing $\mathfrak{B}_d^{d,\leq k}, d \sim^{f}_{\mathcal{ALCO}} \mathfrak{A}, h(d)$ for every $d \in D$.

The implication from Point 2 of Lemma 15 to Point 3 of Theorem 4 can be proved in the same way as for $\mathcal{ALCI}$. ❑

# E Proofs for Section 5

**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.** Assume $\mathcal{L}$-ontologies $\mathcal{O}$ and $\mathcal{O}'$ are given. Let $\Sigma$ be the signature of $\mathcal{O}$. Let $\mathrm{atom}_\Sigma$ denote the set of concepts $A$ with $A \in \Sigma \cap \mathsf{N_C}$, $\{a\}$ with $a \in \Sigma \cap \mathsf{N_I}$, and $\exists r.\top$ with $r \in \Sigma$. If $\mathcal{L}$ admit inverse roles, then we also add $\exists r^-.\top$, for $r \in \Sigma$. We may assume that there exists a concept name $A \in \mathrm{atom}_\Sigma$ such that $\mathcal{O} \models A \equiv \neg C$ for some $C \in \mathrm{atom}_\Sigma$. Indeed, if no such $A$ exists, pick any $X \in \mathrm{atom}_\Sigma$, add $A \sqsubseteq \neg X$, $X \sqsubseteq \neg A$ to $\mathcal{O}$ to obtain $\mathcal{O}_1$, and add $A$ to $\Sigma$. Then clearly $\mathcal{O}_1 \cup \mathcal{O}'$ is a conservative extension of $\mathcal{O}_1$ in $\mathcal{L}$ (projectively or, respectively, non-projectively) iff $\mathcal{O} \cup \mathcal{O}'$ is a conservative extension of $\mathcal{O}$ in $\mathcal{L}$ (projectively or, respectively, non-projectively).

We first consider the case $\mathcal{L} = \mathcal{ALCO}$ which subsumes the case $\mathcal{L} = \mathcal{ALC}$. The *relativization* $C_{|A}$ of a concept $C$ to a concept name $A$ is defined by setting

$$\begin{aligned}
\top_{|A} &= A \\
\bot_{|A} &= \bot \\
B_{|A} &= B \sqcap A \\
\{c\}_{|A} &= \{c\} \sqcap A \\
(\neg C)_{|A} &= A \sqcap \neg(C_{|A}) \\
C \sqcap D)_{|A} &= C_{|A} \sqcap D_{|A} \\
(\exists R.C)_{|A} &= A \sqcap \exists R.(A \sqcap C_{|A})
\end{aligned}$$

The relativization of an inclusion $C \sqsubseteq D$ to $A$ is defined as $C_{|A} \sqsubseteq D_{|A}$. Observe that the relativization of an inclusion to a concept name $A$ is satisfied in $\mathfrak{A}$ whenever $A^{\mathfrak{A}} = \emptyset$. Define the *directed relativization* $\mathcal{O}^A$ of $\mathcal{O}$ to a fresh concept name $A$ by relativizing all inclusions in $\mathcal{O}$ to $A$ and also adding

$$\{c\} \sqsubseteq A,$$

for all $c \in \Sigma$, and

$$A \sqsubseteq \forall r.A$$

for all $r \in \Sigma$. Next define a database $\mathcal{D}$ by taking fresh individual names $a$ and $b$ used as the positive and negative example, a fresh concept name $D$, and a fresh role name $s$ and include in $\mathcal{D}$: $A(b)$, $D(a)$, and

$$s(a, c),$$

for all individual names $c$ in $\mathcal{O} \cup \mathcal{O}'$. Next we take the directed relativization $(\mathcal{O} \cup \mathcal{O}')^{D'}$ to a fresh concept name $D'$, but in this case instead of including the individual names in $\mathcal{O} \cup \mathcal{O}'$ in $D'$ we include $D' \sqsubseteq \forall s.D'$.

Finally, we obtain $\mathcal{O}^*$ as the union of $\mathcal{O}^A$ and $(\mathcal{O} \cup \mathcal{O}')^{D'}$ and

$$D \sqcap E \sqsubseteq D',$$

for all $E \in \mathrm{atom}_\Sigma$. Let $\mathcal{K} = (\mathcal{O}^*, \mathcal{D})$.

*Claim.* $\mathcal{O} \cup \mathcal{O}'$ is a conservative extension of $\mathcal{O}$ in $\mathcal{ALCO}$ iff $(\mathcal{K}, \{a\}, \{b\})$ is not $\mathcal{ALCO}(\Sigma)$-separable, for both the projective and non-projective case.

*Proof of the Claim.* We consider the projective case. The non-projective case is similar and omitted. Consider an $\mathcal{ALCO}(\Sigma \cup \Sigma_{\text{help}})$-concept $C$, where $\Sigma_{\text{help}}$ is a set of fresh concept names. We show the following equivalences:

(1) $C$ is satisfiable w.r.t. $\mathcal{O}$ iff there exists a model $\mathfrak{A}$ of $\mathcal{K}$ such that $b^{\mathfrak{A}} \in C^{\mathfrak{A}}$.

(2) If $C$ is satisfiable w.r.t. $\mathcal{O} \cup \mathcal{O}'$, then there exists a model $\mathfrak{A}$ of $\mathcal{K}$ such that $a^{\mathfrak{A}} \in C^{\mathfrak{A}}$.

(3) Let $E \in \text{atom}_\Sigma$. If there exists a model $\mathfrak{A}$ of $\mathcal{K}$ such that $a^{\mathfrak{A}} \in (E \sqcap C)^{\mathfrak{A}}$, then $E \sqcap C$ is satisfiable w.r.t. $\mathcal{O} \cup \mathcal{O}'$.

For (1), assume that $C$ is satisfiable w.r.t. $\mathcal{O}$. Take a model $\mathfrak{A}$ of $\mathcal{O}$ satisfying $C$ in $d$. We define a model $\mathfrak{B}$ of $\mathcal{K}$ satisfying $C$ in $b^{\mathfrak{A}}$: to define $\text{dom}(\mathfrak{B})$, we add to $\text{dom}(\mathfrak{A})$ the individual $a$ and all individuals $c \in \text{sig}(\mathcal{O} \cup \mathcal{O}') \setminus \Sigma$. Then we set $b^{\mathfrak{B}} = d$ and interpret $a$ and the individuals in $\text{sig}(\mathcal{O} \cup \mathcal{O}') \setminus \Sigma$ by themselves, interpret $A$ by the domain of $\mathfrak{A}$, add the pairs $(a, c^{\mathfrak{A}})$, $c \in \text{sig}(\mathcal{O} \cup \mathcal{O}')$ to $s^{\mathfrak{B}}$, and set $D^{\mathfrak{B}} = a$ and $D'^{\mathfrak{B}} = \emptyset$. Then $\mathfrak{B}$ is a model of $\mathcal{O}$ satisfying $C$ in $b^{\mathfrak{B}}$.

The converse direction of (1) is clear.

For (2), suppose that $C$ is satisfiable w.r.t. $\mathcal{O} \cup \mathcal{O}'$. Take a model $\mathfrak{A}$ of $\mathcal{O} \cup \mathcal{O}'$ satisfying $C$ in $d$. We define a model $\mathfrak{B}$ of $\mathcal{K}$ satisfying $C$ in $a^{\mathfrak{B}}$: take $\mathfrak{A}$ and set $\text{dom}(\mathfrak{B}) = \text{dom}(\mathfrak{A})$. We interpret $b$ arbitrarily and set $A^{\mathfrak{B}} = D^{\mathfrak{B}} = D'^{\mathfrak{B}} = \text{dom}(\mathfrak{B})$. Finally, we add the pairs $(a^{\mathfrak{A}}, c^{\mathfrak{A}})$, $c \in \text{sig}(\mathcal{O} \cup \mathcal{O}')$, to $s^{\mathfrak{B}}$. Then $\mathfrak{B}$ is a model of $\mathcal{O} \cup \mathcal{O}'$ satisfying $C$ in $a^{\mathfrak{B}}$.

For (3), let $E \in \text{atom}_\Sigma$ and assume that $E \sqcap C$ is satisfied in a model $\mathfrak{A}$ of $\mathcal{K}$ at $a^{\mathfrak{A}}$. Then $a^{\mathfrak{A}} \in (D \sqcap E)^{\mathfrak{A}}$ and therefore $a^{\mathfrak{A}} \in D'^{\mathfrak{A}}$ since $D \sqcap E \sqsubseteq D' \in \mathcal{O} \cup \mathcal{O}'$. Hence $E \sqcap C$ is satisfiable w.r.t. $\mathcal{O} \cup \mathcal{O}'$ as it is satisfied in $\mathfrak{A}$ w.r.t. the directed relativization of $\mathcal{O} \cup \mathcal{O}'$ to $D'$.

Now, we finish the proof of the Claim. Suppose first that $(\mathcal{K}, \{a\}, \{b\})$ is projectively $\mathcal{ALCO}(\Sigma)$-separable and let $C$ be a separating concept. Then, there is a model $\mathfrak{A}$ of $\mathcal{K}$ such that $b \in (\neg C)^{\mathfrak{A}}$. By Point 1, $\neg C$ is satisfiable w.r.t. $\mathcal{O}$. Moreover, there is no model $\mathfrak{A}$ of $\mathcal{K}$ with $a \in (\neg C)^{\mathfrak{A}}$. By Point 2, $\neg C$ is not satisfiable w.r.t. $\mathcal{O} \cup \mathcal{O}'$. Hence, $\neg C$ is a witness concept for $\mathcal{O}, \mathcal{O} \cup \mathcal{O}'$.

Conversely, assume that $\mathcal{O} \cup \mathcal{O}'$ is not a projective conservative extension of $\mathcal{O}$ in $\mathcal{ALCO}$ and let $C$ witness this. By our assumption on $\text{atom}_\Sigma$, there exists an $E \in \text{atom}_\Sigma$ such that $E \sqcap C$ is also satisfiable w.r.t. $\mathcal{O}$, but not satisfiable w.r.t. $\mathcal{O} \cup \mathcal{O}'$. Thus, by Point 1, there exists a model $\mathfrak{A}$ of $\mathcal{K}$ such that $b^{\mathfrak{A}} \in (E \sqcap C)^{\mathfrak{A}}$ and, by Point 3, there does not exist a model $\mathfrak{A}$ of $\mathcal{K}$ such that $a^{\mathfrak{A}} \in (E \sqcap C)^{\mathfrak{A}}$. Thus, $(\mathcal{K}, \{a\}, \{b\})$ is projectively $\mathcal{ALCO}(\Sigma)$-separable, namely by $\neg(E \sqcap C)$.

This finishes the proof of the Claim.

The proof above is easily adapted for $\mathcal{ALCI}$ and $\mathcal{ALCIO}$. In fact, one only has to add to the directed relativization of $\mathcal{O}^A$ the inclusions $A \sqsubseteq \forall r^-.A$, for $r$ any role in $\Sigma$. ❑

**Preliminaries for Tree Automata** A *tree* is a non-empty (and potentially infinite) set of words $T \subseteq (\mathbb{N} \setminus 0)^*$ closed under prefixes. A node $w \in T$ is a *successor* of $v \in T$ if $w = v \cdot i$ for some $i \in \mathbb{N}$. Moreover, $w$ is an *ancestor* of $v$ if $w$ is a prefix of $v$. A tree is binary if every node has either zero or two successors. For an alphabet $\Theta$, a $\Theta$-*labeled tree* is a pair $(T, \tau)$ with $T$ a tree and $\tau : T \to \Theta$ a node labeling function.

A *two-way alternating tree automaton (2ATA) over binary trees* is a tuple $\mathcal{A} = (Q, \Theta, q_0, \delta, \Omega)$ where $Q$ is a finite set of *states*, $\Theta$ is the finite *input alphabet*, $q_0 \in Q$ is the *initial state*, $\delta$ is a *transition function* as specified below, and $\Omega : Q \to \mathbb{N}$ is a *priority function*. The transition function maps a state $q$ and some input letter $\theta \in \Theta$ to a *transition condition* $\delta(q, \theta)$ which is a positive Boolean formula over the truth constants $\mathsf{true}$ and $\mathsf{false}$ and transitions of the form $q$, $\langle - \rangle q$, $[-]q$, $\Diamond q$, $\Box q$ where $q \in Q$. Informally, the transition $q$ expresses that a copy of the automaton is sent to the current node in state $q$, $\langle - \rangle q$ means that a copy is sent in state $q$ to the predecessor node, which is then required to exist, $[-]q$ means the same except that the predecessor node is not required to exist, $\Diamond q$ means that a copy is sent in state $q$ to some successor, and $\Box q$ that a copy is sent in state $q$ to all successors. The semantics is defined in terms of runs in the usual way (Vardi 1998), see below. We use $L(\mathcal{A})$ to denote the set of all $\Theta$-labeled binary trees accepted by $\mathcal{A}$. The *emptiness problem*, which asks whether $L(\mathcal{A}) = \emptyset$ for a given 2ATA $\mathcal{A}$, can be decided in time exponential in the number of states of $\mathcal{A}$ (Vardi 1998).

We make precise the semantics of 2ATAs. Let $\mathcal{A} = (Q, \Theta, q_0, \delta, \Omega)$ be a 2ATA and $(T, L)$ a $\Theta$-labeled tree. A *run for $\mathcal{A}$ on* $(T, L)$ is a $T \times Q$-labeled tree $(T_r, r)$ such that:

- $\varepsilon \in T_r$ and $r(\varepsilon) = (\varepsilon, q_0)$;
- For all $y \in T_r$ with $r(y) = (x, q)$ and $\delta(q, L(x)) = \varphi$, there is an assignment $v$ of truth values to the transitions in $\varphi$ such that $v$ satisfies $\varphi$ and:
  - if $v(p) = 1$, then $r(y') = (x, p)$ for some successor $y'$ of $y$ in $T_r$;
  - if $v(\langle - \rangle p) = 1$, then $x \neq \varepsilon$ and there is a successor $y'$ of $y$ in $T_r$ with $r(y') = (x \cdot -1, p)$;
  - if $v([-]p) = 1$, then $x = \varepsilon$ or there is a successor $y'$ of $y$ in $T_r$ such that $r(y') = (x \cdot -1, p)$;
  - if $v(\Diamond p) = 1$, then there is a successor $x'$ of $x$ in $T$ and a successor $y'$ of $y$ in $T_r$ such that $r(y') = (x', p)$;
  - if $v(\Box p) = 1$, then for every successor $x'$ of $x$ in $T$, there is a successor $y'$ of $y$ in $T_r$ such that $r(y') = (x', p)$.

Let $\gamma = i_0 i_1 \cdots$ be an infinite path in $T_r$ and denote, for all $j \geq 0$, with $q_j$ the state such that $r(i_0 \cdots i_j) = (x, q_j)$. The path $\gamma$ is *accepting* if the largest number $m$ such that $\Omega(q_j) = m$ for infinitely many $j$ is even. A run $(T_r, r)$ is accepting, if all infinite paths in $T_r$ are accepting. Finally, a tree is accepted if there is some accepting run for it.

It is well-known that 2ATAs are closed under complementation, intersection, and projection.

**Lemma 16** *Given 2ATAs $\mathcal{A}_1, \mathcal{A}_2$ over alphabet $\Theta$ and a mapping $h : \Theta \to \Theta'$, we can compute:*

- *in polynomial time a 2ATA $\overline{\mathcal{A}_1}$ such that $L(\overline{\mathcal{A}_1}) = \overline{L(\mathcal{A}_1)}$ and the number of states of $\overline{\mathcal{A}_1}$ equals the number of states of $\mathcal{A}_1$;*
- *in polynomial time a 2ATA $\mathcal{A}$ such that $L(\mathcal{A}) = L(\mathcal{A}_1) \cap L(\mathcal{A}_2)$ and the number of states of $\mathcal{A}$ is $1 + n_1 + n_2$, $n_i$ the number of states of $\mathcal{A}_i$;*
- *in exponential time a 2ATA $\mathcal{A}_h$ such that $L(\mathcal{A}_h) = \{(T, h(\tau)) \mid (T, \tau) \in L(\mathcal{A}_1)\}$ and the number of states of $\mathcal{A}_h$ is exponential in the number of states of $\mathcal{A}_1$.*

**Encoding of $\mathcal{L}$-Forest Models** Let $\mathcal{L} \in \{\mathcal{ALCI}, \mathcal{ALCO}\}$ and fix an $\mathcal{L}$-KB $\mathcal{K} = (\mathcal{O}, \mathcal{D})$. In order to work with tree automata, we need to encode $\mathcal{L}$-forest models of $\mathcal{K}$ of finite $\mathcal{L}$-outdegree as input to the tree automata. Since 2ATAs run over binary trees, we need to appropriately encode the arbitrary outdegree, which is done similar to (Jung, Lutz, and Zeume 2020).

More precisely, we use the alphabet $\Theta$ defined by

$$\Theta = \{\circ\} \cup \text{Rol}(\mathcal{K}) \times 2^{\text{ind}(\mathcal{D}) \cup (\text{sig}(\mathcal{K}) \cap \mathsf{N_C})} \times 2^{\mathcal{F}},$$

where $\text{Rol}(\mathcal{K})$ denotes the set of all role names that occur in $\mathcal{K}$ and their inverses, and $\mathcal{F}$ is the set of all pairs $(r, a)$ with $r$ a role name that in $\text{sig}(\mathcal{K})$ and $a \in \text{ind}(\mathcal{D})$. Intuitively, a node $w \in T$ with $\tau(w) = (R, M, F)$ encodes an element that satisfies precisely the concepts in $M$; moreover, $F$ describes its connections to elements in $\text{ind}(\mathcal{D})$ and $R$ is the "incoming role". The symbol '$\circ$' is a label for dummy nodes that we need for encoding arbitrary finite outdegree into binary trees: we simply introduce as many intermediate $\circ$-labeled nodes as needed to achieve the required outdegree at each node.

More formally, let $(T, \tau)$ be a $\Theta$-labeled tree. For each $w \in T$ with $\tau(w) \neq \circ$, let $w^\uparrow$ denote the unique ancestor $w'$ of $w$ in $T$ (if existing) such that $\tau(w') \neq \circ$ and $\tau(w'') = \circ$ for all $w''$ between $w'$ and $w$. We call $(T, \tau)$ *well-formed* if

1. for every $a \in \text{ind}(\mathcal{D})$, there is a unique element $w_a \in T$ such that $\tau(w_a) = (R, M, F)$ for some $R, F$ and $a \in M$;
2. for every $w \in T$ with $\tau(w) \neq \circ$, either $w = w_a$, for some $a$, and all ancestors of $w$ are labeled with $\circ$, or $w$ has an ancestor $w_a$, for some $a$.

A well-formed $\Theta$-labeled tree $(T, \tau)$ gives rise to a structure $\mathfrak{A}_\tau$ with $\text{dom}(\mathfrak{A}_\tau) = \{w \in T \mid \tau(w) \neq \circ\}$ as follows:

$$\begin{aligned}
a^{\mathfrak{A}_\tau} &= w_a \\
A^{\mathfrak{A}_\tau} &= \{w \in T \mid \tau(w) = (S, M, F) \text{ and } A \in M\} \\
r^{\mathfrak{A}_\tau} &= \{(w^\uparrow, w) \mid \tau(w) = (r, M, F) \text{ and } w^\uparrow \text{ defined}\} \cup \\
&\quad \{(w, w^\uparrow) \mid \tau(w) = (r^-, M, F) \text{ and } w^\uparrow \text{ defined}\} \cup \\
&\quad \{(w, w_a) \mid \tau(w) = (S, M, F) \text{ and } (r, a) \in F\}
\end{aligned}$$

for all $a \in \mathsf{N_I}$, $A \in \mathsf{N_C}$, and $r \in \mathsf{N_R}$ (and for $a \in \mathsf{N_I} \setminus \text{ind}(\mathcal{D})$, $w_a$ denotes an arbitrary element of $\mathfrak{A}$).

Conversely, every finite outdegree forest model $\mathfrak{A}$ of $\mathcal{K}$ can be encoded (up to isomorphism) as well-formed $\Theta$-labeled tree. To show this, we start with a not-necessarily binary well-formed $\Theta$-labeled tree $(T, \tau)$ that encodes $\mathfrak{A}$; $(T, \tau)$ can easily be made binary by introducing intermediate $\circ$-labeled nodes. Let $D = \{a^{\mathfrak{A}} \mid a \in \text{ind}(\mathcal{D})\}$ and associate sets $M_d, F_d$ to every element $d \in \text{dom}(\mathfrak{A})$ by taking

$$\begin{aligned}
M_d &= \{A \in \text{sig}(\mathcal{K}) \mid d \in A^{\mathfrak{A}}\} \cup \{a \in \text{ind}(\mathcal{D}) \mid d = a^{\mathfrak{A}}\} \\
F_d &= \{(r, e) \mid (d, e) \in r^{\mathfrak{A}}, e \in D\}
\end{aligned}$$

To start the construction of $(T, \tau)$, we set $\tau(\varepsilon) = \circ$ and add a successor $w_d$ of $\varepsilon$ for every $d \in D$ and label it with $\tau(w_d) = (S, M_d, F_d)$ for an arbitrary role name $S$.

For the rest of the construction, let $\mathfrak{A}_d, d \in D$ be the $\mathcal{L}$-trees which exist since $\mathfrak{A}$ is $\mathcal{L}$-forest model of $\mathcal{K}$. Recall that $\mathfrak{A}_d$ is rooted at $d$. Now $(T, \tau)$ is obtained by exhaustively applying the following rule:

$(*)$ If $w_e$ is defined, for an element $e$ of some $\mathfrak{A}_d$ and $f$ is a successor of $e$ in $\mathfrak{A}_d$ with $w_f$ undefined, add a fresh successor $w_f$ of $w_e$ to $T$ and set $\tau(w_f) = (R, M_f, F_f)$ where $R$ is the unique role such that $(e, f) \in R^{\mathfrak{A}}$.

It is not difficult to construct 2ATAs that accept precisely the forest models of $\mathcal{K}$, see for example (Jung et al. 2017) for full details of a similar automata construction.

**Lemma 17** *For $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}, \mathcal{ALCO}\}$ and $\mathcal{L}$-KBs $\mathcal{K}$, we can construct in time polynomial in $||\mathcal{K}||$ 2ATAs $\mathcal{A}_0, \mathcal{A}_\mathcal{K}$ such that:*

1. *$\mathcal{A}_0$ accepts precisely the well-formed $\Theta$-labeled trees;*
2. *$\mathcal{A}_\mathcal{K}$ accepts a well-formed $\Theta$-labeled tree $(T, \tau)$ iff $\mathfrak{A}_\tau$ is a finite outdegree forest model of $\mathcal{K}$.*

## E.1 $\mathcal{ALCI}$ and $\mathcal{ALC}$

We concentrate on $\mathcal{ALCI}$, the case of $\mathcal{ALC}$ is similar.

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

**Proof.** "if". Take a forest model $\mathfrak{A}$ and some $a \in P$ with $\mathcal{D}_{\text{con}(a)}, a \to_c^\Sigma \mathfrak{A}, b^{\mathfrak{A}}$. Moreover, fix a $\Sigma$-homomorphism $h$ and $\mathcal{K}$-types $t_d$, $d \in \text{ind}(\mathcal{D})$ that witness that. Take models $\mathfrak{B}_d$ of $\mathcal{O}$ such that $\mathfrak{B}_d, d \sim_{\mathcal{ALCI},\Sigma} \mathfrak{A}, h(d)$. We may assume that the $\mathfrak{B}_d$ are tree-shaped with root $d$ and that the bisimulations are functions $f_d$. Now attach to every $d \in \text{ind}(\mathcal{D})$ the model $\mathfrak{B}_d$ and obtain $\mathfrak{B}$ by adding $(d, d')$ to $r^{\mathfrak{B}}$ if $r(d, d') \in \mathcal{D}$. Then

$$f = \bigcup_{d \in \text{ind}(\mathcal{D})} f_d$$

is a functional $\mathcal{ALCI}(\Sigma)$-bisimulation between $\mathfrak{B}$ and $\mathfrak{A}$.

"only if". Take a forest model $\mathfrak{A}$, some $a \in P$, and a model $\mathfrak{B}$ of $\mathcal{K}$ such that $\mathfrak{B}, a^{\mathfrak{B}} \sim^f_{\mathcal{ALCI},\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$. Let $f$ be a functional $\mathcal{ALCI}(\Sigma)$-bisimulation witnessing that. The restriction $h$ of $f$ to $\text{ind}(\mathcal{D})$ and types $t_d = \text{tp}_\mathcal{K}(\mathfrak{B}, d^{\mathfrak{B}})$, for all $d \in \text{ind}(\mathcal{D})$ witness $\mathcal{D}_{\text{con}(a)}, a \to_c^\Sigma \mathfrak{A}, b^{\mathfrak{A}}$. ❏

**Lemma 18** *Let $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$. For each $a \in P$, there is a 2ATA $\mathcal{A}_a$ such that $\mathcal{A}_a$ accepts a well-formed labeled tree $(T, \tau)$ iff $\mathcal{D}_{con(a)}, a \to_c^\Sigma \mathfrak{A}, b^{\mathfrak{A}}$. Moreover, $\mathcal{A}_a$ can be constructed in double exponential time in $||\mathcal{K}||$ and has exponentially many states.*

**Proof.** We sketch the construction of the automata $\mathcal{A}_a$, $a \in P$. Let $(T, \tau)$ be a well-formed input tree. As the first step, $\mathcal{A}_a$ non-deterministically guesses the following:

- types $t_d$, $d \in \text{ind}(\mathcal{D}_{\text{con}(a)})$, such that $(\mathcal{O}, \mathcal{D}')$ is satisfiable where $\mathcal{D}' = \mathcal{D} \cup \{C(d) \mid C \in t_d,\ d \in \text{ind}(\mathcal{D}_{\text{con}(a)})\}$;
- a partition $\mathcal{D}_0, \mathcal{D}_1, \dots, \mathcal{D}_m$ of $\mathcal{D}_{\text{con}(a)}$ such that $\text{ind}(\mathcal{D}_i) \cap \text{ind}(\mathcal{D}_j) = \emptyset$ for $1 \leq i < j \leq m$;
- a mapping $h$ from $\text{ind}(\mathcal{D}_0)$ to $\text{ind}(\mathcal{D})$ such that $h(a) = b$ and there are $a_1, \dots, a_m$ with $h(c) = a_i$ for all $c \in \text{ind}(\mathcal{D}_0 \cap \mathcal{D}_i)$ and $1 \leq i \leq m$.

The entire guess is stored in the state of the automaton. Note that the first item above ensures that Item (ii) from the definition of $\mathcal{D}_{\text{con}(a)}, a \to_c^\Sigma \mathfrak{A}_\tau, b^{\mathfrak{A}_\tau}$ is satisfied for the guessed types $t_d$. Moreover, elements $d^{\mathfrak{B}}$ for $d \in \text{ind}(\mathcal{D}_0)$ are intuitively mapped to $h(d)^{\mathfrak{A}}$ while elements $d^{\mathfrak{B}}$ for $d \in \text{ind}(\mathcal{D}_i) \setminus \text{ind}(\mathcal{D}_0)$ are mapped to the trees below $a_i$.

After making its guess, the automaton verifies first $h$ is a $\Sigma$-homomorphism from $\mathcal{D}_0$ to $\mathfrak{A}_\tau$ by sending out copies for all facts in $\mathcal{D}_0$ to the respective elements in $\mathfrak{A}_\tau$. Then, it verifies that this homomorphism can be extended to a homomorphism from $\mathcal{D}_{\text{con}(a)}$ to $\mathfrak{A}_\tau$ that satisfies Item (i) from the definition of $\mathcal{D}_{\text{con}(a)}, a \to_c^\Sigma \mathfrak{A}_\tau, b^{\mathfrak{A}_\tau}$. To this end, it does a top-down traversal of $\mathfrak{A}_\tau$ checking that each $\mathcal{D}_i$ can be homomorphically mapped to the subtree of $\mathfrak{A}_\tau$ below $h(a_i)^{\mathfrak{A}_\tau}$. During the traversal, the automaton memorizes in its state the set of individuals from $\mathcal{D}_i$ that are mapped to the currently visited element.

The automaton additionally makes sure that Item (i) from the definition of $\mathcal{D}_{\text{con}(a)}, a \to_c^\Sigma \mathfrak{A}_\tau, b^{\mathfrak{A}_\tau}$ is satisfied, in the following way. During the top-down traversal, it spawns copies of itself to verify that, whenever it has decided to map a $d \in \text{ind}(\mathcal{D}_{\text{con}(a)})$ to the current element, then there is a tree-shaped model $\mathcal{B}_d$ of $\mathcal{O}$ with $\text{tp}_{\mathcal{K}}(\mathfrak{B}_d, d) = t_d$ and a bisimulation that witnesses $\mathcal{B}_d, d \sim_{\mathcal{ALCI},\Sigma} \mathfrak{A}_\tau, c$. This is done by 'virtually' traversing $\mathcal{B}_d$ elements-by-element, storing at each moment only the type of the current element in a state. This is possible because $\mathfrak{B}_d$ is tree-shaped. At the beginning, the automaton is at an element of $\mathcal{B}_d$ of type $t_d$ and knows that the bisimulation maps this element to the node of $\mathfrak{A}_\tau$ currently visited by the automaton. It then does two things to verify the two main conditions of bisimulations. First, it transitions to every neighbor of the node of $\mathfrak{A}_\tau$ currently visited, both upwards and downwards, and carries out in its state the corresponding transition in $\mathcal{B}_d$, in effect guessing a new type. Second, it considers the current type of $\mathcal{B}_d$ and guesses successor types that satisfy the existential restrictions in it. For every required successor type, it then guesses a neighbor of the currently visited node in $\mathfrak{A}_\tau$ to which the successor is mapped. The two steps are alternated, exploiting the alternation capabilities of the automaton. Some extra bookkeeping in states is needed for the root node of the input tree as it represents more than one element of $\mathfrak{A}_\tau$.

It can be verified that only exponentially many states are required and that the transition function can be computed in double exponential time in $||\mathcal{K}||$. ❑

We can now finish the proof of the upper bound in Theorem 6. By Lemmas 17, 5, and 18 and Theorem 4, $(\mathcal{K}, P, \{b\})$ is projectively $\mathcal{L}(\Sigma)$-separable iff

$$L(\mathcal{A}_0) \cap L(\mathcal{A}_\mathcal{K}) \cap \textstyle\bigcap_{a \in P} \overline{L(\mathcal{A}_a)} \neq \emptyset$$

where $\overline{L(\mathcal{A}_a)}$ denotes the complement of $L(\mathcal{A}_a)$. By Lemmas 17 and 18 all these automata can be constructed in double exponential time and their number of states is single exponential in $||\mathcal{K}||$ for $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$. By Lemma 16, we can compute in polynomial time an automaton that accepts precisely the language on the left-hand side of the above inequality. It remains to recall that non-emptiness of 2ATAs can be decided in time exponential in the number of states.

## E.2 Upper Bound for $\mathcal{ALCO}$

We give here only the upper bound for Theorem 7. The lower bound is proved for conservative extensions in Section F.

We use the same definition of $\mathcal{K}$-types as in $\mathcal{ALCI}$ except that we assume without loss of generality that $\{c\} \in \text{sub}(\mathcal{K})$, for all $c \in \text{ind}(\mathcal{D})$. Denote with $\text{TP}(\mathcal{K})$ the set of all types.

We work with $(\Theta \times \Theta')$-labeled trees for

$$\Theta' = \{\circ\} \cup 2^{\text{ind}(\mathcal{D})}$$

A $(\Theta \times \Theta')$-labeled tree $(T, \tau_1, \tau_2)$ is *well-formed* if $(T, \tau_1)$ is well-formed and for each $c \in \text{ind}(\mathcal{D})$, there is exactly one $w \in T$ with $c \in \tau_2(w)$; we denote this with $v_c$.

**Lemma 19** *There is a 2ATA $\mathcal{A}'_0$ which accepts precisely the well-formed $(\Theta \times \Theta')$-labeled trees.*

**Lemma 20** *For every $a \in P$, there is a 2ATA $\mathcal{A}'_a$ which accepts a well-formed $(\Theta \times \Theta')$-labeled tree $(T, \tau_1, \tau_2)$ iff there is a forest model $\mathfrak{B}$ of $\mathcal{K}$ and a function $f : dom(\mathfrak{B}) \to dom(\mathfrak{A}_{\tau_1})$ such that*

*(a) $f$ witnesses $\mathfrak{B}, a^{\mathfrak{B}} \sim^f_{\mathcal{ALCO},\Sigma} \mathfrak{A}_{\tau_1}, b^{\mathfrak{A}_{\tau_1}}$;*

*(b) for every $c \in ind(\mathcal{D})$, we have $f(c^{\mathfrak{B}}) = v_c$.*

*Moreover, $\mathcal{A}'_a$ can be constructed in exponential time in $||\mathcal{K}||$ and has at most exponentially many states.*

**Proof.** Let $(T, \tau_1, \tau_2)$ be a well-formed $(\Theta \times \Theta')$-labeled tree. We sketch the function of $\mathcal{A}'_a$. Intuitively, $\mathcal{A}'_a$ constructs an $\mathcal{ALCO}$-forest model $\mathfrak{B}$ of $\mathcal{K}$ and the witnessing bisimulation $f$ "on the fly" by visiting the nodes in the input in states that store the type of the currently visited element of $\mathfrak{B}$. The construction of $\mathfrak{B}$ is started at the individual names, that is, at the nodes $v_c \in T$, $c \in \text{ind}(\mathcal{D})$. Before the actual construction of $\mathfrak{B}$ can start, $\mathcal{A}'_a$ guesses the types of the individual names $c \in \text{ind}(\mathcal{D})$ in the model $\mathfrak{B}$ and keeps that guess in its state throughout the entire run. Then it spawns a copy of itself in every $v_c \in T$, $c \in \text{ind}(\mathcal{D})$ in state corresponding to $t_c$.

Whenever $\mathcal{A}'_a$ visits a node $w \in T$ in some type $t$, this represents an obligation to extend the bisimulation constructed so far to an element $e$ of type $t$ in $\mathfrak{B}$ and the element $w$ of $\mathfrak{A}_{\tau_1}$. This can be done similarly to what is done in the proof of Lemma 18 since $\mathfrak{B}$ is forest-shaped. We need to argue,

however, how to deal with the individuals since they can be accessed from everywhere in $\mathfrak{B}$. When the automaton tries to extend $\mathfrak{B}$ with a successor that satisfies some nominal $\{c\}$, we have to make sure that that successor gets type $t_c$. Conversely, when we try to find a type bisimilar to some successor of $w$, we make sure that we can only visit nodes of the form $v_c$ with types $t_c$. In both cases, it is crucial that the automaton can stop the extension, because we have already started a copy of the automaton with type $t_c$ in $v_c$ (in the very beginning). ❑

We finish the proof of the upper bound of Theorem 7. By Lemma 16, we can compute in exponential time (in the size of $\mathcal{A}'_a$) an automaton $\mathcal{A}_a$ with

$$L(\mathcal{A}_a) = \{(T, \tau_1) \mid (T, \tau_1, \tau_2) \in L(\mathcal{A}'_a)\},$$

that is, $\mathcal{A}_a$ is the projection of $\mathcal{A}'_a$ to the first component $\tau_1$. It should be clear that $\mathcal{A}_a$ accepts a forest model $\mathfrak{A}$ of $\mathcal{K}$ iff there is a model $\mathfrak{B}$ of $\mathcal{K}$ such that $\mathfrak{B}, a^{\mathfrak{B}} \sim^f_{\mathcal{ALCO},\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$. Thus, we can proceed as for $\mathcal{ALCI}$ and compute the desired automaton $\mathcal{A}$ such that it accepts the language

$$L(\mathcal{A}) = L(\mathcal{A}_0) \cap L(\mathcal{A}_\mathcal{K}) \cap \textstyle\bigcap_{a \in P} \overline{L(\mathcal{A}_a)},$$

which is non-empty iff $(\mathcal{K}, P, \{b\})$ is projectively $\mathcal{ALCO}(\Sigma)$-satisfiable. Since $\mathcal{A}_a$ (and thus $\mathcal{A}$) is a 2ATA with double exponential many states and non-emptiness can be checked in exponential time in the number of states, the 3EXPTIME-upper bound follows.

# F 3EXPTIME Lower Bound for Conservative Extensions in $\mathcal{ALCO}$

We start with model theoretic characterizatons of conservative extensions and projective conservative extensions in $\mathcal{ALC}$. We define a model $\mathfrak{A}$ of an $\mathcal{ALCO}$-ontology $\mathcal{O}$ to be a *forest model* of $\mathcal{O}$ of finite outdegree if it is an $\mathcal{ALCO}$-forest model of the KB $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ of finite $\mathcal{ALCO}$-outdegree, where $\mathcal{D} = \{D_b(b) \mid b \in \text{ind}(\mathcal{O})\}$ and the $D_b$ are fresh concept names, one for each individual $b$ in $\mathcal{O}$.

**Theorem 13** *Let $\mathcal{O}$ and $\mathcal{O}'$ be $\mathcal{ALCO}$-ontologies and $\Sigma = sig(\mathcal{O})$. Then*

1. *the following conditions are equivalent:*
   (a) *$\mathcal{O} \cup \mathcal{O}'$ is a conservative extension of $\mathcal{O}$;*
   (b) *for every pointed forest model $\mathfrak{A}, a$ of $\mathcal{O}$ of finite outdegree there exists a pointed model $\mathfrak{B}, b$ of $\mathcal{O} \cup \mathcal{O}'$ such that $\mathfrak{B}, b \sim_{\mathcal{ALCO},\Sigma} \mathfrak{A}, a$.*
2. *the following conditions are equivalent:*
   (a) *$\mathcal{O} \cup \mathcal{O}'$ is a projective conservative extension of $\mathcal{O}$;*
   (b) *for every pointed forest model $\mathfrak{A}, a$ of $\mathcal{O}$ of finite outdegree there exists a pointed model $\mathfrak{B}, b$ of $\mathcal{O} \cup \mathcal{O}'$ such that $\mathfrak{B}, b \sim^f_{\mathcal{ALCO},\Sigma} \mathfrak{A}, a$.*

**Theorem 14** *Given an $\mathcal{ALC}$-ontology $\mathcal{O}$ and an $\mathcal{ALCO}$-ontology $\mathcal{O}'$, it is* 3EXPTIME*-hard to decide whether $\mathcal{O}'$ is a conservative extension of $\mathcal{O}$.*

Since $\mathcal{O}$ is an $\mathcal{ALC}$-ontology, we have no nominals available in witness concepts and every witness concept must actually be an $\mathcal{ALC}$-concept. It is interesting to note that we use only a single nominal in the ontologies $\mathcal{O}'$.

The proof of Theorem 14 follows the general outline of the 2EXPTIME lower bound for conservative extensions in $\mathcal{ALC}$ that is proved in (Ghilardi, Lutz, and Wolter 2006). As in that paper, we proceed in two steps. We first establish a lower bound on the size of witness concepts and then extend the involved ontologies to obtain the 3EXPTIME lower bound. In fact, the first step is the technically subtle one and we present it in full detail. The second step is then rather simple and we only sketch the required constructions.

## F.1 Large Witness Concepts

**Theorem 15** *For every $n > 0$, there is an $\mathcal{ALC}$-ontology $\mathcal{O}_n$ and an $\mathcal{ALCO}$-ontology $\mathcal{O}'_n$ such that the size of $\mathcal{O}_n$ and $\mathcal{O}'_n$ is polynomial in $n$, $\mathcal{O}_n \cup \mathcal{O}'_n$ is not a conservative extension of $\mathcal{O}_n$, and every witness concept $C$ for $\mathcal{O}_n$ and $\mathcal{O}'_n$ is of size at least $2^{2^{2^n}}$.*

To prove Theorem 15, we craft $\mathcal{O}_n$ and $\mathcal{O}'_n$ so that $\mathcal{O}_n \cup \mathcal{O}'_n$ is not a conservative extensions of $\mathcal{O}_n$, but every witness concept $C$ for $\mathcal{O}_n$ and $\mathcal{O}'_n$ must enforce in models of $\mathcal{O}_n$ a binary tree of depth at least $2^{2^{2^n}}$. This implies that $C$ is of size at least $2^m$ because $C$ is an $\mathcal{ALC}$-concept and $\mathcal{O}_n$ does not introduce additional elements via existential restrictions. Let us be a bit more precise about the trees. The 'nodes' of the tree are actually paths of length $2^n \cdot 2^{2^n}$ and no branching occurs inside these paths. Thus, when counting also the intermediate domain element on the 'node paths', then the trees are really of depth $2^n \cdot 2^{2^n} \cdot 2^{2^{2^n}}$. We use a single role name $r$ to attach successors, both when branching occurs and when no branching occurs. At branching nodes, the left successor is marked with concept name $S_L$ and the right successor is marked with concept name $S_R$. Successors of non-branching nodes must be marked with at least one of $S_L$ and $S_R$. And finally, the (first element of the) root (path) is labeled with concept name $A$.

A main feature of $\mathcal{O}_n$ and $\mathcal{O}'_n$ is to implement a binary counter that can count up to $m := 2^{2^{2^n}}$, the desired depth of the trees (not counting intermediate domain elements). In fact, $\mathcal{O}_n$ and $\mathcal{O}'_n$ implement three binary counters that build upon each other so that the third counter can achieve the intended counting range. Counter 1 has $n$ bits and counts from 0 to $2^n - 1$, Counter 2 has $2^n$ bits and counts up to $2^{2^n} - 1$, and Counter 3 has $2^{2^n}$ bits and counts up to $2^{2^{2^n}} - 1$. We use Counter 1 to describe the bit positions of Counter 2 and Counter 2 to describe the bit positions of Counter 3. Counter 1 and Counter 2 count modulo their maximum value plus one while Counter 3 needs to reach its maximum value only once.

Counter 1 uses concept names $C_0, \ldots, C_{n-1}$ as bits. Thus a counter value of Counter 1 can be represented at a single domain element. In contrast, a value of Counter 2 is spread out accross a sequence of $2^n$ domain elements, which we call a *Counter 2 sequence*. The bit positions of Counter 2

in such a sequence are identified by Counter 1 and concept name $X_2$ is used to indicate the bit value of Counter 2 at each position. A value of Counter 3, in turn, is spread out accross a sequence of $2^{2^n}$ Counter 2 sequences, that is, $2^n \cdot 2^{2^n}$ domain elements in total. We call this a *Counter 3 sequence* and it is these Counter 3 sequences that constitute the 'nodes' of the trees mentioned above. Each of the Counter 2 subsequences represents one bit position of Counter 3 and stores one bit value via the concept name $X_3$ that must be interpreted uniformly in that subsequence.

With a *path* in an structure $\mathfrak{A}$, we mean a sequence $p = d_0, \dots, d_n$ of elements of $\mathrm{dom}(\mathfrak{A})$ such that $(d_i, d_{i+1}) \in r^{\mathfrak{A}}$ and $d_{i+1} \in S_L^{\mathfrak{A}} \cup S_R^{\mathfrak{A}}$ for all $i < n$. We say that such a path is *properly counting* if the concept names $C_0, \dots, C_{n-1}, X_2, X_3$ are interpreted along the path in accordance with the counting strategy outlined above, all three counters starting with counter value zero. We now formalize the trees described above. A *counting tree* in $\mathfrak{A}$ is a collection $T$ of (not necessarily distinct) domain elements $d_{w,i}$, $w \in \{L, R\}^*$ with $|w| < m$ and $0 \leq i < 2^n \cdot 2^{2^n}$, such that the following conditions are satisfied:

1. $d_{\varepsilon,0} \in A^{\mathfrak{A}}$,
2. $(d_{w,i}, d_{w,i+1}) \in r^{\mathfrak{A}}$ and $d_{w,i+1} \in S_L^{\mathfrak{A}} \cup S_R^{\mathfrak{A}}$ for all $d_{w,i+1} \in T$;
3. $(d_{w,2^n \cdot 2^{2^n}}, d_{wL,0}) \in r^{\mathfrak{A}}$ and $d_{wL,0} \in S_L^{\mathfrak{A}}$ for all $d_{wL,0} \in T$;
4. $(d_{w,2^n \cdot 2^{2^n}}, d_{wR,0}) \in r^{\mathfrak{A}}$ and $d_{wR,0} \in S_R^{\mathfrak{A}}$ for all $d_{wR,0} \in T$;
5. every path in $\mathfrak{A}$ that uses only elements from $T$ is properly counting.

The element $d_{\varepsilon,0}$ is the *root* of the counting tree. We say that $\mathfrak{A}$ is *witnessing* if there is a $d \in A^{\mathfrak{A}}$ such that the following conditions are satisfied:

- $d$ is the root of a counting tree;
- every path that starts at $d$ and is of length at most $m$ is properly counting.

We are now in a position to describe more concretely what we want to achieve. Let $m' := 2^n \cdot 2^{2^n} \cdot m$. For $0 \leq i \leq m'$, let $S_i$ denote the set of concept names from $S := \{C_0, \dots, C_{n-1}, X_2, X_3\}$ that the $i$-th domain element on a path that is properly counting must satisfy. Then set

$$\begin{array}{rcl} D_0 & := & \top \\ D_{i+1} & := & \exists r.(S_L \sqcap D_i) \sqcap \exists r.(S_R \sqcap D_i) \\ C_i & := & A \sqcap D_i \sqcap \bigsqcap_{0 \leq i \leq m'} \forall r^i.(\bigsqcap S_i \sqcap \neg \bigsqcup S \setminus S_i) \end{array}$$

Note that every model of $C'_m$ is witnessing. We craft $\mathcal{O}_n$ and $\mathcal{O}'_n$ such that the following holds.

**Lemma 21**

1. *$C_{m'}$ is a witness concept for $\mathcal{O}_n$ and $\mathcal{O}'_n$;*
2. *for every pointed model $\mathfrak{A}, d$ of $\mathcal{O}_n$ of finite outdegree that is not witnessing, there is a pointed model $\mathfrak{B}, e$ of $\mathcal{O}_n \cup \mathcal{O}'_n$ such that $\mathfrak{B}, e \sim^f_{\mathcal{ALCO},\Sigma} \mathfrak{A}, d$ where $\Sigma = sig(\mathcal{O}_n)$.*

Point 2 implies that every witness concept for $\mathcal{O}_n$ and $\mathcal{O}'_n$ must have size at least $2^m$. In fact, if an $\mathcal{ALC}$-concept $C$ that is satisfiable w.r.t. $\mathcal{O}_n$ does not mention all paths in a counting tree, including their labeling with $S_L$ and $S_R$, then there is a model $\mathfrak{A}$ of $C$ and $\mathcal{O}_n$ in which there is no counting tree. Informally, this is because $\mathcal{O}_n$ does not introduce additional elements via existential quantifiers; for a rigorous proof of an almost identical statement, see (Ghilardi, Lutz, and Wolter 2006). We argue that such a $C$ cannot be a witness concept. Let $d \in C^{\mathfrak{A}}$. By Point 2, there is a pointed model $\mathfrak{B}, e$ of $\mathcal{O}_n \cup \mathcal{O}'_n$ such that $\mathfrak{B}, e \sim^f_{\mathcal{ALCO},\Sigma} \mathfrak{A}, d$. Since $\mathcal{ALCO}$-concepts are preserved under $\mathcal{ALCO}$-bisimulations, this implies $e \in C^{\mathfrak{B}}$. Consequently, $C$ is satisfiable w.r.t. $\mathcal{O}_n \cup \mathcal{O}'_n$ and is not a witness concept for $\mathcal{O}_n$ and $\mathcal{O}'_n$.

Now for the actual construction of $\mathcal{O}_n$ and $\mathcal{O}'_n$. The ontology $\mathcal{O}_n$ is given in Figure 1. We use $C \equiv D$ as an abbreviation for $C \sqsubseteq D$ and $D \sqsubseteq C$. Line (1) makes sure that $A$ cannot be true at non-root nodes of counting trees, which would enable undesired witness concepts such as $A \sqcap \exists r^{2^n}.A$. Lines (2) and (3) guarantees that Counter 1 starts with value 0 at $A$ and is incremented modulo $2^n$ when passing to an $r$-child where $\forall r.(C_1{+}{+})$ is an abbreviation for a more complex concept (of size polynomial in $n$) that achieves this; it is standard to work out details, see e.g. (Ghilardi, Lutz, and Wolter 2006). The concepts $(C_1 = 0)$, $(C_1 = 2^n - 1)$, and $(C_1 < 2^n - 1)$ are also abbreviations, with the obvious meaning. Lines (4) to (6) make sure that Counter 2 starts with value 0 in all paths that are outgoing from an instance of $A$. Lines (7) to (10) guarantee that concept name $Z_2$ is true at a domain element in a Counter 2 sequence iff there was a zero bit strictly earlier in that sequence. Lines (11)-(12) ensure that the concept name $X_3$ which represents bit values for Counter 3 is interpreted uniformly in Counter 2 sequences (which represent a single bit position of Counter 3). Line (13) guarantees that concept name $E_3$ marks exactly the final domain element on Counter 3 sequences. Lines (14)-(17) enforce that Counter 3 starts with value 0 in all paths that are outgoing from an instance of $A$. Lines (18)-(24) make sure that concept name $Z_3$ is true at a domain element in a Counter 3 sequence iff there was a zero bit strictly earlier in that sequence. Lines (25)-(26) gurantee that $L_2$ is true in the last element of every Counter 2 sequence and Lines (27)-(32) achive that $L_3$ is true in the last Counter 2 subsequence of every Counter 3 sequence. In both cases, we do not need the converse (altough it would be possible to achieve also the converse with further concept inclusions). Note that we do not use the concept names $S_L$ and $S_R$ as this does not turn out to be necessary. However, we want them to be part of $\mathrm{sig}(\mathcal{O}_1)$, which is achieved by Line (33).

We present the ontology $\mathcal{O}'_n$ in two parts, one pertaining to Counter 2 and one pertaining to Counter 3. The first part of $\mathcal{O}'_n$ can be found in Figure 2. We want to achieve that for all forest models $\mathfrak{A}$ of $\mathcal{O}_n$ of finite outdegree and all $d \in A^{\mathfrak{A}}$, there is a pointed model $\mathfrak{B}, e$ of $\mathcal{O}_n \cup \mathcal{O}'_n$ such that $\mathfrak{B}, e \sim^f_{\mathcal{ALCO},\Sigma} \mathfrak{A}, d$ if and only if one of the following holds:

1. $\mathfrak{A}$ does not contain a counting tree rooted at $d$;

$$\top \sqsubseteq \forall r.\neg A \quad (1)$$

$$A \sqsubseteq (C_1 = 0) \quad (2)$$

$$\top \sqsubseteq \forall r.(C_1{+}{+}) \quad (3)$$

$$A \sqsubseteq F_2 \quad (4)$$

$$F_2 \sqcap (C_1 < 2^n) \sqsubseteq \forall r.F_2 \quad (5)$$

$$F_2 \sqsubseteq \neg X_2 \quad (6)$$

$$(C_1 = 0) \sqsubseteq \neg Z_2 \quad (7)$$

$$\neg X_2 \sqcap \neg(C_1 = 2^n - 1) \sqsubseteq \forall r.Z_2 \quad (8)$$

$$Z_2 \sqcap \neg(C_1 = 2^n - 1) \sqsubseteq \forall r.Z_2 \quad (9)$$

$$X_2 \sqcap \neg Z_2 \sqcap \neg(C_1 = 2^n - 1) \sqsubseteq \forall r.\neg Z_2 \quad (10)$$

$$(C_1 < 2^n - 1) \sqcap X_3 \sqsubseteq \forall r.X_3 \quad (11)$$

$$(C_1 < 2^n - 1) \sqcap \neg X_3 \sqsubseteq \forall r.\neg X_3 \quad (12)$$

$$(C_1 = 2^n - 1) \sqcap \neg Z_2 \sqcap X_2 \equiv E_3 \quad (13)$$

$$A \sqsubseteq F_3 \quad (14)$$

$$F_3 \sqcap \neg(C_1 = 2^n - 1) \sqsubseteq \forall r.F_3 \quad (15)$$

$$F_3 \sqcap (C_1 = 2^n - 1) \sqcap \neg E_3 \sqsubseteq \forall r.F_3 \quad (16)$$

$$F_3 \sqsubseteq \neg X_3 \quad (17)$$

$$A \sqsubseteq \neg Z_3 \quad (18)$$

$$E_3 \sqsubseteq \forall r.\neg Z_3 \quad (19)$$

$$(C < 2^n - 1) \sqcap Z_3 \sqsubseteq \forall r.Z_3 \quad (20)$$

$$(C < 2^n - 1) \sqcap \neg Z_3 \sqsubseteq \forall r.\neg Z_3 \quad (21)$$

$$\neg X_3 \sqcap (C = 2^n - 1) \sqcap \neg E_3 \sqsubseteq \forall r.Z_3 \quad (22)$$

$$Z_3 \sqcap (C = 2^n - 1) \sqcap \neg E_3 \sqsubseteq \forall r.Z_3 \quad (23)$$

$$X_3 \sqcap \neg Z_3 \sqcap (C = 2^n - 1) \sqcap \neg E_3 \sqsubseteq \forall r.\neg Z_3 \quad (24)$$

$$E_3 \sqsubseteq L_2 \quad (25)$$

$$(C_1 < 2^n - 1) \sqcap \exists r.L_2 \sqsubseteq L_2 \quad (26)$$

$$E_3 \sqcap \neg Z_3 \sqcap X_3 \sqsubseteq L_3 \sqcap L_3' \quad (27)$$

$$(C_1 < 2^n - 1) \sqcap \exists r.L_3' \sqsubseteq L_3 \sqcap L_3' \quad (28)$$

$$(C_1 < 2^n - 1) \sqcap \exists r.L_3 \sqsubseteq L_3 \quad (29)$$

$$(C_1 < 2^n - 1) \sqcap X_2 \sqcap \exists r.L_3 \sqsubseteq L_3 \sqcap L_3' \quad (30)$$

$$(C_1 = 2^n - 1) \sqcap \exists r.L_3' \sqsubseteq L_3 \quad (31)$$

$$(C_1 = 2^n - 1) \sqcap X_2 \sqcap \exists r.L_3' \sqsubseteq L_3 \sqcap L_3' \quad (32)$$

$$S_L \sqcup S_R \sqsubseteq \top \quad (33)$$

Figure 1: The ontology $\mathcal{O}_n$.

2. Counter 2 is not properly incremented on some path in $\mathfrak{A}$ that starts at $d$;
3. Counter 3 is not properly incremented on some path in $\mathfrak{A}$ that starts at $d$.

We speak of these three options as *defects* of type 1, 2, and 3, respectively. In Line (34), we choose a defect type that is present in the current model $\mathfrak{A}$. More precisely, being able to make $P_i$ true at $e$ in $\mathfrak{B}$ corresponds to a defect of type $i$ in $\mathfrak{A}$.

Lines (35)-(36) implement defects of type 1. To see how this works, first assume that there is a counting tree in $\mathfrak{A}$ rooted at $d \in A^{\mathfrak{A}}$ and let $\mathfrak{B}, e$ be a pointed model of $\mathcal{O}_n'$ with $\mathfrak{B}, e \sim^f_{\mathcal{ALCO},\Sigma} \mathfrak{A}, d$. We have to argue that $e \notin P_1^{\mathfrak{B}}$. This is due to Lines (35) and (36) which then require the existence of a path in the counting tree to an element $f$ that has no $r$-successor that satisfies $S_L$ or no $r$-successor that satisfies $S_R$, such that the maximum value of Counter 3 is not reached on the way to $f$. But no such path exists.

For the converse, assume that for some $d \in A^{\mathfrak{A}}$, there is a no counting tree in $\mathfrak{A}$ rooted at $d$. This implies the existence of a word $w \in \{L, R\}^*$ of length strictly less than $m$ such that there is no path $p$ in $\mathfrak{A}$ starting at $d$ that follows branching pattern $L, R$ and ends in an element where Counter 3 has maximum value and that has no $r$-successor that satisfies $S_L$ or no $r$-successor that satisfies $S_R$. Let $\mathfrak{A}|_d^{\downarrow}$ denote the restriction of $\mathfrak{A}$ to the domain elements that are reachable from $d$, traveling role names only in the forward direction. Further let $\mathfrak{B}$ be obtained from $\mathfrak{A}|_d^{\downarrow}$ by making $P_1$ true on every element on path $p$. Then $\mathfrak{B}$ is a model of $\mathcal{O}_n'$ with $d \in P_1^{\mathfrak{B}}$ and $\mathfrak{B}, d \sim^f_{\mathcal{ALCO},\Sigma} \mathfrak{A}, d$.

Lines (37)-(46) verify that, if $P_2$ is chosen in Line (34), then there is indeed a defect of type 2. Here we use an auxiliary single exponential counter $D_1$ based on concept names $D_0, \ldots, D_{n-1}$. Lines (37) and (38) mark the place where incrementation of Counter 2 fails using the concept name $M_0$. Note that Line (38) ensures that $M_0$ is chosen before the last Counter 2 sequence in the last Counter 3 sequence is reached. When a domain element $d$ is marked with $M_0$, this means that it is a bit of Counter 2 such that, on some path outgoing from $d$, the corresponding bit in the subsequent Counter 2 sequence violates incrementation. There are two ways in which this may happen: first, there may be no 0-bit lower than the bit marked with $M_0$, but the corresponding bit in the subsequent Counter 2 sequence is not toggled. Second, there may be a 0-bit lower than the bit marked with $M_0$, but the corresponding bit in the subsequent Counter 2 sequence is toggled. These two cases are distinguished by Lines (40) and (41). In the first case, the value of $X_2$ is stored in $NX_2$. In the second case, the toggled value of $X_2$ is stored in $NX_2$. The counter $D_1$ is then reset in Line (39) and incremented in Line (42) to identify the corresponding bit in the following configuration. Through lines (43) and (44), the value of $NX_2$ is passed on all the way to this bit. Finally, Lines (45) and (46) ensure that the $X_2$-value of the corresponding bit coincides with $NX_2$. It is not so difficult to prove formally that this works. In particular, if there is a path starting at some $d \in A^{\mathfrak{A}}$ on which Counter 2 is not properly incremented, then we can extend $\mathfrak{A}|_d^{\downarrow}$ in a straightforward way to a model $\mathfrak{B}$ of $\mathcal{O}_n'$ with $d \in P_2^{\mathfrak{B}}$, by interpreting the concept names in $\mathrm{sig}(\mathcal{O}_n') \setminus \mathrm{sig}(\mathcal{O}_n)$.

The part of $\mathcal{O}_n'$ that is concerned with Counter 3 is displayed in Figure 3. It makes sure that if $P_2$ is chosen in Line (34), then there is indeed a defect of type 3. It is here that using a nominal is crucial. Lines (47) and (48) mark

$$A \sqsubseteq P_1 \sqcup P_2 \sqcup P_3 \tag{34}$$

$$P_1 \sqcap E_3 \sqcap \neg Z_3 \sqcap X_3 \sqsubseteq \forall r.(S_L \to P_1) \sqcup \forall r.(S_R \to P_1) \tag{35}$$

$$P_1 \sqcap E_3 \sqcap \neg Z_3 \sqcap X_3 \sqsubseteq \bot \tag{36}$$

$$P_2 \sqsubseteq M_0 \sqcup \exists r.((S_L \sqcup S_R) \sqcap P_2) \tag{37}$$

$$P_2 \sqsubseteq \neg(L_2 \sqcap L_3) \tag{38}$$

$$M_0 \sqsubseteq M \sqcap (D_1 = 0) \tag{39}$$

$$M_0 \sqcap \neg Z_2 \sqsubseteq NX_2 \leftrightarrow X_2 \tag{40}$$

$$M_0 \sqcap Z_2 \sqsubseteq NX_2 \leftrightarrow \neg X_2 \tag{41}$$

$$M \sqcap (D_1 < 2^n - 1) \sqsubseteq \forall r.(D_1{+}{+}) \tag{42}$$

$$M \sqcap (D_1 < 2^n - 1) \sqcap NX_2 \sqsubseteq \exists r.((S_L \sqcup S_R) \sqcap M \sqcap NX_2) \tag{43}$$

$$M \sqcap (D_1 < 2^n - 1) \sqcap \neg NX_2 \sqsubseteq \exists r.((S_L \sqcup S_R) \sqcap M \sqcap \neg NX_2) \tag{44}$$

$$M \sqcap (D_1 = 2^n - 1) \sqcap NX_2 \sqsubseteq X_2 \tag{45}$$

$$M \sqcap (D_1 = 2^n - 1) \sqcap \neg NX_2 \sqsubseteq \neg X_2 \tag{46}$$

Figure 2: The first part of ontology $\mathcal{O}'_n$.

the place where incrementation of Counter 3 fails using the concept name $M_0$. Note that Line (48) ensures that $M_0$ is chosen in some Counter 3 sequence that is not the final one; we refer to it as the 'current' Counter 3 sequence. Line (49) further makes sure that the element chosen by $M_0$ is at the beginning of a Counter 2 sequence, which we refer to as the 'current' Counter 2 sequence. It also chooses via the concept names $M_L$ and $M_R$ whether the defect occurs in the subsequent Counter 3 sequence that is a left child of the current sequence, or a right child. The chosen value is memorized 'forever' in Line (50). Our aim is to set another marker at the beginning of a Counter 2 sequence in a subsequent Counter 3 sequence that encodes the same Counter 2 value as the current Counter 2 sequence, and then to compare the two $X_3$-bit values of the two Counter 2 sequences.

To achieve this, we need to memorize for later comparison *all* (exponentially many) $X_2$-bit values of the current Counter 2 sequence. This cannot be done in a single type and thus we use multiple types. This is implemented in Lines (51)-(55) in which the current Counter 2 sequence is traversed from beginning to end. In each step, a branching takes place via Line (52). It is important to understand that this branching is in model $\mathfrak{B}$, but not necessarily in model $\mathfrak{A}$. Recall that we are interested in models $\mathfrak{B}$ that have a functional $\mathcal{ALCO}, \Sigma$-bisimulation to $\mathfrak{A}$. Informally, we can assume the two models to have the same domain and $r$-structure up to the element $d$ in $\mathfrak{A}$ in which we have chosen to set the marker $M_0$.[2] When setting $M_0$, then we 'are' in an element $e$ of $\mathfrak{B}$ that is $\Sigma$-bisimilar to $d$. This and what follows is illustrated in Figure 4. Now Line (52) creates two $r$-successors $f_1$ and $f_2$ of $e$ in $\mathfrak{B}$ that are both $\Sigma$-bisimilar to $r$-successors of $d$ in $\mathfrak{A}$. We shall argue a bit later that this must actually be the same $r$-successor of $d$. In branch $f_1$ of $\mathfrak{B}$, we stay with marker $M_1$ while in branch $f_2$ of $\mathfrak{B}$, we switch to marker $M_2$. The $M_1$-branch branches again at the next point of the Counter 2 sequence while the $M_2$ path does not, and so on. Via Line (51), at each point of the Counter 2 sequence we memorize in the newly generated $M_1$-branch the value of Counter 1 in the auxiliary counter $D_1$, the value of $X_2$ in $NX_2$, and the value of $X_3$ in $NX_3$. In contrast, the $M_2$-branches retain their memory via Lines (54)-(55). At the end of the Counter 2 sequence whose beginning is marked with $M_0$, we have thus generated $2^n$ branches in $\mathfrak{B}$, each storing the $X_2$-bit value for one bit position of Counter 2, and all of them storing the $X_3$-bit value of the current Counter 2 sequence.

Via Line (53), all the $M_2$-branches extend also beyond the current Counter 2 sequence to the end of the current Counter 3 sequence. In Lines (56) and (57), we make a step to the first element of a subsequent Counter 3 sequence, switching to marker $M_3$. *All* $M_2$-branches in $\mathfrak{B}$ decide to go to an $S_L$-labeled such subsequent sequence or all decide to go to an $S_R$-labeled such sequence, depending on whether we had initially (before the branching) chosen marker $M_L$ or $M_R$. Via Line (58), we proceed down the Counter 3 sequence and set the $M_4$-marker before reaching its end. What we want to achieve is that the $M_4$-marker is set at the Counter 2 subsequence that carries the same Counter 2 value as the Counter 2 sequence, at the Counter 2 bit position that the current branch has stored in counter $D_1$. We verify independently for each branch in $\mathfrak{B}$ that the bit position is correct, in Lines (59)-(60), and there we also make sure that the $X_2$-bit value coincides with the $X_2$-bit value stored in $NX_2$. We also use Line (60) to make sure that there is indeed an incrementation conflict of Counter 3 at this position.

We are done if we can additionally guarantee that the different branches in $\mathfrak{B}$ have really set the $M_4$-marker at the same Counter 2 subsequence in the same path of $\mathfrak{A}$. So far, however, we do not not know that this is the case, nor that the different branches have even followed the same path of

[2] Please compare this to defects of type 1 and 2 where $\mathfrak{B}$ can be assumed to have the same domain and $r$-structure as $\mathfrak{A}$; this is also the case here, up to the $M_0$ marker, but not beyond.

$$
\begin{aligned}
P_3 &\sqsubseteq M_0 \sqcup \exists r.((S_L \sqcup S_R) \sqcap P_3) && (47)\\
P_3 &\sqsubseteq \neg L_3 && (48)\\
M_0 &\sqsubseteq (C_1 = 0) \sqcap M_1 \sqcap (M_L \sqcup M_R) && (49)\\
K &\sqsubseteq \forall r.K \qquad \text{for all } K \in \{M_L, M_R\} && (50)\\
M_1 &\sqsubseteq (D_1 = C_1) \sqcap (NX_2 \leftrightarrow X_2) \sqcap (NX_3 \leftrightarrow X_3) && (51)\\
M_1 \sqcap (C_1 < 2^n - 1) &\sqsubseteq \exists r.((S_L \sqcup S_R) \sqcap M_1) \sqcap \exists r.((S_L \sqcup S_R) \sqcap M_2) && (52)\\
M_2 \sqcap \neg E_3 &\sqsubseteq \exists r.((S_L \sqcup S_R) \sqcap M_2) && (53)\\
M_i \sqcap K &\sqsubseteq \forall r.(M_i \rightarrow K) \quad \text{for all } i \in \{2,3,4\} \text{ and} && (54)\\
&\qquad K \in \{C, \neg C \mid C \in \{D_0, \ldots, D_{n-1}, NX_2, NX_3\}\} && (55)\\
M_2 \sqcap M_L \sqcap E_3 &\sqsubseteq \exists r.(S_L \sqcap M_3) && (56)\\
M_2 \sqcap M_R \sqcap E_3 &\sqsubseteq \exists r.(S_R \sqcap M_3) && (57)\\
M_3 &\sqsubseteq \neg E_3 \sqcap (M_4 \sqcup \exists r.((S_L \cup S_R) \sqcap M_3) && (58)\\
M_4 &\sqsubseteq (C_1 = D_1) \sqcap M_5 \sqcap (X_2 \leftrightarrow NX_2) \sqcap && (59)\\
&\qquad (Z_3 \sqcap (X_3 \leftrightarrow \neg NX_3)) \sqcup (\neg Z_3 \sqcap (X_3 \leftrightarrow NX_3)) && (60)\\
M_5 \sqcap (C_1 < 2^n - 1) &\sqsubseteq \exists r.((S_L \sqcup S_R) \sqcap M_5) && (61)\\
M_5 \sqcap (C_1 = 2^n - 1) &\sqsubseteq \{c\} && (62)
\end{aligned}
$$

Figure 3: The second part of ontology $\mathcal{O}'_n$.

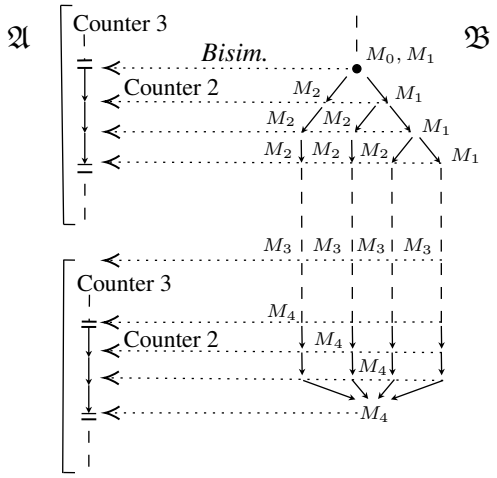



Figure 4: Strategy for comparing bit values of Counter 3

$\mathfrak{A}$. But this is now easily rectified: Lines (61) and (62) force all branches of $\mathfrak{B}$ to further follow the current Counter 2 sequence, until it's end, and that the individual name $c$ is satisfied at the end of all branches. Thus the end of all branches is the same element in $\mathfrak{B}$, which is functionally bisimilar to some element of $\mathfrak{A}$. But $\mathfrak{A}$ is a forest model and, as $\mathcal{O}_1$ does not use nominals, it is (by definition) even a tree model. Consequently, on every branch of $\mathfrak{B}$, the $r$-predecessor of the final element marked with $c$ is functionally bisimilar to the same element of $\mathfrak{A}$, and so on, all the way up to the element of $\mathfrak{B}$ where the $M_0$-marker was set.

Based on what was said above, it can be verified that Lemma 21 indeed holds. We refrain from giving details.

## F.2 3ExpTime-Hardness

An *Alternating Turing Machine (ATM)* is of the form $\mathcal{M} = (Q, \Sigma, \Gamma, q_0, \Delta)$. The set of *states* $Q = Q_\exists \uplus Q_\forall \uplus \{q_a\} \uplus \{q_r\}$ consists of *existential states* from $Q_\exists$, *universal states* from $Q_\forall$, an *accepting state* $q_a$, and a *rejecting state* $q_r$; $\Sigma$ is the *input alphabet* and $\Gamma$ the *work alphabet* containing a *blank symbol* $\square$ and satisfying $\Sigma \subseteq \Gamma$; $q_0 \in Q_\exists$ is the *starting* state; and the *transition relation* $\delta$ is of the form

$$\delta \subseteq Q \times \Gamma \times Q \times \Gamma \times \{L, R\}.$$

We write $\delta(q, a)$ for $\{(q', b, M) \mid (q, a, q', b, M) \in \delta\}$. As usual, we assume that $q \in Q_\exists \cup Q_\forall$ implies $\delta(q, b) \neq \emptyset$ for all $b \in \Gamma$ and $q \in \{q_a, q_r\}$ implies $\delta(q, b) = \emptyset$ for all $b \in \Gamma$. For what follows, we also assume w.l.o.g. that for each $q \in Q_\forall \cup Q_\exists$ and each $b \in \Sigma$, the set $\delta(q, b)$ has exactly two elements. We assume for notational convenience that these elements are ordered, i.e., $\delta(q, b)$ is an ordered pair $((q', b', M'), (a'', b'', M''))$.

A *configuration* of an ATM is a word $wqw'$ with $w, w' \in \Gamma^*$ and $q \in Q$. The intended meaning is that the tape contains the word $ww'$ (with only blanks before and behind it), the machine is in state $q$, and the head is on the leftmost symbol of $w'$. The *successor configurations* of a configuration $wqw'$ are defined in the usual way in terms of the transition relation $\delta$. A *halting configuration* is of the form $wqw'$ with $q \in \{q_a, q_r\}$.

A *computation path* of an ATM $\mathcal{M}$ on a word $w$ is a (finite or infinite) sequence of configurations $c_1, c_2, \ldots$ such

that $c_1 = q_0 w$ and $c_{i+1}$ is a successor configuration of $c_i$ for $i \geq 0$. All ATMs considered in this paper have only *finite* computation paths on any input. A halting configuration is *accepting* iff it is of the form $wq_a w'$. For non-halting configurations $c = wqw'$, the acceptance behaviour depends on $q$: if $q \in Q_\exists$, then $c$ is accepting iff at least one successor configuration is accepting; if $q \in Q_\forall$, then $c$ is accepting iff all successor configurations are accepting. Finally, the ATM $\mathcal{M}$ with starting state $q_0$ *accepts* the input $w$ iff the *initial configuration* $q_0 w$ is accepting. We use $L(\mathcal{M})$ to denote the language accepted by $\mathcal{M}$, i.e., $L(\mathcal{M}) = \{w \in \Sigma^* \mid \mathcal{M} \text{ accepts } w\}$.

To obtain a witness for the acceptance of an input by an ATM, it is common to arrange configurations in a tree. Such an *acceptance tree of an ATM* $\mathcal{M}$ with starting state $q_0$ *on a word* $w$ is a finite tree whose nodes are labelled with configurations such that

- the root node is labelled with the initial configuration $q_0 w$;
- if a node $s$ in the tree is labelled with $wqw'$, $q \in Q_\forall$, then $s$ has exactly two successors, labeled with the two successor configurations of $wqw'$;
- if a node $s$ in the tree is labelled with $wqw'$, $q \in Q_\exists$, then $s$ has exactly two successors, both labelled with a successor configuration of $wqw'$;[3]
- leaves are labelled with accepting halting configurations.

It is clear that there exists an acceptance tree of $\mathcal{M}$ on $w$ if and only if $\mathcal{M}$ accepts $w$.

According to Theorem 3.4 of (Chandra, Kozen, and Stockmeyer 1981), there is a double exponentially space bounded ATM $\mathcal{M}$ whose word problem is 3EXPTIME-hard. We may w.l.o.g. assume that the length of every computation path of $\mathcal{M}$ on any input $w \in \Sigma^n$ is bounded by $2^{2^{2^n}}$, and all the configurations $wqw'$ in such computation paths satisfy $|ww'| \leq 2^{2^n}$.

We prove Theorem 14 by reduction from the word problem for $\mathcal{M}$. Thus let $w \in \Sigma^*$ be an input to $\mathcal{M}$. We have to construct an $\mathcal{ALC}$-ontology $\mathcal{O}$ and an $\mathcal{ALCO}$-ontology $\mathcal{O}'$ such that $\mathcal{O} \cup \mathcal{O}'$ is a conservative extension of $\mathcal{O}$ if and only if $M$ does not accept $w$. This can be achieved by extending the ontologies $\mathcal{O}_n$ and $\mathcal{O}'_n$ from Section F.1, where $n$ is the length of $w$. The main idea is to do this such that models $\mathfrak{A}$ of witness concepts for $\mathcal{O}$ and $\mathcal{O}'$ describe an acceptance tree of $\mathcal{M}$ on $w$ instead of a counting tree. In fact, such models $\mathfrak{A}$ describe a tree that is a counting tree and at the same time a computation tree. We again use only a single role name $r$. Each configuration of $\mathcal{M}$ is represented by a Counter 3 sequence and each tape cell of a configuration is represented by a Counter 2 sequence. Thus, each node of the acceptance tree is spread out over $2^n \cdot 2^{2^n}$ elements in the model and the role name $r$ might indicate moving to the next tape cell in the same configuration, moving to the first tape cell of a successor configuration, and also moving to the next element in the Counter 2 sequence that represents the current tape cell.

[3] A single successor would of course be sufficient; we only use two successors (which can carry the same label) to enable a more uniform reduction.

So we only need to represent the computation of $\mathcal{M}$ on $w$ on top of a counting tree that we have already enforced by $\mathcal{O}_n$ and $\mathcal{O}'_n$, and we can exploit the three counters that $\mathcal{O}_n$ and $\mathcal{O}'_n$ give us. Such a representation is in fact fairly standard and no additional technical tricks are needed, see for instance the 2EXPTIME-hardness proof for conservative extensions in $\mathcal{ALC}$ given in (Ghilardi, Lutz, and Wolter 2006). Since fully worked out concept inclusions are nevertheless lengthy and difficult to comprehend, we only sketch the required extensions of $\mathcal{O}_n$ and $\mathcal{O}'_n$.

In the extension $\mathcal{O}$ of $\mathcal{O}_n$, we use an additional concept name $S_q$ for every state $q \in Q$, $S_a$ for every symbol $a \in \Sigma$, and $H$ for indicating the head position. The ontology $\mathcal{O}$ then makes sure that all these symbols are interpreted uniformly in Counter 2 sequences, that exactly one concept name $S_a$ is true at every Counter 2 sequence, and that there is exactly one Counter 2 sequence in each Counter 3 sequence where $H$ is true, together with a unique concept name $S_q$. It also guarantees that the computation starts with the initial configuration $q_0 w$. All remaining properties of computation trees are achieved by $\mathcal{O}'$. In preparation for this, we add more concept names to $\mathcal{O}$, namely primed versions $S'_q$, $S'_q$, and $H'$ of all concept names introduced above as well as concept names $Y_{q,a,M}$ and $Y_{q,a,M}$ for all $q \in Q$, $a \in \Sigma$, and $M \in \{L, R\}$. Informally, a concept name $Y_{q,a,M}$ indicates that for moving to the current configuration, the Turing machine has decided to write symbol $a$, switch to state $q$, and move in direction $M$. Still in $\mathcal{O}$, we make sure that the primed concept names satisfy the same constraints as their unprimed siblings, that the concept names $Y_{q,a,M}$ are set in successor configurations (Counter 3 sequences) in accordance with the unprimed concept names and the transition relation, and that if some $Y_{q,a,M}$ is set in the current configuration, then the non-primed and the primed concept names relate to each other accordingly.

It remains for $\mathcal{O}'$ to make sure that the transition relation of $\mathcal{M}$ is respected and that the the rejecting state is not reached on any branch. Given what was already done in $\mathcal{O}$, the former can be achieved by enforcing that whenever some unprimed concept name $S_q$, $S_a$, or $H$ is true in some Counter 2 sequence, then its primed version is true in the Counter 2 sequence that represents the same Counter 2 value of all subsequent Counter 3 sequences. We refer to this as *correct copying*.

Strictly speaking, the second ontology $\mathcal{O}'$ is not an extension of $\mathcal{O}$ because we need to replace Line (34) with the following, which admits five different types of defects in place of two:

$$A \sqsubseteq P_1 \sqcup P_2 \sqcup P_3 \sqcup P_4 \sqcup P_5.$$

$P_4$ is for checking that some branch reaches the rejecting state. This is easy to implement, using existential restrictions as in Lines (37)-(46) (rather than universal restrictions as in Lines (35) and (36)). $P_5$ is for verifying that correct copying is taking place. This is achieved by a slight variation of the concept inclusions in Figure 3 (which also use existential restrictions). In fact, those concept inclusions can be viewed as copying the value of an $X_3$-bit to same-value Counter 2 subsequences of subsequent Counter 3 sequences.

We copy the information stored in the concept names $S_q$, $S_a$, and $H$ instead.

## G Proofs for Section 6

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

1. *$(\mathcal{K}, \{a\}, \{b\})$ is strongly FO($\Sigma$)-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.*

**Proof.** 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 $u$. To see this, obtain $C_{\mathcal{K},\Sigma,a}$ from $\mathcal{K}$ by taking the conjunction of the following concepts:

- $\forall u.(C \to D)$, for $C \sqsubseteq D \in \mathcal{O}$;
- $\forall u.(\{c\} \to \exists R.\{d\})$, for $R(c,d) \in \mathcal{D}$;
- $\forall u.(\{c\} \to A)$, for $A(c) \in \mathcal{D}$;

and then replacing

- all concept and role names $X$ not in $\Sigma$ by fresh and distinct symbols $X_a$;
- all individual names $c$ not in $\Sigma \cup \{a\}$ by fresh and distinct individual names $c_a$;
- the individual name $a$ by a fresh individual name $m$;
- if $a \in \Sigma$ then add $\{m\} \leftrightarrow \{a\}$ as a conjunct.

Define $C_{\mathcal{K},\Sigma,b}$ in the same way with $a$ replaced by $b$. Then $\varphi_{\mathcal{K},\Sigma,a}(x) \wedge \varphi_{\mathcal{K},\Sigma,b}(x)$ is satisfiable if $m \wedge C_{\mathcal{K},\Sigma,a} \wedge C_{\mathcal{K},\Sigma,b}$ is satisfiable. ❑

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

**Proof.** "2. $\Rightarrow$ 1." is trivial. For the converse direction, assume that Condition 1. holds.

Note that we can view $\mathcal{K}_{\Sigma,a}$ as the union of $\mathcal{O}_a$ and $\mathcal{D}_a$, where

- $\mathcal{O}_a$ is a copy of $\mathcal{O}$ in which all concept and role names $X \notin \Sigma$ have been replaced by fresh symbols $X_a$;
- $\mathcal{D}_a$ is a copy of $\mathcal{D}$ in which every concept and role name $X \notin \Sigma$ is replaced by $X_a$ and in which every individual $c \notin \Sigma \cup \{a\}$ is replaced by a variable $x_{c,a}$ and $a$ is replaced by $x$. Moreover, $x = a$ is added if $a \in \Sigma$.

Thus, by taking the conjunction of all members of $\mathcal{D}_a$ and existentially quantifying over all variables distinct from $x$ we obtain a formula in CQ$^{\mathcal{ALCI}}$. $\mathcal{K}_{\Sigma,b}$ can be viewed accordingly with $a$ replaced by $b$.

In what follows we write

- $\mathfrak{A}, d \Leftrightarrow_{\mathrm{CQ}^{\mathcal{ALCIO}},\Sigma} \mathfrak{B}, e$ if $\mathfrak{A} \models \varphi(d)$ iff $\mathfrak{B} \models \varphi(e)$, for all $\varphi(x)$ in CQ$^{\mathcal{ALCIO}}(\Sigma)$.
- $\mathfrak{A}, d \Leftrightarrow^{\mathrm{mod}}_{\mathrm{CQ}^{\mathcal{ALCIO}},\Sigma} \mathfrak{B}, e$ if for all finite $D \subseteq \mathrm{dom}(\mathfrak{A})$ containing $d$ we have $\mathfrak{A}, d \to_{D,\mathcal{ALCIO},\Sigma} \mathfrak{B}, e$, and vice versa.

By Condition 1, we have $\varphi_{\mathcal{K},\Sigma,a}(x) \models \neg\varphi_{\mathcal{K},\Sigma,b}(x)$. Assume there does not exist a separating formula in BoCQ$^{\mathcal{ALCIO}}(\Sigma)$. We first show the following claim.

*Claim 1.* There exist pointed structures $\mathfrak{A}, d$ and $\mathfrak{B}, e$ such that $\mathfrak{A} \models \varphi_{\mathcal{K},\Sigma,a}(d)$ and $\mathfrak{B} \models \varphi_{\mathcal{K},\Sigma,b}(e)$ and $\mathfrak{A}, d \Leftrightarrow_{\mathrm{CQ}^{\mathcal{ALCIO}},\Sigma} \mathfrak{B}, e$.

For the proof of Claim 1, consider the set $\Gamma$ of all formulas $\psi(x)$ in BoCQ$^{\mathcal{ALCIO}}(\Sigma)$ such that $\varphi_{\mathcal{K},\Sigma,a}(x) \models \psi(x)$. By compactness and our assumption,

$$\Gamma \cup \{\varphi_{\mathcal{K},\Sigma,b}(x)\}$$

is satisfiable. Take a pointed model $\mathfrak{B}, e$ of $\Gamma \cup \{\varphi_{\mathcal{K},\Sigma,b}(x)\}$. Next, let $\Psi$ be the set of all $\psi(x)$ in BoCQ$^{\mathcal{ALCIO}}(\Sigma)$ such that $\mathfrak{B} \models \psi(e)$. By compactness and assumption

$$\Psi \cup \{\varphi_{\mathcal{K},\Sigma,a}(x)\}$$

is satisfiable. Take a pointed model $\mathfrak{A}, d$ of $\Psi \cup \{\varphi_{\mathcal{K},\Sigma,a}(x)\}$. By definition, the pointed models $\mathfrak{A}, d$ and $\mathfrak{B}, e$ are as required in Claim 1.

Using $\omega$-saturated elementary extensions of the pointed models $\mathfrak{A}, d$ and $\mathfrak{B}, e$ from Claim 1 we obtain the following claim using Lemma 3.

*Claim 2.* There exist pointed structures $\mathfrak{A}, d$ and $\mathfrak{B}, e$ such that $\mathfrak{A} \models \varphi_{\mathcal{K},\Sigma,a}(d)$ and $\mathfrak{B} \models \varphi_{\mathcal{K},\Sigma,b}(e)$ and such that $\mathfrak{A}, d \Leftrightarrow^{\mathrm{mod}}_{\mathrm{CQ}^{\mathcal{ALCI}},\Sigma} \mathfrak{B}, e$.

Now take assignments $v_a$ from the variables of $\varphi_{\mathcal{K},\Sigma,a}$ into $\mathfrak{A}$ witnessing $\mathfrak{A} \models \varphi_{\mathcal{K},\Sigma,a}(d)$ and $v_b$ from the variables of $\varphi_{\mathcal{K},\Sigma,b}$ into $\mathfrak{B}$ witnessing $\mathfrak{B} \models \varphi_{\mathcal{K},\Sigma,b}(e)$. Let $D_a$ and $D_b$ be the images of $v_a$ in $\mathfrak{A}$ and of $v_b$ in $\mathfrak{B}$, respectively. By definition, we have $\Sigma$-homomorphisms

- $h_a : \mathfrak{A}_{|D_a} \to \mathfrak{B}$ mapping $d$ to $e$ and such that $\mathfrak{A}, c \sim_{\mathcal{ALCIO},\Sigma} \mathfrak{B}, h_a(c)$ for all $c \in D$;
- $h_b : \mathfrak{B}_{|D_b} \to \mathfrak{A}$ mapping $e$ to $d$ and such that $\mathfrak{B}, c \sim_{\mathcal{ALCIO},\Sigma} \mathfrak{A}, h_b(c)$ for all $c \in D_b$.

We also have by definition that for any $c \in \mathrm{dom}(\mathfrak{A})$ there exists a $c' \in \mathrm{dom}(\mathfrak{B})$ such that $\mathfrak{A}, c \sim_{\mathcal{ALCIO},\Sigma} \mathfrak{B}, c'$, and vice versa. We now merge $\mathfrak{A}$ and $\mathfrak{B}$ to a single structure by taking their bisimulation product $\mathfrak{C}$, defined as follows. The domain of $\mathfrak{C}$ is

$$\{(c, c') \in \mathrm{dom}(\mathfrak{A}) \times \mathrm{dom}(\mathfrak{B}) \mid \mathfrak{A}, c \sim_{\mathcal{ALCI},\Sigma} \mathfrak{B}, c'\}$$

and we set

- $(c, c') \in A^{\mathfrak{C}}$ if $c \in A^{\mathfrak{A}}$ (equivalently, if $c' \in A^{\mathfrak{B}}$) for all $A \in \Sigma$;
- $(c, c') \in A^{\mathfrak{C}}$ if $c \in A^{\mathfrak{A}}$ for all $A \in \mathrm{sig}(\varphi_{\mathcal{K},\Sigma,a}) \setminus \Sigma$;
- $(c, c') \in A^{\mathfrak{C}}$ if $c' \in A^{\mathfrak{B}}$ for all $A \in \mathrm{sig}(\varphi_{\mathcal{K},\Sigma,b}) \setminus \Sigma$;
- $((c_1, c_1'), (c_2, c_2')) \in r^{\mathfrak{C}}$ if $(c_1, c_2) \in r^{\mathfrak{A}}$ and $(c_1', c_2') \in r^{\mathfrak{B}}$ for all $r \in \Sigma$;

- $((c_1, c_1'), (c_2, c_2')) \in r^{\mathfrak{C}}$ if $(c_1, c_2) \in r^{\mathfrak{A}}$ for all $r \in \text{sig}(\varphi_{\mathcal{K},\Sigma,a}) \setminus \Sigma$;
- $((c_1, c_1'), (c_2, c_2')) \in r^{\mathfrak{C}}$ if $(c_1', c_2') \in r^{\mathfrak{B}}$ for all $r \in \text{sig}(\varphi_{\mathcal{K},\Sigma,b}) \setminus \Sigma$;
- $c^{\mathfrak{C}} = (c^{\mathfrak{A}}, c^{\mathfrak{B}})$ for all $c \in \Sigma$.

We show that $\mathfrak{C} \models (\varphi_{\mathcal{K},\Sigma,a} \wedge \varphi_{\mathcal{K},\Sigma,b})(d, e)$ which contradicts the assumption that $\varphi_{\mathcal{K},\Sigma,a}(x) \models \neg\varphi_{\mathcal{K},\Sigma,b}(x)$. To show that $\mathfrak{C}$ is a model of $\mathcal{O}_a$ and $\mathcal{O}_b$ it suffices to show the following claim.

*Claim 3.* (1) The projection $p_a : \mathfrak{C} \to \mathfrak{A}$ defined by setting $p_a(c, c') = c$ is an $\mathcal{ALCIO}(\text{sig}(\varphi_{\mathcal{K},\Sigma,a}))$-bisimulation between $\mathfrak{C}$ and $\mathfrak{A}$.

(2) The projection $p_b : \mathfrak{C} \to \mathfrak{B}$ defined by setting $p_b(c, c') = c'$ is an $\mathcal{ALCIO}(\text{sig}(\varphi_{\mathcal{K},\Sigma,b}))$-bisimulation between $\mathfrak{C}$ and $\mathfrak{B}$.

The proof of Claim 3 is straightforward and omitted. It follows from Claim 3 and the assumption that $\mathfrak{A}$ is a model of $\mathcal{O}_a$ and $\mathfrak{B}$ a model of $\mathcal{O}_b$ that $\mathfrak{C}$ is a model of $\mathcal{O}_a \cup \mathcal{O}_b$.

Next we lift the variable assignments $v_a$ and $v_b$ from $\mathfrak{A}$ and, respectively, $\mathfrak{B}$ to $\mathfrak{C}$. Thus, we set

- $\bar{v}_a(x_c) = (v_a(x_c), h_a(v_a(x_c))$ for all variables of the form $x_c$ in $\varphi_{\mathcal{K},\Sigma,a}$ and
- $\bar{v}_b(y_c) = (v_b(y_c), h_b(v_b(y_c))$ for all variables of the form $y_c$ in $\varphi_{\mathcal{K},\Sigma,b}$.
- $\bar{v}_a(x) = \bar{v}_b(x) = (v_a(x), v_b(x))$.

The following claim is straightforward now.

*Claim 4.* $\mathfrak{C} \models_{\bar{v}_a} \mathcal{D}_a(d, e)$ and $\mathfrak{C} \models_{\bar{v}_b} \mathcal{D}_b(d, e)$.

Claim 4 implies $\mathfrak{C} \models (\varphi_{\mathcal{K},\Sigma,a} \wedge \varphi_{\mathcal{K},\Sigma,b})(d, e)$ as we have established already that $\mathfrak{C}$ is a model of $\mathcal{O}_a \cup \mathcal{O}_b$. This concludes the proof. ❑

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

We first prove the upper bounds. To this end, we formulate the complexity results proved in (Artale et al. 2021) for interpolant existence in detail. Let $\mathcal{L} \in \mathsf{DL_{ni}}$. Let $\mathcal{O}_1, \mathcal{O}_2$ be $\mathcal{L}$-ontologies and let $C_1, C_2$ be $\mathcal{L}$-concepts. Let $\Sigma = \text{sig}(\mathcal{O}_1, C_1) \cap \text{sig}(\mathcal{O}_2, C_2)$. An $\mathcal{L}$-interpolant for the *$\mathcal{L}$-tuple* $\mathcal{O}_1, \mathcal{O}_2, C_1, C_2$ is an $\mathcal{L}(\Sigma)$-concept $C$ such that

- $\mathcal{O}_1 \models C_1 \sqsubseteq C$;
- $\mathcal{O}_2 \models C \sqsubseteq C_2$.

The following is shown in (Artale et al. 2021).

**Theorem 16** *Let $\mathcal{L} \in \mathsf{DL_{ni}}$. Then the problem to decide whether an $\mathcal{L}$-interpolant exists for $\mathcal{L}$-tuples $\mathcal{O}_1, \mathcal{O}_2, C_1, C_2$ is* 2ExpTime*-complete.*

Now assume that $\mathcal{L} \in \mathsf{DL_{ni}}$ and a labeled $\mathcal{L}$-KB $(\mathcal{K}, \{a\}, \{b\})$ with $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ and $\Sigma \subseteq \text{sig}(\mathcal{K})$ are given. Let $\mathcal{LO}$ denote the extension of $\mathcal{L}$ by nominals (if $\mathcal{L}$ contains nominals already then set $\mathcal{LO} = \mathcal{L}$). Obtain an $\mathcal{LO}$-ontology $\mathcal{O}_{\mathcal{K},\Sigma,a}$ from $\mathcal{K}$ by taking the following inclusions:

- all inclusions in $\mathcal{O}$;
- $\{c\} \sqsubseteq \exists R.\{d\}$, for $R(c, d) \in \mathcal{D}$;
- $\{c\} \sqsubseteq A$, for $A(c) \in \mathcal{D}$;

and then replacing

- all concept and role names $X$ not in $\Sigma$ by a fresh symbol $X_a$;
- all individuals $c$ not in $\Sigma \cup \{a\}$ by fresh and distinct individuals $c_a$;
- the individual $a$ by a fresh individual $m_a$. If $a \in \Sigma$ then the CI $\{m_a\} \equiv \{a\}$ is added.

$\mathcal{O}_{\mathcal{K},\Sigma,b}$ is obtained from $\mathcal{K}$ in the same way by replacing $a$ by $b$. Observe that an $\mathcal{LO}(\Sigma)$-concept strongly separates $(\mathcal{K}, \{a\}, \{b\})$ iff it is an $\mathcal{LO}$-interpolant for the $\mathcal{LO}$-tuple $\mathcal{O}_{\mathcal{K},\Sigma,a}$, $\mathcal{O}_{\mathcal{K},\Sigma,b}$, $m_a, \neg m_b$. If $\mathcal{L}$ contains nominals, then the upper bounds stated in Theorem 10 follow immediately. If $\mathcal{L}$ does not contain nominals, then we may assume that $\Sigma$ does not contain individual names. Then $\mathcal{O}_{\mathcal{K},\Sigma,a}$ and $\mathcal{O}_{\mathcal{K},\Sigma,b}$ do not share any individual names and therefore an $\mathcal{L}(\Sigma)$-concept strongly separates $(\mathcal{K}, \{a\}, \{b\})$ iff it is an $\mathcal{LO}$-interpolant for the $\mathcal{LO}$-tuple $\mathcal{O}_{\mathcal{K},\Sigma,a}, \mathcal{O}_{\mathcal{K},\Sigma,b}, m_a, \neg m_b$. Thus, the upper bound follows again.

Now we come to the lower bounds. We first give a model-theoretic characterization of strong $\mathcal{L}$-separability using $\mathcal{L}$-bisimulations.

**Lemma 22** *Let $\mathcal{L} \in \mathsf{DL_{ni}}$. Let $(\mathcal{K}, \{a\}, \{b\})$ be a labeled $\mathcal{L}$-KB and $\Sigma \subseteq sig(\mathcal{K})$ a signature. Then the following conditions are equivalent:*

1. *$(\mathcal{K}, \{a\}, \{b\})$ is strongly $\mathcal{L}(\Sigma)$-separable;*
2. *There are no models $\mathfrak{A}$ and $\mathfrak{B}$ of $\mathcal{K}$ such that $\mathfrak{A}, a^{\mathfrak{A}} \sim_{\mathcal{L},\Sigma} \mathfrak{B}, b^{\mathfrak{B}}$.*

The proof is straightforward using Lemma 2. The following result is shown as part of the lower bound proof for interpolant existence in (Artale et al. 2021):

**Theorem 17** *Let $\mathcal{L} \in \mathsf{DL_{ni}}$. For $\mathcal{L}$-ontologies $\mathcal{O}$ and database $\mathcal{D}$ of the form $\{R(a, a)\}$ it is* 2ExpTime*-hard to decide the following: do there exist models $\mathfrak{A}$ and $\mathfrak{B}$ of $(\mathcal{O}, \mathcal{D})$ such that $\mathfrak{A}, a^{\mathfrak{A}} \sim_{\mathcal{L},\Sigma} \mathfrak{B}, d$ for some $d \neq a^{\mathfrak{B}}$.*

2ExpTime-hardness of Condition 2 of Lemma 22 is a direct consequence of Theorem 17. For suppose that $\mathcal{L}$, $\mathcal{K} = (\mathcal{O}, \mathcal{D})$, $\Sigma$, and $a$ from Theorem 17 are given. Let $b$ be a fresh individual name and $A_1, A_2$ fresh concept names. Add $A_1(a)$ and $A_2(b)$ to $\mathcal{D}$ to obtain $\mathcal{D}'$ and add $A_1 \sqsubseteq \neg A_2$ to $\mathcal{O}$ to obtain $\mathcal{O}'$. Then exist models $\mathfrak{A}$ and $\mathfrak{B}$ of $(\mathcal{O}, \mathcal{D})$ such that $\mathfrak{A}, a^{\mathfrak{A}} \sim_{\mathcal{L},\Sigma} \mathfrak{B}, d$ for some $d \neq a^{\mathfrak{B}}$ iff there exist models $\mathfrak{A}$ and $\mathfrak{B}$ of $\mathcal{K}$ such that $\mathfrak{A}, a^{\mathfrak{A}} \sim_{\mathcal{L},\Sigma} \mathfrak{B}, b^{\mathfrak{B}}$.

# H Proofs for Section 7

**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 $\mathcal{ALC}$.*

The proof of Theorem 11 is inspired by the proof of the undecidability of conservative extensions and projective conservative extensions in every extension of the three-variable fragment GF$^3$ of GF (Jung et al. 2017). Unfortunately, it is not clear how to achieve a direct reduction

of conservative extensions to separability for GF. The relativization that was used for languages in $\mathsf{DL}_{\mathsf{ni}}$ does not work because non-conservativity in GF is witnessed by *sentences* while separability is witnessed by *open formulas*.

The proof is by a reduction from the halting problem of two-register machines. A (deterministic) *two-register machine (2RM)* is a pair $M = (Q, P)$ with $Q = q_0, \dots, q_\ell$ a set of *states* and $P = I_0, \dots, I_{\ell-1}$ a sequence of *instructions*. By definition, $q_0$ is the *initial state*, and $q_\ell$ the *halting state*. For all $i < \ell$,

- either $I_i = +(p, q_j)$ is an *incrementation instruction* with $p \in \{0, 1\}$ a register and $q_j$ the subsequent state;
- or $I_i = -(p, q_j, q_k)$ is a *decrementation instruction* with $p \in \{0, 1\}$ a register, $q_j$ the subsequent state if register $p$ contains 0, and $q_k$ the subsequent state otherwise.

A *configuration* of $M$ is a triple $(q, m, n)$, with $q$ the current state and $m, n \in \mathbb{N}$ the register contents. We write $(q_i, n_1, n_2) \Rightarrow_M (q_j, m_1, m_2)$ if one of the following holds:

- $I_i = +(p, q_j)$, $m_p = n_p + 1$, and $m_{1-p} = n_{1-p}$;
- $I_i = -(p, q_j, q_k)$, $n_p = m_p = 0$, and $m_{1-p} = n_{1-p}$;
- $I_i = -(p, q_k, q_j)$, $n_p > 0$, $m_p = n_p - 1$, and $m_{1-p} = n_{1-p}$.

The *computation* of $M$ on input $(n, m) \in \mathbb{N}^2$ is the unique longest configuration sequence $(p_0, n_0, m_0) \Rightarrow_M (p_1, n_1, m_1) \Rightarrow_M \cdots$ such that $p_0 = q_0$, $n_0 = n$, and $m_0 = m$. The halting problem for 2RMs is to decide, given a 2RM $M$, whether its computation on input $(0, 0)$ is finite (which implies that its last state is $q_\ell$).

We convert a given 2RM $M$ into a labeled GF$^3$ KB $(\mathcal{K}, \{a\}, \{b\})$, $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ and signature $\Sigma$ such that $M$ halts iff $(\mathcal{K}, \{a\}, \{b\})$ is (non-)projectively GF($\Sigma$)-separable iff $(\mathcal{K}, \{a\}, \{b\})$ is (non-)projectively $\mathcal{ALC}(\Sigma)$-separable. Let $M = (Q, P)$ with $Q = q_0, \dots, q_\ell$ and $P = I_0, \dots, I_{\ell-1}$. We assume w.l.o.g. that $\ell \geq 1$ and that if $I_i = -(p, q_j, q_k)$, then $q_j \neq q_k$. In $\mathcal{K}$, we use the following set of relation symbols:

- a binary symbol $N$ connecting a configuration to its successor configuration;
- binary symbols $R_1$ and $R_2$ that represent the register contents via the length of paths;
- unary symbols $q_0, \dots, q_\ell$ representing the states of $M$;
- a unary symbol $S$ denoting points where a computation starts.
- a unary symbol $D$ used to represent that there is some defect;
- binary symbols $D_p^+, D_p^-, D_p^=$ used to describe defects in incrementing, decrementing, and keeping register $p \in \{0, 1\}$;
- ternary symbols $H_1^+, H_2^+, H_1^-, H_2^-, H_1^=, H_2^=$ used as guards for existential quantifiers.

The signature $\Sigma$ consists of the symbols from the first four points above.

We define the ontology $\mathcal{O}$ as the set of several GF$^3$ sentences.[4] The first sentence initializes the starting configuration:

$$\forall x(Sx \to (q_0 x \wedge \neg\exists y\, R_0 xy \wedge \neg\exists y\, R_1 xy))$$

Second, whenever $M$ is not in the final state, there is a next configuration with the correctly updated state. For $0 \leq i < \ell$, we include:

$$\begin{aligned}
&\forall x(q_i x \to \exists y\, Nxy) \\
&\forall x(q_i x \to \forall y(Nxy \to q_j y)) && \text{if } I_i = +(p, q_j) \\
&\forall x((q_i x \wedge \neg\exists y R_p xy) \to \forall y(Nxy \to q_j y)) \\
& && \text{if } I_i = -(p, q_j, q_k) \\
&\forall x((q_i x \wedge \exists y R_p xy) \to \forall y(Nxy \to q_k y)) \\
& && \text{if } I_i = -(p, q_j, q_k)
\end{aligned}$$

Moreover, if $M$ is in the final state, there is no successor configuration:

$$\forall x(q_\ell x \to \neg\exists y\, Nxy).$$

The next conjunct expresses that either $M$ does not halt or the representation of the computation of $M$ contains a defect. It crucially uses non-$\Sigma$ relation symbols. It takes the shape of

$$\forall x\,(Dx \to \exists y\,(Nxy \wedge \psi xy))$$

where $\psi xy$ is the following disjunction which ensures that there is a concrete defect $(D_p^+, D_p^-, D_p^=)$ here or some defect $(D)$ in some successor state:

$$\begin{aligned}
&D(y) \vee \\
&\bigvee_{I_i = +(p,q_j)} (q_i x \wedge q_j y \wedge (D_p^+ xy \vee D_{1-p}^= xy)) \vee \\
&\bigvee_{I_i = -(p,q_j,q_k)} (q_i x \wedge q_k y \wedge (D_p^- xy \vee D_{1-p}^= xy)) \vee \\
&\bigvee_{I_i = -(p,q_j,q_k)} (q_i x \wedge q_j y \wedge (D_p^= xy \vee D_{1-p}^= xy))
\end{aligned}$$

Finally, using the ternary symbols we make sure that the defects are realized, for example, by taking:

$$\begin{aligned}
&\forall x \forall y \big(D_p^+ xy \to \\
&\quad (\neg\exists z\, R_p yz \vee (\neg\exists z\, R_p xz \wedge \exists z\,(R_p yz \wedge \exists x R_p zx)) \vee \\
&\quad \exists z(H_1^+ xyz \wedge R_p xz \wedge \exists x(H_2^+ xzy \wedge R_p yx \wedge D_p^+ zx)))\big).
\end{aligned}$$

Similar conjuncts implement the desired behaviour of $D_p^=$ and $D_p^-$; since they are constructed analogously to the last three lines above (but using guards $H_j^-$ and $H_j^=$), details are omitted.

Finally, we define a database $\mathcal{D}$ by taking

$$\mathcal{D} = \{S(a), D(a), S(b)\}.$$

Lemmas 23 and 24 below establish correctness of the reduction and thus Theorem 11.

[4] The formulas that are not syntactically guarded can easily be rewritten into such formulas.

**Lemma 23** *If $M$ halts, then there is an $\mathcal{ALC}(\Sigma)$ concept that non-projectively separates $(\mathcal{K}, \{a\}, \{b\})$.*

**Proof.** The idea is that the separating $\mathcal{ALC}(\Sigma)$ concept describes the halting computation of $M$, up to $\mathcal{ALC}(\Sigma)$-bisimulations. More precisely, assume that $M$ halts. We define an $\mathcal{ALC}(\Sigma)$ concept $C$ such that $\mathcal{K} \models \neg C(a)$, but $\mathcal{K} \not\models \neg C(b)$. Intuitively, $C$ represents the computation of $M$ on input $(0, 0)$, that is: if the computation is $(q_0, n_0, m_0), \ldots, (q_k, n_k, m_k)$, then there is an $N$-path of length $k$ (but not longer) such that any object reachable in $i \leq k$ steps from the beginning of the path is labeled with $q_i$, has an outgoing $R_0$-path of length $n_i$ and no longer outgoing $R_0$-path, and likewise for $R_1$ and $m_i$. In more detail, consider the $\Sigma$-structure $\mathfrak{A}$ with

$$\begin{aligned}\mathrm{dom}(\mathfrak{A}) = \{0, \ldots, k\} \cup \{a^i_j \mid 0 < i \leq k, 0 < j < n_i\} \cup \\ \{b^i_j \mid 0 < i \leq k, 0 < j < m_i\}\end{aligned}$$

in which

$$\begin{array}{rcl} N^{\mathfrak{A}} & = & \{(i, i+1) \mid i < k\} \\ R_1^{\mathfrak{A}} & = & \bigcup_{i \leq k} \{(i, a^i_1), (a^i_1, a^i_2), \ldots, (a^i_{n_i-2}, a^i_{n_i-1})\} \\ R_2^{\mathfrak{A}} & = & \bigcup_{i \leq k} \{(i, b^i_1), (b^i_1, b^i_2), \ldots, (b^i_{m_i-2}, b^i_{m_i-1})\} \\ S^{\mathfrak{A}} & = & \{0\} \\ q^{\mathfrak{A}} & = & \{i \mid q_i = q\} \text{ for any } q \in Q. \end{array}$$

Then let $C$ be the $\mathcal{ALC}(\Sigma)$ concept that describes $\mathfrak{A}$ from the point of 0 up to $\mathcal{ALC}(\Sigma)$-bisimulations. Clearly, $\mathcal{K} \cup \{C(b)\}$ is satisfiable. However, $\mathcal{K} \cup \{C(a)\}$ is unsatisfiable since the enforced computation does not contain a defect and cannot be extended to have one. In particular, there are no $N$-paths of length $> k$ in any model of $\mathcal{K} \cup \{C(a)\}$ and there are no defects in register updates in any model of $\mathcal{K} \cup \{C(a)\}$. ❑

The following lemma implies that if $M$ does not halt, then $(\mathcal{K}, \{a\}, \{b\})$ is neither projectively $\mathcal{L}(\Sigma)$-separable nor non-projectively $\mathcal{L}(\Sigma)$-separable for $\mathcal{L} =$ GF and in fact for every logic $\mathcal{L}$ between GF and FO.

**Lemma 24** *If $M$ does not halt, then for every model $\mathfrak{A}$ of $\mathcal{K}$, there is a model $\mathfrak{B}$ of $\mathcal{K}$ such that $(\mathfrak{A}, b^{\mathfrak{A}})$ is $\Gamma$-ismorphic to $(\mathfrak{B}, a^{\mathfrak{B}})$ where $\Gamma$ consists of all symbols except $\mathrm{sig}(\mathcal{O}) \setminus \Sigma$.*

**Proof.** Let $\mathfrak{A}$ be a model of $\mathcal{K}$. We obtain $\mathfrak{B}$ from $\mathfrak{A}$ by re-interpreting $a^{\mathfrak{B}} = b^{\mathfrak{A}}$ and inductively defining the extensions of the symbols from

$$\begin{aligned}\mathrm{sig}(\mathcal{O}) \setminus \Sigma = \{D, D_p^+, D_p^-, D_p^=, \\ H_1^+, H_2^+, H_1^-, H_2^-, H_1^=, H_2^=\}.\end{aligned}$$

We start with $D^{\mathfrak{B}} = \{a^{\mathfrak{B}}\}$ and $X^{\mathfrak{B}} = \emptyset$ for all other symbols $X$ from $\mathrm{sig}(\mathcal{O}) \setminus \Sigma$. Then, whenever $d \in D^{\mathfrak{B}}$ we distinguish two cases:

- If there is an $N$-successor $e$ of $d$ such that the counters below $d$ and $e$ are not correctly updated with respect to the states at $d, e$, set the extensions of the symbols in $D_p^+, D_p^-, D_p^=, H_1^+, H_2^+, H_1^-, H_2^-, H_1^=, H_2^=$ so as to represent the defect and finish the construction of $\mathfrak{B}$.
- Otherwise, choose an $N$-successor $e$ of $d$ and add $e$ to $D^{\mathfrak{B}}$.

Note that, since $M$ does not halt, we can always find such an $N$-successor as in the second item. ❑

**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.*

The proof of Theorem 12 relies on the encoding of GF and $FO^2$-KBs into formulas in GF and $FO^2$.

For Points 1 and 2 we can use the extensions of GF and $FO^2$ with constants as we only require Craig interpolants in FO and the upper bounds for satisfiability of GF and $FO^2$ still hold for their extensions with constants. We start with GF. Assume a labeled GF-KB $(\mathcal{K}, \{a\}, \{b\})$ with $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ is given. Let $\Sigma$ be a signature. Obtain $\varphi'_{\mathcal{K},\Sigma,a}$ from $\mathcal{K}$ by

- replacing all relation symbols $R \notin \Sigma$ by fresh relation symbols $R_a$;
- replacing all individual names $c \notin \Sigma \cup \{a\}$ by fresh individual names $c_a$;
- replacing $a$ by a fresh variable $x$ and adding $x = a$ if $a \in \Sigma$;

and taking the conjunction of the resulting set of formulas. Define $\varphi'_{\mathcal{K},\Sigma,b}$ in the same way but with $a$ replaced by $b$. Then an FO-formula $\varphi$ strongly $\Sigma$-separates $(\mathcal{K}, \{a\}, \{b\})$ iff $\varphi$ is an FO-interpolant for $\varphi'_{\mathcal{K},\sigma,a}, \neg\varphi'_{\mathcal{K},\Sigma,b}$, and we have proved Point 1. For Point 2, observe that $\varphi'_{\mathcal{K},\sigma,a}, \varphi'_{\mathcal{K},\Sigma,b}$ are in $FO^2$ if $\mathcal{K}$ is an $FO^2$-KB. Thus, the argument applies to $FO^2$ as well and we have proved Point 2.

For Points 3 and 4 we assume that $\Sigma$ is a relational signature and as we aim to apply results on the complexity of interpolant existence that have been proved for GF and $FO^2$ *without* constants we cannot use constants in the construction of the encodings.

Assume a labeled GF-KB $(\mathcal{K}, \{a\}, \{b\})$ with $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ is given and $\Sigma$ is a relational signature. Consider the formulas $\varphi_{\mathcal{K},\Sigma,a}$ and $\varphi_{\mathcal{K},\Sigma,b}$ from the main paper (defined in the obvious way for GF). To obtain GF-formulas $\varphi^{\mathrm{GF}}_{\mathcal{K},\Sigma,a}$ and $\varphi^{\mathrm{GF}}_{\mathcal{K},\Sigma,b}$, take fresh relation symbols $R_{\mathcal{D},a}$ and $R_{\mathcal{D},b}$ of arity $n$, where $n$ is the number of individuals in $\mathcal{D}$. Then add $R_{\mathcal{D},a}(\vec{y})$ to $\mathcal{K}_{\Sigma,a}$ when constructing $\varphi_{\mathcal{K},\Sigma,a}(x)$, where $\vec{y}$ is an enumeration of the variables in $\mathcal{K}_{\Sigma,a}$. Do the same to construct $\varphi^{\mathrm{GF}}_{\mathcal{K},\Sigma,b}(x)$, using $R_{\mathcal{D},b}$ instead of $R_{\mathcal{D},a}$. The formulas $\varphi^{\mathrm{GF}}_{\mathcal{K},\Sigma,a}$ and $\neg\varphi^{\mathrm{GF}}_{\mathcal{K},\Sigma,b}$ are in GF and play the same role as the formulas $\varphi_{\mathcal{K},\Sigma,a}$ and $\varphi_{\mathcal{K},\Sigma,b}$. In particular, for any formula $\varphi$ the following are equivalent:

1. $\varphi$ strongly $\Sigma$-separates $(\mathcal{K}, \{a\}, \{b\})$;
2. $\varphi$ is an interpolant for $\varphi^{\mathrm{GF}}_{\mathcal{K},\Sigma,a}(x), \neg\varphi^{\mathrm{GF}}_{\mathcal{K},\Sigma,b}(x)$.

The complexity upper bound now follows from the result that interpolant existence in GF is decidable in 3EXPTIME (Jung and Wolter 2021). For $\mathrm{FO}^2$ we proceed as follows. We introduce for every individual name $c$ in $\mathcal{D}$ a unary relation symbol $A_c$ and encode $\mathcal{D}$ using the sentences $\exists x A_c(x) \wedge \forall x \forall y (A_c(x) \wedge A_c(y) \rightarrow x = y)$, for $c \in \mathrm{ind}(\mathcal{D})$, and

- $\exists x \exists y R(x,y) \wedge A_c(x) \wedge A_{c'}(y)$ for every $R(c,c') \in \mathcal{D}$;
- $\exists x A(x) \wedge A_c(x)$ for every $A(c) \in \mathcal{D}$.

Let $\varphi^2_{\mathcal{K},\Sigma,a}$ be the conjunction of the sentences above, the sentences in $\mathcal{O}$, and the formula $A_a(x)$, where we also replace all relation symbols in $\mathcal{K}$ that are not in $\Sigma$ by fresh relation symbols $R_a$. Define $\varphi^2_{\mathcal{K},\Sigma,b}$ in the same way with $a$ replaced by $b$ and where the unary relation symbols $A'_c$ used to encode individuals $c$ are disjoint from the unary relation symbols used for this purpose in $\varphi^2_{\mathcal{K},\Sigma,a}$. Then we have that $\varphi^2_{\mathcal{K},\Sigma,a}$ and $\varphi^2_{\mathcal{K},\Sigma,b}$ are in $\mathrm{FO}^2$ and a formula $\varphi$ strongly $\Sigma$-separates $(\mathcal{K}, \{a\}, \{b\})$ iff it is an interpolant for $\varphi^2_{\mathcal{K},\Sigma,a}(x), \neg\varphi^2_{\mathcal{K},\Sigma,b}(x)$. The complexity upper bound now follows from the result that interpolant non-existence in $\mathrm{FO}^2$ is decidable in N2EXPTIME (Jung and Wolter 2021).

The complexity lower bounds can be proved by generalizing in a straightforward way the reductions from interpolant existence to the existence of strongly separating formulas from the languages in $\mathsf{DL}_{\mathsf{ni}}$ in the previous section to GF and $\mathrm{FO}^2$ and using the complexity lower bounds for interpolant existence proved in (Jung and Wolter 2021).