University of Liverpool
Department of Computer Science

Logical Separability *of* Open-World Data

Thesis submitted in accordance with the requirements of the University of Liverpool for the degree of Doctor in Philosophy by

Hadrien Pulcini

October 2022

FIRST SUPERVISOR	Pr. Frank Wolter, *University of Liverpool*
SECOND SUPERVISOR	Pr. Boris Konev, *University of Liverpool*
INTERNAL EXAMINER	Dr. Louwe B. Kuijer, *University of Liverpool*
EXTERNAL EXAMINER	Dr. Víctor Guttiérrez-Basulto, *Cardiff University*

Abstract

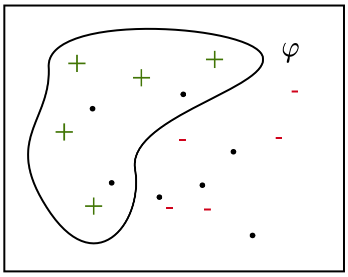
For any two fragments $\mathcal{L}_O, \mathcal{L}_S$ of first-order logic, we define as $(\mathcal{L}_O, \mathcal{L}_S)$-separability the problem of, given an ontology formulated in $\mathcal{L}_O$ and a database containing positively and negatively labeled tuples, the existence of a *separating* formula φ in $\mathcal{L}_S$, *i.e.*, that applies to all positive tuples but no negative one. We distinguish several versions of that problem, depending on how the notion of separation is defined or on the symbols φ may contain. For each version and various combinations of languages $(\mathcal{L}_O, \mathcal{L}_S)$ (ranging over first-order logic, its guarded fragment, its two-variable fragment, expressive description logics, conjunctive queries or unions thereof), we provide model-theoretic characterisations and complexity bounds for $(\mathcal{L}_O, \mathcal{L}_S)$-separability. To that end, we uncover and make use of the interplay between separability and well-studied decision problems such as query evaluation, satisfiability, interpolation and deciding conservative extensions.

Contents

Chapter **0**

Introduction

THIS thesis discusses the following problem. Given two sets of data points E^+ and E^- of *positive* and *negative* examples (possibly both within a larger set) labeled with information under logical form, is there a logical expression φ that only applies to the positive ones?



A crucial feature of our framework is that we assume our information to be incomplete: if all one knows is that a satisfies A, one cannot conclude that a does not satisfy B. This feature is well-known in Knowledge Representation terminology as the *open-world assumption*, as opposed to *closed-world* where information is assumed complete and absence of explicit satisfaction equates to satisfaction of absence. With open-world semantics, what we mean by "separating" is not as straightforward: the positive examples may share a common property that, to our knowledge, is not *forced* onto any negative example, but still *allowed* for at least one of them.

a satisfies A	*a and b have A in common;*
b satisfies A	*c may or may not satisfy A.*
c satisfies B	

In consequence, we distinguish two approaches to separation: either we require from a separating formula that it *does not necessarily* apply to any negative example, or that it *necessarily does not* apply to any negative example. We say the formula *weakly* separates E^+ from E^- in the former case and *strongly* in the latter. In the above example, a and b are weakly separated from c but not strongly.

Our main motivation for looking at the logical separability problem comes from its applications to Knowledge Representation, as we explain in detail further below. Knowledge Representation is a four-decade-old subfield of Artificial Intelligence that aims at designing structured computer representations of the world. *Ontologies* are currently one of its main formalisms. They express general knowledge through semantic links between *concepts*. They can be understood as a generalization of dictionaries or taxonomies. On top of information that is provided by our data points (*databases*), we also take ontologies into account (again, under the open-world assumption, customary for ontologies).

DATABASE	a satisfies A b satisfies A c satisfies B	ONTOLOGY	B implies A

Now, in any world realizing the above,
c satisfies A so a, b are not weakly separable from c.

We work within first-order logic (**FO**): all the information given by the ontology and the database is assumed to be expressed as first-order formulas, and so is the separating formula we look for. In our framework, the informal example above is understood as follows.

DATABASE	$A(a)$ $A(b)$ $B(c)$	ONTOLOGY	$\forall x(B(x) \to A(x))$

We also work with strictly less expressive, but decidable, fragments of first-order logic. Examples might be separable by a first-order formula but not by any formula from some less expressive fragment. If the database consists of $\{A(a), B(b), C(c)\}$ where A, B, C are atomic, a and b are not separable from c in a language that cannot express disjunction.

Separating sets of data points is at the heart of Supervised Machine Learning. The information carried by our data corresponds to features in standard Machine Learning terminology. One difference is that data points there are usually described independently from each other whereas we also consider relations between them.

DATABASE	$R(a,b)$ $R(b,c)$ $B(c)$	*a and b are weakly separated from c by $\varphi(x) = \exists y R(x,y)$.*

A more precise formulation of the separability problem is as follows.

> Given a database $\mathcal{D}$ and a constantless ontology $\mathcal{O}$ that together constitute a *knowledge base* $\mathcal{K}$, formulated in a fragment $\mathcal{L}_O$ of first-order logic, sets E^+, E^- of tuples of constants occurring in $\mathcal{D}$ and a separation language $\mathcal{L}_S$ (also a fragment of **FO**), does $\mathcal{L}_S$ contain a constantless formula φ such that
>
> $$\text{for all } \mathbf{a} \in E^+, \mathbf{b} \in E^-, \quad \begin{cases} \mathcal{K} \models \varphi(\mathbf{a}) \\ \begin{cases} \mathcal{K} \not\models \varphi(\mathbf{b}) & \text{(weak case)} \\ \mathcal{K} \models \neg\varphi(\mathbf{b}) & \text{(strong case)} \end{cases} \end{cases} \quad ?$$

We refer to that decision problem as $(\mathcal{L}_O, \mathcal{L}_S)$*-separability*. We mostly focus on the case where constants are not allowed in ontology and separating formulas. While it obviously trivializes the problem to allow constants from E^+, E^- in the separating formulas (as the equality symbol is present), allowing other constants is of interest as well but will only briefly be discussed in this thesis. Aside from the distinction between weak and strong separability, we also include other interesting dimensions. If φ is allowed to contain symbols that do not occur in $\mathcal{O} \cup \mathcal{D}$, we speak of *projective* separability (*non-projective* otherwise). We also look at the case in which φ is only allowed to feature a predetermined set of symbols from $\mathcal{O} \cup \mathcal{D}$, as part of the input. We denote it as *restricted* and the opposite case – where no restriction is given – as *full signature* separability.

The practical suitability of weak versus strong separability depends on how complex the ontology is, as we illustrate later on in Section 1.6. Restricted separability is not considered in any of the applications we describe; we include it as a natural generalisation. Projective separability is also not yet considered in applications. In a theoretical context, as we show, it proves important to equivalence results.

The contribution of this thesis can be summarized as follows. We study, for various combinations of the above variations, ontology language $\mathcal{L}_O$ and separation language $\mathcal{L}_S$,

- the computational complexity of $(\mathcal{L}_O, \mathcal{L}_S)$-separability, either with the ontology as part of the input (*combined complexity*) or without (*data complexity*),
- relations of inclusion between each variation - how languages relate to each other in terms of separating power, the consequences of projectivity, etc.

To this end, we often aim at characterising separability in model-theoretic terms. Those characterisations, often unexpectedly, revealed fruitful connections between separability and well-known problems such as *satisfiability, query evaluation, interpolation* and deciding *conservative extensions*.

Aside from first-order logic **FO**, we pay particular attention to *description logics*. Responding to KR's need for a logical formalism that would support automated reasoning and overcome the drawbacks of first-order logic, description logics were designed to offer a satisfactory trade-off between expressivity and practicality, the latter ranging from user interface (readability for non-domain experts) to decidability of the core reasoning problems (satisfiability, instance checking...). The reader is referred to [BHLS17, BCMNP03] for an extensive account of their history and their applications. We consider the 'standard' DL $\mathcal{ALC}$ and its lattice of extensions induced by the constructors $\mathcal{I}, \mathcal{O}, \mathcal{Q}$ (inverse roles, nominals and number restrictions). As separation languages, we also consider the fundamental database language **CQ** (of conjunctive queries) and its closure **UCQ** under disjunction. We include the Guarded Fragment **GF** and the two-variable fragment $\mathbf{FO}^2$ of **FO** for the interesting compromise they offer between expressiveness (they both subsume $\mathcal{ALCI}$) and desirable properties such as decidability or the finite model property. More anecdotally, we discuss strictly positive description logics $\mathcal{EL}, \mathcal{ELI}$ and the Guarded Negation Fragment **GNF** of **FO** that subsumes **GF** and **UCQ**.

A related notion to separability, which we choose to leave outside of our discussion, is *definability*. It corresponds to the particular case where the negative data points are exactly all the non-positive ones. Definability is relevant to many of the same applications as separability and notable theoretical results have been obtained in parallel to separability. For the most part, they are corollaries of results on separability and the techniques used to obtain them are essentially the same.

Following the way our results and proof techniques tend to cluster, we dedicate a chapter to every combination of weak/strong and restricted/full separability.

OUTLINE

Chapter 2. Full weak separability. We give a semantic characterisation of projective weak full $(\mathbf{FO},\mathbf{FO})$-separability. The main consequences are that it coincides with projective and non-projective $(\mathbf{FO},\mathcal{L}_S)$-separability whenever $\mathbf{UCQ} \subseteq \mathcal{L}_S \subseteq \mathbf{FO}$ and that for all $\mathbf{FO}$-fragments $\mathcal{L}_O, \mathcal{L}_S$ such that $\mathcal{L}_S$ contains $\mathbf{UCQ}$ and $\mathcal{L}$ has the relativization property, weak (projective) $(\mathcal{L}_O, \mathcal{L}_S)$-separability is reducible to the complement of rooted $\mathbf{UCQ}$-evaluation on $\mathcal{L}$-knowledge bases.

As opposed to $\mathbf{FO}$, the satisfiability problem in $\mathbf{FO}^2$ is decidable, then separability for $\mathbf{FO}^2$-knowledge bases is potentially decidable. We show that (projective or not) full weak $(\mathbf{FO}^2,\mathbf{FO}^2)$ and $(\mathbf{FO}^2,\mathbf{FO})$-separability are still undecidable. This is done via reduction from a tiling problem, without using any model-theoretic characterisation of separability. We show that, in contrast to $\mathbf{FO}$, projectivity can make a difference with $\mathbf{FO}^2$ as a separating language. We show however that even with the help of projectivity, $\mathbf{FO}^2$ has strictly less separating power than $\mathbf{FO}$.

We then consider $(\mathcal{L},\mathcal{L})$-separability for $\mathcal{L} \in \mathbf{DL}_{\mathcal{IQ}}$, where $\mathbf{DL}_{\mathcal{IQ}}$ denotes the set of description logics that can be constructed from $\mathcal{ALC}$ using the extensions $\mathcal{I}, \mathcal{Q}$. We establish a model-theoretic characterisation of separability that is uniform over all languages in $\mathbf{DL}_{\mathcal{IQ}}$. Projectivity is crucial. The characterisation reveals a connection between the decision problems of separability and UCQ evaluation, from which we deduce combined complexity bounds for full signature weak projective $(\mathcal{L},\mathcal{L})$-separability, $\mathcal{L} \in \mathbf{DL}_{\mathcal{IQ}}$ (NEXP-complete for all except EXP-complexity for $\mathcal{ALCQI}$). We also show PSPACE-completeness in data complexity for $\mathcal{ALC}$.

By tweaking the model-theoretic characterisation of projective separability, we (much less easily) establish a characterisation of non-projective full weak $(\mathcal{ALCI},\mathcal{ALCI})$-separability. It follows that we can easily reduce the projective problem to the non-projective one in polynomial time. Then, non-projective full weak $(\mathcal{ALCI},\mathcal{ALCI})$-separability is NEXP-complete in combined complexity and in data complexity.

Projective and non-projective $(\mathbf{GF},\mathbf{GF})$-separability turn out to behave similarly to $(\mathcal{ALCI},\mathcal{ALCI})$-separability in many ways. The projective and non-projective cases also do not coincide. Then, $\mathbf{GF}$ also admits a notion of bisimulation, which we use to characterise separability as for $\mathcal{ALCI}$: in the projective case we also characterise separability following a "bisimulation-simulation-homomorphism" pattern, while in the non-projective case we also rely on a notion of "type incompleteness". The results are, however, significantly more difficult to establish. An analogous connection with $\mathbf{UCQ}$-evaluation then yields 2EXP-completeness of (projective) $(\mathbf{GF},\mathbf{GF})$-separability.

Finally, we show undecidability of (projective) full weak $(\mathcal{L}_O, \mathcal{L}_S)$-separability for $\mathcal{L}_O \supseteq \mathcal{ALC}$ and $\mathcal{L}_S \in \{\mathcal{EL}, \mathcal{ELI}\}$.

Chapter 3. Full strong separability. A first observation is that projectivity does not affect strong separability. As in the weak case, full signature strong $(\mathcal{L}_O, \mathcal{L}_S)$-separability coincides with full signature strong $(\mathcal{L}_O, \mathcal{L}'_S)$-separability for all **FO**-fragments $\mathcal{L}_O, \mathcal{L}_S, \mathcal{L}'_S$ such that $\mathcal{L}_S, \mathcal{L}'_S$ contain **UCQ** and either $\mathcal{L}_O \in \{\mathcal{ALC}, \mathbf{GF}\} \cup \{\mathcal{L} \mid \mathbf{UCQ} \subseteq \mathcal{L} \subseteq \mathbf{FO}\}$. However, the proofs are fundamentally different from the weak case. While the useful link for complexity was between separability and query evaluation in the weak case, here our characterisations imply that *satisfiability* provides an upper bound on the complexity of separability. We then obtain, in combined complexity, ExP-completeness of full strong $(\mathcal{L}, \mathcal{L})$-separability where $\mathcal{L} \in \{\mathcal{ALC}; \mathcal{ALCI}, \mathbf{FO}^2\}$ and 2ExP-completeness for **GF**. We also obtain the same data complexity as (the complement of) satisfiability, that is coNP-completeness.

Chapter 4. Restricted weak separability. The main equivalence results from the full case fail when a signature is part of the input. We give model-theoretic characterisations for projective separability in DL and observe that they do not apply to $\mathcal{ALCIO}$. We observe that it follows from our main characterisation theorem that projective $\mathcal{ALCI}$ and $\mathcal{ALCO}$-separability can be non-projectively captured by a language combining UCQs and DLs. We show that for the DLs we consider, separability is tightly connected to the problem of deciding *conservative extensions*. That reduction implies 2ExP-hardness for projective $(\mathcal{ALC}, \mathcal{ALC})$ and $(\mathcal{ALCI}, \mathcal{ALCI})$-separability. We sketch a matching 2ExP upper bound via tree automata and the characterisation theorem.

Our other main investigation is on the impact of admitting only concept names as helper symbols for projective separability, versus also admitting role names and/or constants. In particular we prove undecidability of projective weak $(\mathcal{ALC}, \mathcal{ALCO})$-separability if constants are allowed as helper symbols. This is done via reduction from an undecidable tiling problem. Finally, we use the same tiling problem to show that (projective) $(\mathcal{ALC}, \mathcal{ALCFIO})$-separability is undecidable.

Chapter 5. Restricted strong separability. We first look at the problem in the context of $\mathbf{DL}_{\mathcal{IO}}$ ontologies. As in the full case, projective and non-projective separability coincide. Our main observation is that restricted strong separability can be seen as an *interpolation* problem. Because **FO** has the Craig Interpolation Property, we can even see it as an entailment problem, via a characterisation of $(\mathcal{L}, \mathbf{FO})$-separability for all $\mathcal{L} \in \mathbf{DL}_{\mathcal{IO}}$. From that characterisation, it follows that $(\mathcal{L}, \mathbf{FO})$-separability is ExP-complete and that **FO**'s separating power is matched by Boolean hybrid CQs – extensions of hybrid CQs introduced in the previous section. From the link between separability and interpolation, we deduce that restricted strong $(\mathcal{L}, \mathcal{L})$-separability is 2EXP-complete for all $\mathcal{L} \in$ $\mathbf{DL}_{\mathcal{IO}}$. We then focus on **GF** and $\mathbf{FO}^2$. While it needs some tweaking, the same link between separability and interpolation holds there. We can, once again, use the CIP of **FO** and results on the complexity of interpolants to deduce combined complexity bounds (3EXP-completeness for $(\mathbf{GF}, \mathbf{GF})$, 2EXP-hardness and CON2EXP for $(\mathbf{FO}^2, \mathbf{FO}^2)$).

RELATED WORK

Aside from the consideration of relations between data points, another key difference between our approach and a statistical ML approach is that the latter's aim is less to find the explicit nature of what separates the points than to be able to accurately classify future examples. In the applications that motivated our investigation however, the explicit nature of the separating property is interesting in and of itself, especially when under logical form. In the next paragraphs we give an overview of those applications and discuss how they relate to our treatment of the separability problem.

Early efforts towards inductive learning of first-order formulas converged in the 1990s into what became known as *Inductive Logic Programming* (ILP), baptized in [Mu91]. Aside from better explainability, ILP offered an interesting alternative to statistical ML by generalizing better over small datasets and taking relational data into account. With the diversification of KR formalisms came the birth of what could be considered as other forms of Inductive Logic Programming, applied to the languages underlying those systems. Description logics are now ubiquitous in KR thus primary candidates for any research that involves inductive learning of logical formulas in the context of KR. The need for such research quickly arose to answer a fundamental bottleneck of KR: that the implementation and maintenance of ontologies require a great deal of manual input from domain experts. The subfield of *Ontology Learning*, a term coined in [MS01], emerged as an attempt to automatize this process. Among other research directions in Ontology Learning, to expand ontologies based on description logic concepts, much attention has been brought upon what is known as *Concept Learning*.

ILP & DL Concept Learning. Inspired by techniques from Inductive Logic Programming, *refinement operators* are used in DL concept learning to construct a concept that generalizes positive examples while not encompassing any negative ones. An ontology may or may not be present. A downward (resp. upward) refinement operator is a function that maps any hypothesis (*i.e.* formula) to a set of refinements (resp. generalisations) of that hypothesis: formulas that are more (resp. less) specific. The first transposition of the ILP refinement approach to DLs was done in [BN00] for the logic $\mathcal{ALER}$. Many have followed since then, both for weak separation [HL10, LS15, HS19] and strong separation [AEF08, Li12, AFR20]. Prominent systems include the DL LEARNER [BBLPW18, BLW16], DL-FOIL [AEFR18a] and its extension DL-FOCL [AEFR18b], SPaCEL [DGMT17], YINYANG [FIP07], and PFOIL-DL [MS15]. A method for generating strongly separating concepts based on bisimulations has been developed in [HHNST12, HNT15, DHNN18] and an approach based on answer set programming was proposed in [Li16].

Algorithms for DL concept learning typically aim to be complete, that is, to find a separating concept whenever there is one. Complexity lower bounds for separability as studied in this thesis then point to inherent limitations on any such algorithm. Undecidability even means that no learning algorithm can be both terminating and complete.

Computing the *least common subsumer* (LCS) of a set of concepts and the *most specific concept* (MSC) applying to a single data item [BCH92, N90, BKM99] can be viewed as DL concept learning in the case that only positive examples are available. The problem is trivialized if the separation language expresses disjunction. [TZ13] studies LCS/MSC in the context of $\mathcal{EL}$ in the presence of an ontology, gives a semantic characterisation of their existence and computing procedures: in $\mathcal{EL}$, with a single data item, MCS verification and existence are tractable. A recent study of LCS and MSC from a separability angle is done in [JLW20]. It extends the MSC to multiple examples, looks at special cases like empty ontologies, arbitrary signature restrictions, and adds $\mathcal{ELI}$ to the discussion. Complexity is dramatically increased from [TZ13]. It shows that for $\mathcal{EL}, \mathcal{ELI}$, the complement of separability can be mutually reduced with MCS verification.

Query-by-Example. Reverse engineering of database queries, or *Query-By-Example* (QBE) is another active field of relevant applications, see *e.g.* [CPT09, EPSZ13, CW17, KLS18, DG19, SW12] and [Mart19] for a recent survey. A query can be understood as a logical expression, typically used to retrieve elements from a database. QBE grew from the difficulty for non-expert users to formulate their desired queries. Instead, users may simply input some database examples (and counterexamples) for the 'notion' they are targeting and receive query suggestions from the QBE system in return. This can easily be seen as a separation problem where one looks for a separating formula in a query language. In particular, the language **CQ** of *conjunctive queries* is in fact the positive existential fragment of first-order logic.

A theoretical side to learnability and separability of queries burgeoned in the last 5 years, under both closed-world [AD16, BR17, KR18, CD21] and open-world [O19, CCL21, ADK16] semantics. We give a brief description of the closest approaches to ours. Under closed-world assumption, [AD16] shows **GI**-completeness of **FO**-*definability*, which can be seen as separability with no negative example. Earlier, coNEXP-completeness of **CQ**-separability was proved using a folklore semantic characterization based on product homomorphisms [CD15]. In [BR17] coNP-completeness is shown for **UCQ**-separability and approximation methods are designed to reach tractability. Separability with fragments of the query langage SPARQL on (closed-world) RDF graphs is also studied for its complexity in [ADK16], as well as definability.

In [GJS18], QBE is brought to the open-world setting and the separability problem there includes ontologies to the background knowledge. Model-theoretic characterisations and complexity bounds are given for what we, in our terminology, would refer as weak $(\mathcal{L}_O, \mathcal{L}_S)$-separability, where $\mathcal{L}_S \in \{\mathbf{CQ}, \mathbf{UCQ}\}$ and $\mathcal{L}_O \in \{\mathsf{Horn}\mathcal{ALCI}, \mathsf{Horn}\mathcal{ALC}, \mathcal{ELI}\}$. A distinction is made between restricted and full signature separability, as is done here. The presence of an ontology is shown to be computationally detrimental as complexity bounds range from EXP to 2EXP. As shown in [O19], using a more rudimentary language for ontologies such as DL-LITE$_\mathcal{R}$ reduces the complexity down to coNP-completeness for **UCQ** (arguably less interesting than **CQ** for generalizing, as disjunctions make it prone to overfitting) but the complexity stays as prohibitively high as before for **CQ**.

As is done here for more expressive logics, [F19] investigates "Concept-By-Example", *i.e.* full weak separability under ontologies formulated in $\mathcal{EL}$, with model-theoretic characterisations. This approach intersects both QBE and DL Concept Learning, as $\mathcal{ELI}$-concepts are equivalent to tree-shaped conjunctive queries. EXP-completeness in combined complexity is shown for $(\mathcal{L}_O, \mathcal{EL})$-separability for any $\mathcal{L}_O$ contained in $\mathcal{EL}$. Separability for a given role depth of the separating concept is also characterised and shown to be NP-complete.

Generating referring expressions. *Generating referring expressions* (GRE) has originated from linguistics (see [DK12] for a survey). A referring expression is any noun phrase that identifies an object in a given context (*e.g.* "the man on the left"). In GRE, one aims to design algorithms that can produce accurate referring expressions in the sense that they fit human intuition. It can be seen as a definability problem in our framework. Both weak and strong separability are conceivable: weak separability means that the positive data item is the only one that we are certain to satisfy the separating formula and strong separability means that in addition we are certain that the other data items do not satisfy the formula. Approaches to GRE such as the ones in [BTW16, AMOW21] aim for stronger guarantees, for instance by demanding that a referring expression for an object refers only to that object, in the context imposed by the ontology. GRE has recently been converging towards KR framework. In a closed-world context description logic concepts have also been proposed for singling out a domain element in an interpretation [AKS18]. The computation of referring expressions has recently received interest in the context of ontology-mediated querying [TW19].

Entity Comparison. As relevant field we can finally mention *Entity Comparison*, in which one aims to extract the similarities and the difference between two data points. RDF graphs are the standard format for displaying information from the Semantic Web, and SPARQL queries are the standard language for query answering on RDF data. An approach to entity comparison in RDF graphs is presented in [GHPS17, GHKP19]. There, SPARQL queries are used to describe both similarities and differences, under an open world semantics. The 'computing similarities' part is closely related to the LCS and MSC mentioned above. The 'computing differences' part is closely related to QBE and fits into our framework.

This dissertation draws on the following joint work with Maurice Funk, Jean Christoph Jung, Carsten Lutz and Frank Wolter.

- Sections 2.3/2.6/3.2:
  Learning Description Logic Concepts:
  When can Positive and Negative Examples be Separated?,

  published in *Proceedings of the 28th International Joint Conference on Artificial Intelligence* (IJCAI 2019). [FJLPW19]
- Chapters 2 and 3, aside from Sections 2.3/2.6/3.2:
  Logical Separability of Incomplete Data under Ontologies,

  published in *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning* (KR 2020). [JLPW20]
- Chapters 4 and 5:
  Separating Data Examples by Description
  Logic Concepts with Restricted Signatures,

  published in *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning* (KR 2021). [JLPW21]

Material from the above publications that is mentioned in the thesis but to which I have not contributed includes all results about full weak separability on $\mathcal{EL}(\mathcal{I})$-ontologies (mentioned in Ch. 2), 3EXP-completeness of restricted projective and non-projective weak $(\mathcal{ALCO}, \mathcal{ALCO})$ separability (Ch. 4), and undecidability of restricted weak $(\mathbf{GF}^3, \mathcal{L}_S)$-separability for $\mathcal{L}_S \supseteq \mathcal{ALC}$ (Ch. 4).

An article [JLPW22] building upon material presented in Chapters 2 and 3 has been published in the December 2022 issue of *Artificial Intelligence (AIJ)*.

Chapter **1**

Preliminaries

We introduce the required material to enunciate and investigate the separability problem. In particular, we introduce all languages studied in this thesis, with some of their essential model-theoretical and computational properties. Ultimately, we give a formal definition to the separability problem, together with some of its overarching properties.

§ 1.1. FIRST-ORDER LOGIC BACKGROUND

All logics considered in this thesis can be interpreted as fragments of first-order logic (**FO**), either directly or by some syntactic translation. We introduce **FO** and remind the reader of general facts in the context of **FO**. For a thorough introduction, see *e.g.* [Mark02].

We work with the standard syntax of first-order logic, but without function symbols.

1.1. *Definition.* Let var, cons and rel_n be countably infinite sets, for every $n \geqslant 1$. We respectively call their elements *variables*, *constants* and *relations of arity* n. Let $\mathsf{rel} = \bigcup_{n\geqslant 1} \mathsf{rel}_n$. Let the *full alphabet* be the set $\mathsf{var} \cup \mathsf{cons} \cup \mathsf{rel}$. Let an *atomic* **FO**-*formula* be any formula of the form $t_1 = t_2$ or $R(t_1, \ldots, t_n)$, where $n \geqslant 1, R \in \mathsf{rel}_n$ and $t_1, \ldots, t_n \in \mathsf{var} \cup \mathsf{cons}$. Let the set of **FO**-*formulas* be the smallest set S containing all atomic **FO**-formulas and such that, if $\varphi, \psi \in S$ and $x \in \mathsf{var}$, then $\neg\varphi, \varphi \wedge \psi, \exists x \varphi \in S$. For all **FO**-formulas φ, ψ and $x \in \mathsf{var}$, we use the following abbreviations.

$(\varphi \vee \psi)$	$\neg(\neg\varphi \wedge \neg\psi)$
$(\varphi \rightarrow \psi)$	$(\neg\varphi \vee \psi)$
$\top$	$(\varphi \vee \neg\varphi)$
$\bot$	$\neg\top$
$\forall x \varphi$	$\neg\exists x \neg\varphi$

We call *signature* any subset of $\mathsf{rel} \cup \mathsf{cons}$. We call a signature *relational* if it does not contain any constant symbol. For any **FO**-formula φ we define the *signature of* φ, written $\mathsf{sig}(\varphi)$, as the set of symbols from $\mathsf{rel} \cup \mathsf{cons}$ occurring in φ. For any set S of **FO**-formulas let $\mathsf{sig}(S) := \bigcup_{\varphi\in S} \mathsf{sig}(\varphi)$. We call *fragment* of **FO** any set $\mathcal{L}$ of **FO**-formulas closed under conjunction. We call its elements $\mathcal{L}$-formulas. Let $\mathcal{L}$ be a fragment of **FO** and Σ a signature. We call $\mathcal{L}(\Sigma)$-*formula* any φ in $\mathcal{L}$ with $\mathsf{sig}(\varphi) \subseteq \Sigma$. We respectively call $\mathcal{L}$-*sentence* and $\mathcal{L}(\Sigma)$-*sentence* any such formula with no unquantified (or *free*) variable.

We also use standard semantics, with the following notation.

1.2. *Definition.* Let a *model* be any triple $\mathfrak{A} = (\mathsf{dom}(\mathfrak{A}), (R^{\mathfrak{A}})_{R\in\mathsf{rel}}, (c^{\mathfrak{A}})_{c\in\mathsf{cons}})$ where $R^{\mathfrak{A}} \subseteq \mathsf{dom}(\mathfrak{A})^n$ if $R \in \mathsf{rel}_n$, and $c^{\mathfrak{A}} \in \mathsf{dom}(\mathfrak{A})$ for every $c \in \mathsf{cons}$. We call $\mathsf{dom}(\mathfrak{A})$ the *domain* of $\mathfrak{A}$, $R^{\mathfrak{A}}$ the *extension* of R in $\mathfrak{A}$ and $c^{\mathfrak{A}}$ the *interpretation* of c in $\mathfrak{A}$. We call *pointed model* any pair $(\mathfrak{A}, \mathbf{a})$ where $\mathfrak{A}$ is a model and $a \in \mathsf{dom}(\mathfrak{A})^n$ for some $n \geqslant 1$. Using the standard semantics, for any **FO**-formula $\varphi(x_1, \ldots, x_n)$, model $\mathfrak{A}$ and $\mathbf{a} = (a_1, \ldots, a_n) \in \mathsf{dom}(\mathfrak{A})^n$, we define $c^{\mathfrak{A}}(\mathbf{a}) = c^{\mathfrak{A}}$ and $x_i^{\mathfrak{A}}(\mathbf{a}) = a_i$ and we write $\mathfrak{A} \models \varphi(\mathbf{a})$ ($\mathfrak{A} \not\models \varphi(\mathbf{a})$ otherwise) to denote that $\mathfrak{A}$ *satisfies* φ *in* $\mathbf{a}$. It is inductively defined as such.

$$
\begin{array}{ll}
\mathfrak{A} \models (t_1 = t_2)(\mathbf{a}) & \text{if } t_1^{\mathfrak{A}}(\mathbf{a}) = t_2^{\mathfrak{A}}(\mathbf{a}) \\
\mathfrak{A} \models (R(t_1, \ldots, t_n))(\mathbf{a}) & \text{if } (t_1^{\mathfrak{A}}(\mathbf{a}), \ldots, t_n^{\mathfrak{A}}(\mathbf{a})) \in R^{\mathfrak{A}} \\
\mathfrak{A} \models (\neg\varphi)(\mathbf{a}) & \text{if } \mathfrak{A} \not\models \varphi(\mathbf{a}) \\
\mathfrak{A} \models (\varphi \land \psi)(\mathbf{a}) & \text{if } \mathfrak{A} \models \varphi(\mathbf{a}) \text{ and } \mathfrak{A} \models \psi(\mathbf{a}) \\
\mathfrak{A} \models (\exists x_i \varphi)(\mathbf{a}) & \text{if } \mathfrak{A} \models \varphi(a_1, \ldots, a_{i-1}, b, a_{i+1}, \ldots, a_n) \\
 & \quad \text{for some } b \in \mathsf{dom}(\mathfrak{A})
\end{array}
$$

We next introduce the Compactness Theorem (see *e.g.* [Mark02, §2]).

1.3. *Definition.* An **FO**-*theory* is a set of **FO**-sentences. For any **FO**-theory T and any model $\mathfrak{A}$ we write $\mathfrak{A} \models T$ ("$\mathfrak{A}$ *satisfies* T", or "$\mathfrak{A}$ *is a model of* T") if $\mathfrak{A} \models \varphi$ for every $\varphi \in T$. T is *satisfiable* if there exists $\mathfrak{A}$ such that $\mathfrak{A} \models T$. T is *finitely satisfiable* if every finite subset of T is satisfiable. We write $S(x_1, \ldots, x_n)$ if S is a set of formulas whose free variables are all among $\{x_1, \ldots, x_n\}$. We say S is satisfiable if there exists a pointed model $(\mathfrak{A}, \mathbf{a})$ such that $\mathfrak{A} \models \varphi(\mathbf{a})$ for all $\varphi \in S$, and S is finitely satisfiable if every finite subset S_f of S is satisfiable.

1.4. Theorem (Compactness)**.** *Every finitely satisfiable* **FO**-*theory is satisfiable.*

An immediate application of the compactness theorem extends it to sets of formulas that are not necessarily closed.

1.5. Corollary. *Every finitely satisfiable set* $S(x_1, \ldots, x_n)$ *of* **FO**-*formulas is satisfiable.*

The remaining definitions are basic model-theoretic notions, but we define them with respect to relations only and not constants; the separability problem involves constantless formulas in the vast majority of cases treated in this thesis.

1.6. *Definition.* Let $\mathfrak{A}, \mathfrak{B}$ be models, Σ a signature and $h : \mathsf{dom}(\mathfrak{A}) \to \mathsf{dom}(\mathfrak{B})$. h is a Σ-*homomorphism* if, for all $n \geqslant 1$, $R \in \mathsf{rel}_n \cap \Sigma$ and $(a_1, \ldots, a_n) \in \mathsf{dom}(\mathfrak{A})^n$, $(a_1, \ldots, a_n) \in R^{\mathfrak{A}}$ implies $(h(a_1), \ldots, h(a_n)) \in R^{\mathfrak{B}}$. We write $h : \mathfrak{A} \to^{\Sigma} \mathfrak{B}$. For every $\mathbf{a} = (a_1, \ldots, a_n)$ in $\mathsf{dom}(\mathfrak{A})^n$ and $\mathbf{b} = (b_1, \ldots, b_n)$ in $\mathsf{dom}(\mathfrak{B})^n$ we write $h : (\mathfrak{A}, \mathbf{a}) \to^{\Sigma} (\mathfrak{B}, \mathbf{b})$ if $h(a_1) = b_1, \ldots, h(a_n) = b_n$. We omit Σ if $\Sigma \supseteq \mathsf{rel}$. h is a Σ-*embedding* if h is an injective homomorphism and, for all $n \geqslant 1$, $R \in \mathsf{rel}_n \cap \Sigma$ and $(a_1, \ldots, a_n) \in \mathsf{dom}(\mathfrak{A})^n$, $(a_1, \ldots, a_n) \in R^{\mathfrak{A}}$ iff $(h(a_1), \ldots, h(a_n)) \in R^{\mathfrak{B}}$. h is an Σ-*isomorphism* if it is a surjective Σ-embedding. We then denote it as $(\mathfrak{A}, \mathbf{a}) \simeq_\Sigma (\mathfrak{B}, h(\mathbf{a}))$. h is an *elementary embedding* if, for all $n \geqslant 1$, φ constantless **FO**-formula and $(a_1, \ldots, a_n) \in \mathsf{dom}(\mathfrak{A})^n$, $\mathfrak{A} \models \varphi(a_1, \ldots, a_n)$ iff $\mathfrak{B} \models \varphi(h(a_1), \ldots, h(a_n))$. $\mathfrak{A}$ is an *extension* of $\mathfrak{B}$ if the identity map $id : \mathsf{dom}(\mathfrak{B}) \to \mathsf{dom}(\mathfrak{A})$ is a rel-embedding. $\mathfrak{A}$ is an *elementary extension* of $\mathfrak{B}$ if the identity map from $\mathfrak{B}$ to $\mathfrak{A}$ is an elementary embedding. $\mathfrak{A}$ and $\mathfrak{B}$ are *elementary equivalent* if, for all constantless **FO**-sentences φ, $\mathfrak{A} \models \varphi$ iff $\mathfrak{B} \models \varphi$.

Suppose Σ is a signature, $n \geq 1$, $\mathfrak{A}$ is a model, $\mathbf{a} = (a_1, \ldots, a_n) \in \mathsf{dom}(\mathfrak{A})^n$ and $A \subseteq \mathsf{dom}(\mathfrak{A})$. For any fragment $\mathcal{L}$ of **FO**, let $\mathcal{L}_A(\Sigma)$ denote the set of $\mathcal{L}(\Sigma^{+A})$-formulas where $\Sigma^{+A} = \Sigma \cup \{c_a : a \in \mathsf{dom}(\mathfrak{A})\}$ and where for all $a \in A$, c_a is assumed without loss of generality not belonging to cons and such that $c_a^{\mathfrak{A}} = a$. We define the $\mathcal{L}(\Sigma)$*-type of* $\mathbf{a}$ *in* $\mathfrak{A}$ *over* A as

$$\mathsf{tp}^{\mathfrak{A}}_{\mathcal{L},\Sigma}(\mathbf{a}/A) := \{\varphi(x_1, \ldots, x_n) \in \mathcal{L}_A(\Sigma) \mid \mathfrak{A} \models \varphi(\mathbf{a})\}.$$

We call $\mathcal{L}(\Sigma)$-n-type any set t such that $t = \mathsf{tp}^{\mathfrak{B}}_{\mathcal{L},\Sigma}(\mathbf{b}/B)$ for some model $\mathfrak{B}, B \subseteq \mathsf{dom}(\mathfrak{B})$ and $\mathbf{b} \in \mathsf{dom}(\mathfrak{B})^n$. We say t is *realized* in $\mathfrak{B}$ by $\mathbf{b}$ over B. We call $\mathcal{L}$-n*-type* any $\mathcal{L}(\Sigma)$-n-type for some Σ. An **FO**-n-type t is *consistent* with an **FO**-theory T if T has a model realizing t. Let $S_n(T)$ denote the set of all $\mathcal{L}$-n-types consistent with T. Let $T_A(\mathfrak{A})$ denote the set of all $\mathbf{FO}_A$-sentences satisfied by $\mathfrak{A}$. Let $S^{\mathfrak{A}}_n(A) = S_n(T_A(\mathfrak{A}))$. We say $\mathfrak{A}$ is ω*-saturated* if, for any $n \geqslant 1$, any finite subset $A \subseteq \mathsf{dom}(\mathfrak{A})$ and any **FO**-n-type $t \in S^{\mathfrak{A}}_n(A)$, t is realized in $\mathfrak{A}$.

We also make use of the following well-known property of **FO** (see *e.g.* [Mark02]).

1.7. Theorem. *Any model has an* ω*-saturated elementary extension.*

§ 1.2. KR TERMINOLOGY

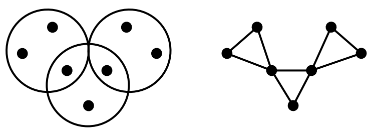
Our statement of the separability problem is expressed using Knowledge Representation terminology, which we now make precise in the most general context of **FO**.

1.8. *Definition.* Let $\mathcal{L}$ be a fragment of **FO**. We call $\mathcal{L}$*-ontology* any finite set of $\mathcal{L}$-sentences. We call *database* any finite set of formulas of the form $R(\mathbf{a})$ with $R(\mathbf{x}) \in \mathsf{rel}_n$ and $\mathbf{a} \in \mathsf{cons}^n$ for some $n \geqslant 1$. We call $\mathcal{L}$*-knowledge base* any pair consisting of an $\mathcal{L}$-ontology and a database. For a database $\mathcal{D}$, let $\mathsf{cons}(\mathcal{D})$ be the set of constants occurring in $\mathcal{D}$. For any $\mathcal{L}$-knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ and any model $\mathfrak{A}$ we write $\mathfrak{A} \models \mathcal{K}$ ("$\mathfrak{A}$ *satisfies* $\mathcal{K}$", "$\mathfrak{A}$ *is a model of* $\mathcal{K}$") if $\mathfrak{A} \models \mathcal{O}$ and $\mathfrak{A} \models \mathcal{D}$. We write $\mathcal{K} \models \varphi$ ("$\mathcal{K}$ *entails* φ") if $\mathfrak{A} \models \varphi$ for all $\mathfrak{A} \models \mathcal{K}$.

1.9. *Remark.* For any **FO**-knowledge base $(\mathcal{O}, \mathcal{D})$, the set $\mathcal{O} \cup \mathcal{D}$ is an **FO** theory (using constants). If constants are allowed in the ontology, the database is redundant thus, conversely, every **FO**-theory T can be seen as a knowledge base $(T, \emptyset)$. In a KR setting, the ontology usually expresses general knowledge about the world (*e.g.* "every X is a Y") and leaves every information concerning particular individuals to the database. If the ontology does not use constants, our definition of a knowledge base is a specific case of first-order theory in which all formulas containing constants need to be atomic.

1.10. *Model induced by a database.* Any database $\mathcal{D}$ can be seen as a model $\mathfrak{A}_\mathcal{D}$ defined by $\mathsf{dom}(\mathfrak{A}_\mathcal{D}) = \mathsf{cons}(\mathcal{D})$, $c^{\mathfrak{A}_\mathcal{D}} = c$ for all $c \in \mathsf{cons}(\mathcal{D})$ and $R^{\mathfrak{A}_\mathcal{D}} = \{\mathbf{c} \mid R(\mathbf{c}) \in \mathcal{D}\}$ for all $R \in \mathsf{rel}$.

1.11. *Definition.* For a database $\mathcal{D}$, $n \geqslant 1$ and $\mathbf{a} \in \mathsf{cons}(\mathcal{D})^n$, we denote by $\mathcal{D}_\mathbf{a}$ the "connected component of $\mathbf{a}$ in $\mathcal{D}$", that is, to be formally exact given our definitions, the set of all $R(\mathbf{b})$, $R \in \mathsf{rel}$, such that $\mathbf{b}$ is in the connected component of $\mathbf{a}$ in the Gaifman graph $\mathcal{G}_{\mathfrak{A}_\mathcal{D}}$ of the model $\mathfrak{A}_\mathcal{D}$ induced by $\mathcal{D}$, where the *Gaifman graph* $\mathcal{G}_\mathfrak{A}$ of a model $\mathfrak{A}$ is defined by $\mathcal{G}_\mathfrak{A} = (V, E)$ with $V = \mathsf{dom}(\mathfrak{A})$ and $E = \{(x, y) \mid x, y$ occur in some tuple $\mathbf{a} \in R^\mathfrak{A}$ for some $R \in \mathsf{rel}\}$. An example is given below.



$\mathfrak{A}$ $\qquad$ $\mathcal{G}_\mathfrak{A}$

§ 1.3. Conjunctive queries

Conjunctive queries and their unions happen to play a pivotal role in characterising separability and are themselves an interesting language for separation.

1.12. *Definition.* A *conjunctive query* is a formula of the form $q(\mathbf{x}) = \exists \mathbf{y} \varphi(\mathbf{x}, \mathbf{y})$, where $\varphi(\mathbf{x}, \mathbf{y})$ is a finite conjunction of constantless atomic **FO**-formulas. A *union of conjunctive queries* (UCQ) is a finite disjunction of conjunctive queries that all share the same free variables. We write (**U**)**CQ** for the language of (unions of) conjunctive queries. The free variables of a UCQ are often called *answer variables* in the Knowledge Representation or Description Logic literature [BHLS17].

1.13. *Model-database-CQ correspondence.* Let a *pointed database* be any pair $(\mathcal{D}, \mathbf{a})$ such that $\mathbf{a}$ is a tuple of constants occurring in $\mathcal{D}$. Any conjunctive query

$q(x_1,\ldots,x_n)$ can be seen as a pointed database $(\mathcal{D}_q,[x_1],\ldots,[x_n])$, by defining $\mathcal{D}_q$ to be the set of all $R([y_1],\ldots,[y'_m])$ such that there exist $y'_1 \in [y_1],\ldots,y'_m \in [y_m]$ with $R(y'_1,\ldots,y'_m) \in \mathcal{D}$, where $[\cdot]$ denotes the equivalence class induced by the smallest equivalence relation over the variables of q that contains all pairs (x,y) such that $(x=y)$ is a conjunct of q. We can assume $[y'_1],\ldots,[y'_m] \in \mathsf{cons}$ without loss of generality as cons is infinite. Any conjunctive query can therefore be seen as a model $\mathfrak{A}_q$, by setting $\mathfrak{A}_q = \mathfrak{A}_{\mathcal{D}_q}$, where $\mathfrak{A}_{\mathcal{D}_q}$ is the model induced by the database $\mathcal{D}_q$, as defined in Remark 1.10. The *Gaifman graph* G_q of a conjunctive query q can be defined as the Gaifman graph of the model $\mathfrak{A}_q$ induced by q. Conversely, any finite pointed model $(\mathfrak{A},\mathbf{a})$ with $\mathbf{a}=(a_1,\ldots,a_n)$ can be seen as the CQ

$$q(\mathfrak{A},\mathbf{a})(x_1,\ldots,x_n) = \exists_{y\in\mathsf{dom}(\mathfrak{A})\setminus\{a_1,\ldots,a_n\}}\, y \bigwedge_{\substack{R\in\mathsf{rel}\\ \mathbf{y}\in R^{\mathfrak{A}}}} R(\mathbf{y})[a_1/x_1,\ldots,a_n/x_n] \wedge \bigwedge_{a_i=a_j} x_i = x_j$$

(assuming without loss of generality that $x_1,\ldots,x_n \notin \mathsf{dom}(\mathfrak{A})$), thus any pointed database $(\mathcal{D},\mathbf{a})$ can be seen as a CQ $q(\mathcal{D},\mathbf{a}) := q(\mathfrak{A}_{\mathcal{D}},\mathbf{a})$.

1.14. *Definition.* Call a CQ q *rooted* if every variable is reachable from an answer variable in G_q.

1.15. *Remark.* As conjunctive queries can be seen as models, satisfying a CQ for a pointed model $(\mathfrak{A},\mathbf{a})$ is the same as being able to homomorphically embed the (model induced by the) query into that model, matching the answer variables with $\mathbf{a}$. Let $q(x_1,\ldots,x_n)$ and $\mathbf{a}=(a_1,\ldots,a_n) \in \mathsf{dom}(\mathfrak{A})^n$ for some model $\mathfrak{A}$. Then $\mathfrak{A} \models q(\mathbf{a})$ iff there is a homomorphism $h:\mathfrak{A}_q \to \mathfrak{A}$ with $h(x_i)=a_i$ for all i.

1.16. *Definition.* Let $\mathcal{L}$ be a fragment of **FO**. We call *UCQ evaluation on $\mathcal{L}$-knowledge bases* the decision problem associated with the set of all triples $(\mathcal{K},q,\mathbf{a})$ such that $\mathcal{K}=(\mathcal{O},\mathcal{D})$ is an $\mathcal{L}$-knowledge base, $q(\mathbf{x})$ is a UCQ, $\mathbf{a} \in \mathsf{cons}(\mathcal{D})^{|\mathbf{x}|}$, and $\mathcal{K} \models q(\mathbf{a})$. Similarly, we define CQ evaluation and rooted UCQ evaluation on $\mathcal{L}$-knowledge bases.

(Rooted) (U)CQ-evaluation on empty ontologies is NP-complete. By Remark 1.15, it follows from NP-completeness of the Graph Homomorphism Problem [BHLS17]. On the other hand, undecidability of (rooted) (U)CQ-evaluation on **FO**-knowledge bases follows from undecidability of satisfiability in **FO**.

1.17. *Definition.* Let $\mathcal{L}$ be a fragment of **FO**. Evaluating queries from a query language $\mathcal{Q}$ contained in **FO** is *finitely controllable* on $\mathcal{L}$-knowledge bases if for every $\mathcal{L}$-ontology $\mathcal{O}$, database $\mathcal{D}$, formula $\varphi(\mathbf{x})$ in $\mathcal{Q}$, tuple of constants $\mathbf{a} \in \mathsf{cons}(\mathcal{D})^{|\mathbf{x}|}$, if $(\mathcal{O}, \mathcal{D}) \not\models \varphi(\mathbf{a})$, then there is a finite model $\mathfrak{A}$ of $\mathcal{K}$ such that $\mathfrak{A} \not\models \varphi(\mathbf{a}^{\mathfrak{A}})$ [JK84, Ro11].

Note that $\mathcal{L}$ has the finite model property (for all φ in $\mathcal{L}$, $\emptyset \not\models \varphi$ iff $\mathfrak{A} \not\models \varphi$ for some finite model $\mathfrak{A}$) if evaluating queries from $\mathcal{L}$ is finitely controllable on $\mathcal{L}$-knowledge bases. Conversely, the finite model property does not always imply finite controllability of evaluating (rooted) CQs or UCQs, but it does for $\mathcal{ALCI}$ and even **GF** [BGO14].

§ 1.4. Description logics

Description logics are a family of languages originally designed and popularized in the context of Knowledge Representation, for their good computational behaviour and easy readability by engineers unfamiliar with logical syntax. Those two qualities also make them interesting candidates as separating languages, from an applied standpoint. Their model-theoretic and computational properties have been extensively studied in the last forty years, see [BHLS17, BCMNP03] for standard textbooks.

1.4.1. *Syntax & Semantics*

We introduce the foundational description logic $\mathcal{ALC}$ and its negation-free fragment $\mathcal{EL}$.

1.18. *Definition.* Let the languages $\mathcal{ALC}$ and $\mathcal{EL}$ be defined by the following grammars, for all $A \in \mathsf{rel}_1, R \in \mathsf{rel}_2$.

$$C, D ::= \begin{cases} A \mid C \sqcap D \mid \neg C \mid \exists R.C & \mathcal{ALC} \\ \top \mid A \mid C \sqcap D \mid \exists R.C & \mathcal{EL} \end{cases}$$

Let $\mathcal{L}$ be a description logic inductively defined over constructors and symbols from rel_1 and $R \in \mathsf{rel}_2$. We define

- $\mathcal{LI}$, by extending rel_2 to the closure of rel_2 under $R \mapsto R^-$ (*inverse roles*)
- $\mathcal{LO}$, by extending rel_1 to $\mathsf{rel}_1 \cup \{\{c\} : c \in \mathsf{cons}\}$ (*nominals*)
- $\mathcal{LQ}$, by adding "$C \in \mathcal{LQ}$ implies $\geqslant nr.C \in \mathcal{LQ}$" (*number restrictions*) for all $n \in \mathbb{N}, R \in \mathsf{rel}_2$.

If $\mathcal{L}$ is a description logic, we call its elements *$\mathcal{L}$-concepts*. If $C \in \mathcal{L}$, let $\mathsf{sig}(C)$ denote the set of symbols from $\mathsf{rel} \cup \mathsf{cons}$ occurring in C. If $\mathsf{sig}(C) = \Sigma$, we say C is an *$\mathcal{L}(\Sigma)$-concept*. In Description Logic terminology, unary relations in rel_1 are usually called *concept names* and binary relations in rel_2 *role names*. We write rel_2^- for $\mathsf{rel}_2 \cup \{R^- \mid R \in \mathsf{rel}_2\}$.

1.19. *Remark.* The order in which suffixes $\mathcal{I}, \mathcal{O}, \mathcal{Q}$ are positioned is purely conventional. While outside of our scope, many other suffixes and extensions exist in the DL literature; see [BHLS17].

We use the following standard abbreviations.

$\top$	$C \sqcup \neg C$
$\bot$	$\neg\top$
$C \sqcup D$	$\neg(\neg C \sqcap \neg D)$
$C \rightarrow D$	$\neg C \sqcup D$
$\forall R.C$	$\neg\exists R.\neg C$
R	$(R^-)^-$
$\leqslant nR.C$	$\neg \geqslant (n+1)R.C$
$= nR.C$	$\leqslant nR.C \sqcap \geqslant nR.C$

1.20. *Definition.* Let $\mathcal{L}$ be a description logic. An *$\mathcal{L}$-ontology* is a set of expressions of the form $C \sqsubseteq D$ (*concept inclusion*) where C, D are $\mathcal{L}$-concepts. An *$\mathcal{L}$-database* is a set of expressions of the form $R(a, b)$ (*role assertion*) or $A(a)$ (*concept assertion*) for some role name $R \in \mathsf{rel}_2$ and concept name $A \in \mathsf{rel}_1$. An *$\mathcal{L}$-knowledge base* is a pair $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ consisting of an $\mathcal{L}$-ontology and an $\mathcal{L}$-database. We write $\mathsf{sub}(\mathcal{K})$ to denote the closure under subconcepts and single negation of the set of concepts occurring in $\mathcal{K}$, and $\mathsf{sub}(\mathcal{O})$ analogously. For any pointed model $(\mathfrak{A}, x)$, let $\mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, x) = \{C \in \mathsf{sub}(\mathcal{K}) \mid x \in C^{\mathfrak{A}}\}$ and similarly for $\mathcal{O}$. Call *$\mathcal{K}$-type* (resp. $\mathcal{O}$-type) any t such that $t = \mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, x)$ (resp. $\mathcal{O}$) for some $(\mathfrak{A}, x)$.

Unlike in the traditional definition of databases (also called ABoxes) in Description Logic, we do not consider complex concepts in databases but only atomic ones, unless specified otherwise. For instance, a database containing $(\exists R.A)(a)$ is not allowed in our framework.

1.21. *DLs as fragments of* **FO**. The description logics defined above can be seen as fragments of **FO**: for each $\mathcal{L}$, every $\mathcal{L}$-concept, concept inclusion or role/concept assertion can be translated into first-order via the function $(\cdot)^\dagger$, defined below for any $C, D \in \mathcal{L}$, $A \in \mathsf{rel}_1, R \in \mathsf{rel}_2, c \in \mathsf{cons}$ and $n \in \mathbb{N}$.

$$\begin{aligned}
A^\dagger &= A(x) \\
\{c\}^\dagger &= (x = c) \\
(C \sqcap D)^\dagger &= (C^\dagger(x) \wedge D^\dagger(x)) \\
(\exists R.C)^\dagger &= \exists y (R(x,y) \wedge C^\dagger(y)) \\
(\exists R^-.C)^\dagger &= \exists y (R(y,x) \wedge C^\dagger(y)) \\
(\geqslant nR.C)^\dagger &= \exists y_1 \ldots \exists y_n \Big(\bigwedge_{i \neq j} \neg(y_i = y_j) \wedge \bigwedge_i (R(x, y_i) \wedge C^\dagger(y_i)) \Big) \\
(C \sqsubseteq D)^\dagger &= \forall x (C^\dagger(x) \rightarrow D^\dagger(x)) \\
(C(c))^\dagger &= C^\dagger(c) \\
(R(a,b))^\dagger &= R(a,b)
\end{aligned}$$

We then interchangeably use the term of "$\mathcal{L}$-formulas" when speaking of $\mathcal{L}$-concepts, for any description logic $\mathcal{L}$.

Thanks to the translatability of description logic concepts, we can use first-order semantics to interpret Description Logic concepts in relational structures, as first-order formulas. The definitions of $\mathcal{L}$-ontologies, databases and knowledge bases in the Description Logic context are then also consistent with their definitions in the first-order context.

1.22. *Definition.* For any $\mathcal{L}$-concept C and model $\mathfrak{A}$, let $C^{\mathfrak{A}} = \{x \in \mathsf{dom}(\mathfrak{A}) \mid \mathfrak{A} \models C^\dagger(x)\}$. We write the following for any $C, D \in \mathcal{ALCQIO}$, any pointed model $(\mathfrak{A}, a)$ and $c \in \mathsf{cons}$.

$\mathfrak{A} \models C(a)$	if $a \in C^{\mathfrak{A}}$
$\mathfrak{A} \models C(c)$	if $c^{\mathfrak{A}} \in C^{\mathfrak{A}}$
$\mathfrak{A} \models C \sqsubseteq D$	if $C^{\mathfrak{A}} \subseteq D^{\mathfrak{A}}$

Let $\mathsf{tp}_{\mathcal{L},\Sigma}(\mathfrak{A}, x) = \{C \text{ in } \mathcal{L}(\Sigma) \mid x \in C^{\mathfrak{A}}\}$. and $\mathcal{L}(\Sigma)$-types denote any set of the aforementioned form for some pointed model $(\mathfrak{A}, x)$.

1.23. *DLs as syntactic variants of modal logics.* While DLs and modal logics –see [BRV01] for an overview– were developed independently, it was first pointed out in [Sc91] that every $\mathcal{ALC}$-concept can be translated into a multi-modal formula via the function $(\cdot)^\dagger$, defined below for all concepts C, D, role names R and concept names A.

$$A^\dagger = A$$
$$(C \sqcap D)^\dagger = (C^\dagger \wedge D^\dagger)$$
$$(\exists R.C)^\dagger = \Diamond_r C^\dagger$$
$$(\neg C)^\dagger = \neg C^\dagger$$

1.4.2. *Model theory*

1.24. *Definition.* Given a signature Σ, an $\mathcal{ALC}(\Sigma)$-bisimulation is a binary relation $B \subseteq \mathsf{dom}(\mathfrak{A}) \times \mathsf{dom}(\mathfrak{B})$ for two models $\mathfrak{A}, \mathfrak{B}$ satisfying the following conditions, for all $(x, y) \in B$, all concept names, role names and constants $A, R, c \in \Sigma$.

ATOM $x \in A^\mathfrak{A}$ iff $y \in A^\mathfrak{B}$.

FORTH If $(x, x') \in R^\mathfrak{A}$ for some x', then there exists y' such that $(y, y') \in R^\mathfrak{B}$ and $(x', y') \in B$.

BACK If $(y, y') \in R^\mathfrak{A}$ for some y', then there exists x' such that $(x, x') \in R^\mathfrak{B}$ and $(x', y') \in B$.

To define $\mathcal{LI}$-bisimulations from $\mathcal{L}$-bisimulations, let $R \in \Sigma \cup \{S^- \mid S \in \Sigma\}$.

To define $\mathcal{LO}$-bisimulations from $\mathcal{L}$-bisimulations, add

OATOM $x = c^\mathfrak{A}$ iff $y = c^\mathfrak{B}$.

To define $\mathcal{LQ}$-bisimulations from $\mathcal{L}$-bisimulations, replace FORTH, BACK by

QFORTH For any finite $X \subseteq \{x' : (x, x') \in R^\mathfrak{A}\}$, B contains a bijection $X \to Y$ for some $Y \subseteq \{y' : (y, y') \in R^\mathfrak{B}\}$.

QBACK For any finite $Y \subseteq \{y' : (y, y') \in R^\mathfrak{B}\}$, B contains a bijection $X \to Y$ for some $X \subseteq \{x' : (x, x') \in R^\mathfrak{A}\}$.

If B is an $\mathcal{L}(\Sigma)$-bisimulation for some $\mathcal{L}$ between two pointed models $(\mathfrak{A}, a)$ and $(\mathfrak{B}, b)$, we write $(\mathfrak{A}, a) \sim_{\mathcal{L},\Sigma} (\mathfrak{B}, b)$ and say that $(\mathfrak{A}, a)$ and $(\mathfrak{B}, b)$ are $\mathcal{L}(\Sigma)$-*bisimilar.* We say that $(\mathfrak{A}, a)$ and $(\mathfrak{B}, b)$ are *logically* $\mathcal{L}(\Sigma)$*-equivalent*, which we write $(\mathfrak{A}, a) \equiv_{\mathcal{L},\Sigma} (\mathfrak{B}, b)$, if $\mathfrak{A} \models C(a) \Leftrightarrow \mathfrak{B} \models C(b)$ for all $\mathcal{L}(\Sigma)$-concepts C.

The following is well-known in modal and description logics. It is usually only proved for $\mathcal{ALC}$; we quickly show that it can be extended to any extension within $\mathbf{DL}_{\mathcal{IOQ}}$.

1.25. Lemma. *The following items hold for any* $\mathcal{L} \in \mathbf{DL}_{\mathcal{IOQ}}$*, pointed models* $(\mathfrak{A}, d), (\mathfrak{B}, e)$*, and signature* Σ.

1. $(\mathfrak{A}, d) \sim_{\mathcal{L},\Sigma} (\mathfrak{B}, e) \Rightarrow (\mathfrak{A}, d) \equiv_{\mathcal{L},\Sigma} (\mathfrak{B}, e)$.
2. $(\mathfrak{A}, d) \sim_{\mathcal{L},\Sigma} (\mathfrak{B}, e) \Leftrightarrow (\mathfrak{A}, d) \equiv_{\mathcal{L},\Sigma} (\mathfrak{B}, e)$ *if* $\mathfrak{A}$ *has finite outdegree.*

3. $(\mathfrak{A}, d) \sim_{\mathcal{L},\Sigma} (\mathfrak{B}, e) \Leftrightarrow (\mathfrak{A}, d) \equiv_{\mathcal{L},\Sigma} (\mathfrak{B}, e)$ *if* $\mathfrak{A}$ *and* $\mathfrak{B}$ *are* ω*-saturated.*

Proof. We prove it for $\mathcal{L} = \mathcal{ALCQIO}$. Arguments for the less expressive languages are easily derived from the $\mathcal{ALCQIO}$ proof. (1) is straightforward. To prove the converse implication, suppose $(\mathfrak{A}, d) \equiv_{\mathcal{L},\Sigma} (\mathfrak{B}, e)$. We show that the relation $\equiv_{\mathcal{L},\Sigma}$ is itself the desired bisimulation if $\mathfrak{A}$ has finite outdegree or if $\mathfrak{A}, \mathfrak{B}$ are ω-saturated. Let $(x, y) \in \equiv_{\mathcal{L},\Sigma}$. Let $R \in \Sigma \cap \mathsf{rel}_2^-$. ATOM and OATOM are trivially satisfied. For all x, y, $(\mathfrak{A}, x) \equiv_{\mathcal{L},\Sigma} (\mathfrak{B}, y)$ iff $\mathsf{tp}^{\mathfrak{B}}_{\mathcal{L},\Sigma}(x) = \mathsf{tp}^{\mathfrak{B}}_{\mathcal{L},\Sigma}(y)$. It is clear that QFORTH and QBACK immediately follow if for every $\mathcal{L}(\Sigma)$-type t, x and y have equally many R-successors of type t. It only remains to prove the latter when either $\mathfrak{A}$ has finite outdegree or $\mathfrak{A}, \mathfrak{B}$ are ω-saturated. Let t be an arbitrary $\mathcal{L}(\Sigma)$-type and n_x, n_y denote the respective numbers of R-successors of x and y of type t.

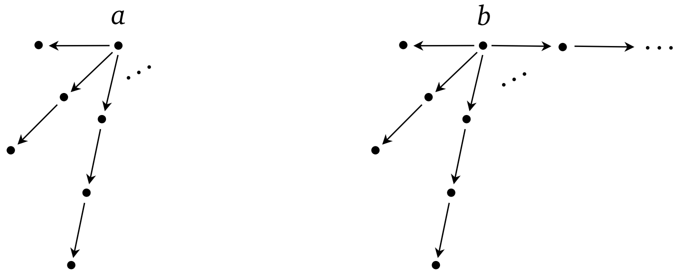
(**2**) Suppose $\mathfrak{A}$ has finite outdegree and let n be the number of R-successors of x in $\mathfrak{A}$. Then, $\mathfrak{A} \models (= nR.\top)(x)$. Then, $\mathfrak{B} \models (= nr.\top)(y)$ by logical equivalence, so y also has n R-successors. For each R-successor x' of type $t' \neq t$, there is an $\mathcal{L}(\Sigma)$-concept $C_{x'} \in t' \setminus t$. Let I denote the set of such successors. Since $\mathfrak{A}$ has finite outdegree, I is finite, so we can write $\mathfrak{A} \models = (n - n_x)R.\bigsqcup_{x' \in I} C_{x'}(x)$ thus $\mathfrak{B} \models = (n - n_x)R.\bigsqcup_{x' \in I} C_{x'}(y)$ by logical equivalence. Then, y has at least $n - n_x$ R-successors of different type from t. To show y has, in fact, exactly $n - n_x$ such successors, suppose it has more and denote their set by J. Again, each $y' \in J$ satisfies a concept $C_{y'} \notin t$, thus $\mathfrak{B} \models = (n - n_y)R.\bigsqcup_{y' \in J} C_{y'}(y)$, and $\mathfrak{A} \models = (n - n_y)R.\bigsqcup_{y' \in J} C_{y'}(x)$ by logical equivalence. Then x has at least $n - n_y$ R-successors of different type from t, so $n_x = n_y$.

(**3**) Suppose $\mathfrak{A}$ and $\mathfrak{B}$ are ω-saturated. Suppose n_x is finite. The $\mathbf{FO}(\Sigma)$-n_x-type $t'(z_1, \dots, z_{n_x}) = \bigcup_{1 \leqslant i \leqslant n_x} \{\varphi(z_i) \wedge R(y, z_i) \mid \varphi \in t\}$ over the finite set $\{y\}$ is realized in $\mathfrak{B}$. Then, $n_x \leqslant n_y$. That set is indeed a type as it is finitely satisfiable: for any $\varphi_1, \dots, \varphi_m \in t$, $\mathfrak{A} \models \exists z_1 \dots \exists z_{n_x} \bigwedge_{1 \leqslant i \leqslant n_x} \bigwedge_{1 \leqslant j \leqslant m} \varphi_j(z_i) \wedge R(x, z_i)$, therefore $\mathfrak{B} \models \exists z_1 \dots \exists z_{n_x} \bigwedge_{1 \leqslant i \leqslant n_x} \bigwedge_{1 \leqslant j \leqslant m} \varphi_j(z_i) \wedge R(y, z_i)$ since we assumed $(\mathfrak{A}, x) \equiv_{\mathcal{L},\Sigma} (\mathfrak{B}, y)$. If n_x is infinite, then $t'(z_1, \dots, z_n)$, which is a type by the same argument as above, is realized in $\mathfrak{B}$ for any $n \geqslant 1$, so n_y is infinite. It only remains to show that $n_y \leqslant n_x$ if n_x is finite. If n_y was infinite, the last argument above would imply that n_x is infinite. Then n_y is finite, so we can apply to n_y the argument we applied to n_x in the beginning and obtain $n_y \leqslant n_x$. $\dashv$

1.26. *Example.* This example shows the converse of (1) in Lemma 1.25 does not hold with arbitrary models. It is well-known in classical modal logic ($\mathcal{ALC}$ syntax) [BRV01] and easily extendable to any $\mathcal{L} \in \mathbf{DL}_{\mathcal{IOQ}}$. Let $\mathfrak{A}$ and $\mathfrak{B}$ (figure below) be defined as follows.

$$\begin{aligned}
\mathsf{dom}(\mathfrak{A}) &= \{a\} \cup \bigcup_{j \geqslant 1} \{a_{i,j} : 1 \leqslant i \leqslant j\} \\
\mathsf{dom}(\mathfrak{B}) &= \{b\} \cup \bigcup_{j \geqslant 1} \{b_{i,j} : 1 \leqslant i \leqslant j\} \cup \{b_{i,\infty} : i \geqslant 1\} \\
R^{\mathfrak{A}} &= \{(a, a_{1,j}) : j \geqslant 1\} \cup \{(a_{i,j}, a_{i+1,j}) : j \geqslant 1\} \\
R^{\mathfrak{B}} &= \{(b, b_{1,j}) : j \geqslant 1\} \cup \{(b_{i,j}, b_{i+1,j}) : j \geqslant 1 \text{ or } j = \infty\} \\
a = c^{\mathfrak{A}}, b &= c^{\mathfrak{B}} \qquad \text{for all } c \in \mathsf{cons}
\end{aligned}$$

Then with $\Sigma := \{R\}$ we have $(\mathfrak{A}, a) \equiv_{\mathcal{ALCIOQ},\Sigma} (\mathfrak{B}, b)$ but $(\mathfrak{A}, a) \not\sim_{\mathcal{ALC},\Sigma} (\mathfrak{B}, b)$.

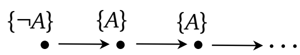


On top of having one R-successor chain of length n for every $n \in \mathbb{N}$, b also has an infinite R-successor chain.

1.27. *Definition.* $\mathcal{L}$ has the *finite model property* if for any $\mathcal{L}$-concept C and ontology $\mathcal{O}$, whenever there exists $\mathfrak{A} \models \mathcal{O}$ such that $C^{\mathfrak{A}} \neq \emptyset$, there exists $\mathfrak{B} \models \mathcal{O}$ finite such that $C^{\mathfrak{B}} \neq \emptyset$.

1.28. Theorem ([BHLS17]). *$\mathcal{ALCI}$ and $\mathcal{ALCQO}$ have the finite model property. $\mathcal{ALCQI}$ does not.*

Proof. The proof for $\mathcal{ALCI}$ and $\mathcal{ALCQO}$ is based on a standard filtration argument (*e.g.* [BRV01]). To see that $\mathcal{ALCQI}$ does not have the finite model property, consider $C = \neg A \sqcap \exists R.A$ and $\mathcal{O} = \{A \sqsubseteq \exists R.A, \top \sqsubseteq\ \leqslant 1R^-.\top\}$. Then for any model $\mathfrak{A}$ and $x \in C^{\mathfrak{A}}$, there is an infinite R-chain starting at x. ⊣



1.29. *Definition.* To each model $\mathfrak{A}$ we associate a directed graph $G^d_{\mathfrak{A}} = (\mathsf{dom}(\mathfrak{A}), \bigcup_{R \in \mathsf{rel}_2} R^{\mathfrak{A}})$. We denote its undirected counterpart by $G^u_{\mathfrak{A}}$.

If $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCO}, \mathcal{ALCQ}, \mathcal{ALCQO}\}$, we say

1. $\mathfrak{A}$ has *finite $\mathcal{L}$-outdegree* if $G^d_{\mathfrak{A}}$ has finite outdegree.
2. $\mathfrak{A}$ is *$\mathcal{L}$-rooted in a* if every node in $\mathfrak{A}$ is reachable from $a^{\mathfrak{A}}$ in $G^d_{\mathfrak{A}}$.

3. $\mathfrak{A}$ is an *$\mathcal{L}$-tree* if $G^d_{\mathfrak{A}}$ is acyclic, has maximum indegree 1 and $R_0^{\mathfrak{A}} \cap R_1^{\mathfrak{A}} = \emptyset$ for all distinct $R_0, R_1 \in \mathsf{rel}_2$.

If $\mathcal{L} \in \{\mathcal{ALCI}, \mathcal{ALCIO}, \mathcal{ALCQI}\}$, we say

1. $\mathfrak{A}$ has *finite $\mathcal{L}$-outdegree* if $G^u_{\mathfrak{A}}$ has finite outdegree.
2. $\mathfrak{A}$ is *$\mathcal{L}$-rooted in a* if every node in $\mathfrak{A}$ is reachable from $a^{\mathfrak{A}}$ in $G^u_{\mathfrak{A}}$.
3. $\mathfrak{A}$ is an *$\mathcal{L}$-tree* if $G^u_{\mathfrak{A}}$ is acyclic and $R_0^{\mathfrak{A}} \cap R_1^{\mathfrak{A}} = \emptyset$ for all distinct $R_0, R_1 \in \mathsf{rel}_2$.

1.30. The definition of $\mathcal{L}$-tree only depends on $\mathcal{L}$ using inverse roles or not, so any $\mathcal{L}$-tree is either an $\mathcal{ALC}$-tree or an $\mathcal{ALCI}$-tree. Any $\mathcal{ALC}$-tree is an $\mathcal{ALCI}$-tree and has finite $\mathcal{ALC}$-outdegree iff it has finite $\mathcal{ALCI}$-outdegree.

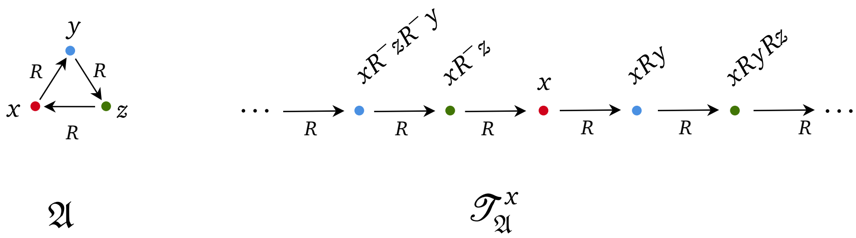
1.31. *Definition.* We define the set $\mathsf{path}(G^u_{\mathfrak{A}})$ of all *undirected paths* in $\mathfrak{A}$ as the set of all $x_0 R_1 x_1 R_2 \dots R_n x_n$, $n \geqslant 0$, such that

$$
\begin{aligned}
&x_i \in \mathsf{dom}(\mathfrak{A}) && \text{for all } i \in \{0, \dots, n\},\\
&R_i \in \mathsf{rel}_2^- && \text{for all } i \in \{1, \dots, n\},\\
&(x_i, x_{i+1}) \in R_i^{\mathfrak{A}} && \text{for all } i \in \{0, \dots, n-1\},\\
&R_i = R_{i+1}^- \Rightarrow x_{i-1} \neq x_{i+1} && \text{for all } i \in \{1, \dots, n-1\}.
\end{aligned}
$$

Let $\mathsf{head}(p) = x_0$ and $\mathsf{tail}(p) = x_n$.

1.32. *Definition.* Let $(\mathfrak{A}, x)$ be a pointed model. The *tree unfolding* $\mathscr{T}^x_{\mathfrak{A}}$ of x in $\mathfrak{A}$ is the model defined as follows.

$$
\begin{aligned}
\mathsf{dom}(\mathscr{T}^x_{\mathfrak{A}}) &= \{p \in \mathsf{path}(G^u_{\mathfrak{A}}) \mid \mathsf{head}(p) = x\}\\
R^{\mathscr{T}^x_{\mathfrak{A}}} &= \{(p, p') \in \mathsf{dom}(\mathscr{T}^x_{\mathfrak{A}})^2 \mid \exists x\ (p' = pRx \vee p = p'R^- x)\} && \text{for all } R \in \mathsf{rel}_2\\
A^{\mathscr{T}^x_{\mathfrak{A}}} &= \{p \in \mathsf{dom}(\mathscr{T}^x_{\mathfrak{A}}) \mid \mathsf{tail}(p) \in A^{\mathfrak{A}}\} && \text{for all } A \in \mathsf{rel}_1
\end{aligned}
$$



Tree unfolding of some pointed model $(\mathfrak{A}, x)$*.*

A routine check shows that unfolding $\mathfrak{A}$ at x gives an $\mathcal{ALCI}$-tree and preserves the truth of $\mathcal{ALCQI}$-concepts.

1.33. Lemma. *For any pointed model* $(\mathfrak{A}, x)$ *and any* $\mathcal{L} \in \mathbf{DL}_{\mathcal{IQ}}$*,*

1. *$\mathscr{T}^x_{\mathfrak{A}}$ is an $\mathcal{ALCI}$-tree,*
2. *the directedly connected component of x in $\mathscr{T}^x_{\mathfrak{A}}$ is an $\mathcal{ALC}$-tree,*
3. *the relation $\{(p, \mathsf{tail}(p)) \mid p \in \mathsf{dom}(\mathscr{T}^x_{\mathfrak{A}})\}$ witnesses $(\mathfrak{A}, x) \sim_{\mathcal{ALCQI}} (\mathscr{T}^x_{\mathfrak{A}}, x)$.*

1.34. *Remark.* If $\mathcal{L}$ extends $\mathcal{ALCQ}$, it is necessary to obtain an $\mathcal{L}$-bisimulation in Lemma 1.33 that paths cannot 'turn back' (fourth condition in Def. 1.31). Otherwise, for any $(x, y) \in R^{\mathfrak{A}}$, it follows that, in the tree unfolding, xRy has at least two R^--successors (x and $xRyR^-x$) while y may only have one (x) in $\mathfrak{A}$. If $\mathcal{L}$ does not extend $\mathcal{ALCQ}$, whether we add that requirement or not does not make any difference with respect to bisimulations; we thus add it everywhere for uniformity.

1.35. *Definition.* $\mathcal{L}$ has the *tree model property* if, for any $\mathcal{L}$-concept C and $\mathcal{L}$-ontology $\mathcal{O}$, whenever there exists $\mathfrak{A} \models \mathcal{O}$ such that $C^{\mathfrak{A}} \neq \emptyset$, there exists an $\mathcal{L}$-tree model $\mathfrak{B} \models \mathcal{K}$ such that $C^{\mathfrak{B}} \neq \emptyset$.

1.36. Theorem. *Every $\mathcal{L} \in \mathbf{DL}_{\mathcal{IQ}}$ has the tree model property.*

Proof. Let $\mathcal{L} \in \mathbf{DL}_{\mathcal{IQ}}$. Let $\mathcal{O}$ and C be such that there exists $\mathfrak{A} \models \mathcal{O}$ with $C^{\mathfrak{A}} \neq \emptyset$. Let $x \in C^{\mathfrak{A}}$. Then let $\mathfrak{B}$ be the tree unfolding $\mathscr{T}^x_{\mathfrak{A}}$ of $\mathfrak{A}$ at x if $\mathcal{L} \in \{\mathcal{ALCQI}, \mathcal{ALCI}\}$ (resp. its directedly connected component if $\mathcal{L} \in \{\mathcal{ALCQ}, \mathcal{ALC}\}$), which is an $\mathcal{L}$-tree by Lemma 1.33. By the same Lemma, x is $\mathcal{ALCQI}$-bisimilar (thus $\mathcal{L}$-bisimilar) to the path consisting of the single point x. That makes them logically equivalent with respect to $\mathcal{L}$ (Lem. 1.25), so $x \in C^{\mathfrak{A}}$ implies $x \in C^{\mathscr{T}^x_{\mathfrak{A}}}$. ⊣

1.37. Corollary. *For $\mathcal{L} \in \mathbf{DL}_{\mathcal{IQ}}$ the tree model property is witnessed by an $\mathcal{L}$-tree of finite outdegree.*

Proof. It suffices to show there exists a submodel $(\mathscr{T}^x_{\mathfrak{A}})_{\mathsf{fo}}$ of $\mathscr{T}^x_{\mathfrak{A}}$ that contains x and has finite outdegree, such that $(\mathscr{T}^x_{\mathfrak{A}})_{\mathsf{fo}} \models \mathcal{O}$ and $(\mathscr{T}^x_{\mathfrak{A}})_{\mathsf{fo}} \models C(x)$. Let $\mathfrak{B}_0$ be the submodel of $\mathscr{T}^x_{\mathfrak{A}}$ induced by $\{x\}$. For any pointed model $(\mathfrak{B}, x)$ and $R \in \mathsf{rel}^-_2$ we write $R^{\mathfrak{B}}[x] := \{y \in \mathsf{dom}(\mathfrak{B}) \mid (x, y) \in R^{\mathfrak{B}}\}$. Assume w.l.o.g. that $\mathcal{O}$ is in negation normal form and that $\geqslant 1R.D$ is replaced by $\exists R.D$ for any concept D and relation R. We define $\mathfrak{B}_{i+1}$ from $\mathfrak{B}_i$ for all $i \geq 0$ as follows.

```
for all d ∈ dom(𝔅_i):
  for all concepts of the form (⩾ nR.D) occurring in 𝒪 ∪ {C}:
    while |R^{𝔅_{i+1}}[d] ∩ dom(𝔅_{i+1}) ∩ C^{𝒯^x_𝔄}| < n:
      pick one e ∈ R^{𝒯^x_𝔄}[d] ∩ C^{𝒯^x_𝔄} − dom(𝔅_{i+1})
      𝔅_{i+1} ← submodel of 𝒯^x_𝔄 induced by dom(𝔅_{i+1}) ∪ {e}
```

Then, let $(\mathscr{T}^x_{\mathfrak{A}})_{\mathsf{fo}} = \bigcup_{i \geq 0} \mathfrak{B}_i$. A quick inductive argument shows that $(\mathscr{T}^x_{\mathfrak{A}})_{\mathsf{fo}} \models D(d)$ iff $(\mathscr{T}^x_{\mathfrak{A}})_{\mathsf{fo}} \models D(d)$ for all $d \in \mathsf{dom}((\mathscr{T}^x_{\mathfrak{A}})_{\mathsf{fo}})$ and all D occurring in $\mathcal{O} \cup \{C\}$, which implies that $(\mathscr{T}^x_{\mathfrak{A}})_{\mathsf{fo}} \models \mathcal{O}$ and $(\mathscr{T}^x_{\mathfrak{A}})_{\mathsf{fo}} \models C(x)$. ⊣

1.38. $\mathcal{ALCO}$ does not have the tree model property: it is easy to create an $\mathcal{ALCO}$-concept that forces a cycle. For instance, $\mathcal{O} = \emptyset$ and $C = \{a\} \to \exists R.\{a\}$.

For the unfolding to preserve satisfaction of the database (instead of just the ontology), one needs to "unfold at each constant". Instead of a tree, we obtain a union of trees whose roots (the database individuals) are connected as the database requires them to be. We can also adapt that notion to logics containing nominals by allowing cycles from constants to themselves as long as the rest of the structure is tree-shaped.

1.39. *Definition.* Let $\mathfrak{A}$ be a model of some database $\mathcal{D}$ and $\mathcal{L} \in \mathbf{DL}_{\mathcal{IOQ}}$. We define the *$\mathcal{D}_\mathcal{L}$-forest unfolding* $\mathscr{F}^{\mathcal{D}}_{\mathfrak{A},\mathcal{L}}$ as follows. If $\mathcal{L}$ does not contain nominals, $\mathscr{F}^{\mathcal{D}}_{\mathfrak{A},\mathcal{L}}$ is the union of all tree unfoldings $\mathscr{T}^{c^{\mathfrak{A}}}_{\mathfrak{A}}$ for $c \in \mathsf{cons}(\mathcal{D})$, with their roots connected by $R(c^{\mathfrak{A}}, d^{\mathfrak{A}})$ whenever $R(c,d) \in \mathcal{D}$, except that for each $R(c^{\mathfrak{A}}, d^{\mathfrak{A}})$ between the tree unfoldings, the subtree of root $c^{\mathfrak{A}}Rd^{\mathfrak{A}}$, which is redundant if $\mathcal{L}$ has counting, is removed from $\mathscr{T}^{c^{\mathfrak{A}}}_{\mathfrak{A}}$. Formally, for all $R \in \mathsf{rel}_2, A \in \mathsf{rel}_1$,

$$\begin{aligned}
\mathsf{dom}(\mathscr{F}^{\mathcal{D}}_{\mathfrak{A}}) = & \bigcup_{c \in \mathsf{cons}(\mathcal{D})} \mathsf{dom}(\mathscr{T}^{c^{\mathfrak{A}}}_{\mathfrak{A}}) \\
& - \{c^{\mathfrak{A}} S d^{\mathfrak{A}} \cdots \mid S(c,d) \in \mathcal{D}\} \\
& - \{c^{\mathfrak{A}} S^- d^{\mathfrak{A}} \cdots \mid S(d,c) \in \mathcal{D}\} \\
R^{\mathscr{F}^{\mathcal{D}}_{\mathfrak{A}}} = & \Big(\bigcup_{c \in \mathsf{cons}(\mathcal{D})} R^{\mathscr{T}^{c^{\mathfrak{A}}}_{\mathfrak{A}}}\Big)_{|\mathsf{dom}(\mathscr{F}^{\mathcal{D}}_{\mathfrak{A}})} \cup \{(c^{\mathfrak{A}}, d^{\mathfrak{A}}) \mid R(c,d) \in \mathcal{D}\} \\
A^{\mathscr{F}^{\mathcal{D}}_{\mathfrak{A}}} = & \Big(\bigcup_{c \in \mathsf{cons}(\mathcal{D})} A^{\mathscr{T}^{c^{\mathfrak{A}}}_{\mathfrak{A}}}\Big)_{|\mathsf{dom}(\mathscr{F}^{\mathcal{D}}_{\mathfrak{A}})}
\end{aligned}$$

If $\mathcal{L}$ contains nominals, $\mathscr{F}^{\mathcal{D}}_{\mathfrak{A},\mathcal{L}}$ is obtained by identifying in the construction above all paths of the form $p_0 a^{\mathfrak{A}} p_1 a^{\mathfrak{A}} p_2$ with $p_0 a^{\mathfrak{A}} p_2$ for all $a \in \mathsf{cons}(\mathcal{D})$.

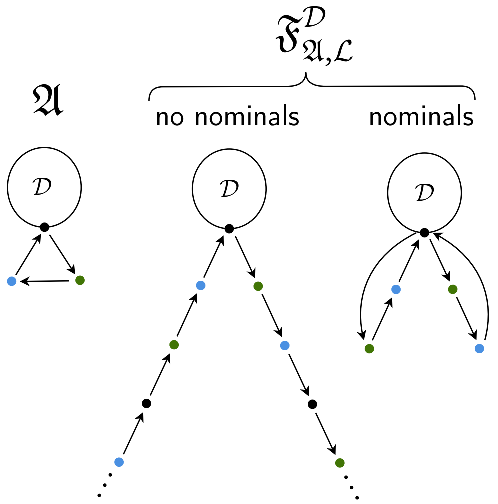
A routine check now gives the following.

1.40. Lemma. *For any $\mathcal{L} \in \mathbf{DL}_{\mathcal{IOQ}}$, model $\mathfrak{A}$ of a database $\mathcal{D}$ and $c \in \mathsf{cons}(\mathcal{D})$, the relation $\{(p, \mathsf{tail}(p)) \mid p \in \mathsf{dom}(\mathscr{F}^{\mathcal{D}}_{\mathfrak{A}})\}$ witnesses $(\mathfrak{A}, c^{\mathfrak{A}}) \sim_{\mathcal{L}} (\mathscr{F}^{\mathcal{D}}_{\mathfrak{A},\mathcal{L}}, c^{\mathfrak{A}})$.*

Such unfoldings give rise to *forest models*.

1.41. *Definition.* If $\mathcal{L} \in \mathbf{DL}_{\mathcal{IQ}}$ and $\mathcal{D}$ is a database, we say $\mathfrak{A}$ is an *$\mathcal{L}$-forest model of $\mathcal{D}$* if $\mathfrak{A}'$ is a disjoint union of $\mathcal{L}$-trees rooted in elements of $\{c^{\mathfrak{A}'} \mid c \in \mathsf{cons}(\mathcal{D})\}$, where $\mathfrak{A}'$ is defined as $\mathfrak{A}$ except for $R^{\mathfrak{A}'} = R^{\mathfrak{A}} \setminus \{(a^{\mathfrak{A}}, b^{\mathfrak{A}}) : R(a,b) \in \mathcal{D}\}$. If $\mathcal{L} = \mathbf{DL}_{\mathcal{IOQ}} \setminus \mathbf{DL}_{\mathcal{IQ}}$ and $\mathcal{D}$ is a database, we say $\mathfrak{A}$ is an *$\mathcal{L}$-forest model of $\mathcal{D}$* if $\mathfrak{A}'$ is a disjoint union of $\mathcal{L}$-trees rooted in elements of $\{c^{\mathfrak{A}'} \mid c \in \mathsf{cons}(\mathcal{D})\}$, where $\mathfrak{A}'$

is defined as $\mathfrak{A}$ except for $R^{\mathfrak{A}'} = R^{\mathfrak{A}} \setminus \{(x, b^{\mathfrak{A}}) : x \in \mathsf{dom}(\mathfrak{A}), b \in \mathsf{cons}(\mathcal{D})\}$ for all $R \in \mathsf{rel}_2$.



$\mathcal{D}_{\mathcal{L}}$-forest unfoldings of an example model $\mathfrak{A}$ of some database $\mathcal{D}$ illustrating the difference between $\mathcal{L}$ having nominals and not. The right unfolding is an $\mathcal{ALCQIO}$-forest model but not an $\mathcal{ALCQI}$ one.

We can then establish some form of "forest model property" for $\mathbf{DL}_{\mathcal{IQ}}$. It can be proved using the same arguments as the tree model property and Lemma 1.40.

1.42. Lemma. *Let $\mathcal{L} \in \mathbf{DL}_{\mathcal{IOQ}} \setminus \{\mathcal{ALCIO}, \mathcal{ALCQIO}\}$, $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ an $\mathcal{L}$-knowledge base, $a \in \mathsf{cons}(\mathcal{D})$ and C an $\mathcal{L}$-concept. If $\mathcal{K} \not\models C(a)$, then $a^{\mathfrak{A}} \notin C^{\mathfrak{A}}$ for some $\mathcal{L}$-forest model of $\mathcal{K}$ of finite outdegree.*

1.43. Example 4.3 witnesses failure of the above Lemma with $\mathcal{L} = \mathcal{ALCIO}$. As D separates, it holds that $\mathcal{K} \not\models D(b)$. By point (2) and $\mathcal{K} \models D(a)$, for any $\mathcal{ALCIO}$-forest model $\mathfrak{A} \models \mathcal{K}$ of finite $\mathcal{ALCIO}$-outdegree, $\mathfrak{A} \models D(b^{\mathfrak{A}})$.

§ 1.5. GUARDED FRAGMENT

The Guarded Fragment (**GF**) of first-order logic, introduced in [ABN98], subsumes $\mathcal{ALCI}$ while enjoying desirable model-theoretic and decidability properties that **FO** does not. It is then also considered as an ontology language, in particular in the context of ontology-mediated querying [BGO14]. We include it as an ontology language and a separation language. The following section consists of basic definitions and properties.

For any tuple $\mathbf{x} = (x_1, \ldots, x_n)$, let $[\mathbf{x}] = \{x_1, \ldots, x_n\}$.

1.44. *Definition.*

1. $(x = y)$ is a **GF**-formula for any $x, y \in \mathsf{var}$.
2. $R(\mathbf{x})$ is a **GF**-formula for any $n \geqslant 1$, $R \in \mathsf{rel}_n$ and $\mathbf{x} \in \mathsf{var}^n$.
3. If $\varphi(\mathbf{x}, \mathbf{y}), \psi$ are **GF**-formulas, then
   - $(\varphi \wedge \psi)$ is a **GF**-formula,
   - $\neg\varphi$ is a **GF**-formula,
   - $\exists \mathbf{y}(\alpha(\mathbf{x}, \mathbf{y}) \wedge \varphi)$ is a **GF**-formula, where $\alpha(\mathbf{x}, \mathbf{y})$ (called the *guard*) is any atomic **FO**-formula such that every variable in $[\mathbf{x}] \cup [\mathbf{y}]$ occurs in it.

1.45. The same standard translation function that shows $\mathcal{ALCI}$ can be seen as a fragment of **FO** also shows that $\mathcal{ALCI}$ can be seen as a fragment of **GF**.

The notion of bisimulation from $\mathcal{ALCI}$ can be extended to **GF** and, there, also characterises logical equivalence. We introduce it for **GF** as well as for its fragment **oGF** (defined next) that is involved in the study of separability in **GF**.

1.46. *Definition.* Let **oGF** be the fragment of **GF** that consists of all open (*i.e.* not quantifier-closed) formulas in **GF** whose subformulas are all open.

1.47. **oGF** was first considered in [HLPW20] where it is also observed that a **GF** formula is equivalent to an **oGF** formula iff it is invariant under disjoint unions. It is more natural for separability, as it only speaks locally about the neighbourhood of tuples and not disconnected parts: when it comes to separating positive from negative examples, properties of points "unrelated" to either are not relevant.

1.48. *Definition.* Let $\mathfrak{A}$ be a model and Σ a signature. For any tuple $\mathbf{a} = (a_1, \dots, a_n)$ we write $[\mathbf{a}]$ for $\{a_1, \dots, a_n\}$. A set $G \subseteq \mathsf{dom}(\mathfrak{A})$ is *guarded* in $\mathfrak{A}$ if G is a singleton or there exists R with $\mathfrak{A} \models R(\mathbf{a})$ such that $G = [\mathbf{a}]$. By $S(\mathfrak{A})$, we denote the set of all guarded sets in $\mathfrak{A}$. A tuple $\mathbf{a} \in \mathsf{dom}(\mathfrak{A})^n$ is *guarded* in $\mathfrak{A}$ if $[\mathbf{a}]$ is a subset of some guarded set in $\mathfrak{A}$.

For tuples $\mathbf{a} = (a_1, \dots, a_n)$ in $\mathfrak{A}$ and $\mathbf{b} = (b_1, \dots, b_n)$ in $\mathfrak{B}$ we call a mapping p from $[\mathbf{a}]$ to $[\mathbf{b}]$ with $p(a_i) = b_i$ for $1 \leqslant i \leqslant n$ (written $p : \mathbf{a} \mapsto \mathbf{b}$) a *partial Σ-homomorphism* if p is a homomorphism from the Σ-reduct of $\mathfrak{A}_{|[\mathbf{a}]}$ to $\mathfrak{B}_{|[\mathbf{b}]}$. We call p a *partial Σ-isomorphism* if, in addition, the inverse of p is a partial Σ-homomorphism with domain $\mathfrak{B}_{|[\mathbf{b}]}$.

A set I of partial Σ-isomorphisms $p : \mathbf{a} \mapsto \mathbf{b}$ from guarded tuples $\mathbf{a}$ in $\mathfrak{A}$ to guarded tuples $\mathbf{b}$ in $\mathfrak{B}$ is called a

- *guarded Σ-bisimulation* if the following hold for all $p : \mathbf{a} \mapsto \mathbf{b} \in I$:
  1. for every guarded tuple $\mathbf{a}'$ in $\mathfrak{A}$ there exists a guarded tuple $\mathbf{b}'$ in $\mathfrak{B}$ and $p' : \mathbf{a}' \mapsto \mathbf{b}' \in I$ such that p' and p coincide on $[\mathbf{a}] \cap [\mathbf{a}']$;

2. for every guarded tuple $\mathbf{b}'$ in $\mathfrak{B}$ with there exists a guarded tuple $\mathbf{a}'$ in $\mathfrak{A}$ and $p' : \mathbf{a}' \mapsto \mathbf{b}' \in I$ such that p'^{-1} and p^{-1} coincide on $[\mathbf{b}] \cap [\mathbf{b}']$.

- *connected guarded Σ-bisimulation* if the following hold for all $p : \mathbf{a} \mapsto \mathbf{b} \in I$:
  1. for every guarded tuple $\mathbf{a}'$ in $\mathfrak{A}$ with $[\mathbf{a}] \cap [\mathbf{a}'] \neq \emptyset$ there exists a guarded tuple $\mathbf{b}'$ in $\mathfrak{B}$ and $p' : \mathbf{a}' \mapsto \mathbf{b}' \in I$ such that p' and p coincide on $[\mathbf{a}] \cap [\mathbf{a}']$;
  2. for every guarded tuple $\mathbf{b}'$ in $\mathfrak{B}$ with $[\mathbf{b}] \cap [\mathbf{b}'] \neq \emptyset$ there exists a guarded tuple $\mathbf{a}'$ in $\mathfrak{A}$ and $p' : \mathbf{a}' \mapsto \mathbf{b}' \in I$ such that p'^{-1} and p^{-1} coincide on $[\mathbf{b}] \cap [\mathbf{b}']$.

Assume that $\mathbf{a}$ and $\mathbf{b}$ are (possibly not guarded) tuples in $\mathfrak{A}$ and $\mathfrak{B}$. Then we say that $(\mathfrak{A}, \mathbf{a})$ and $(\mathfrak{B}, \mathbf{b})$ are *(resp. connected) guarded Σ-bisimilar*, in symbols $(\mathfrak{A}, \mathbf{a}) \sim_{\mathbf{GF},\Sigma} (\mathfrak{B}, \mathbf{b})$ (resp. **oGF**), if there exists a partial Σ-isomorphism $p : \mathbf{a} \mapsto \mathbf{b}$ and a (resp. connected) guarded Σ-bisimulation I such that conditions (i) and (ii) hold for p [HLPW20].

If there exist sets $I_\ell, \dots, I_0$ of partial Σ-isomorphisms such that I_ℓ contains the partial Σ-isomorphism $p : \mathbf{a} \mapsto \mathbf{b}$ and for any $1 \leqslant i \leqslant \ell$ and $p \in I_i$, conditions (1) and (2) above are each witnessed by some $p' \in I_{i-1}$ then we say that $(\mathfrak{A}, \mathbf{a})$ and $(\mathfrak{B}, \mathbf{b})$ are *(connected) guarded Σ ℓ-bisimilar* and write $(\mathfrak{A}, \mathbf{a}) \sim^{\ell}_{\mathbf{oGF},\Sigma} (\mathfrak{B}, \mathbf{b})$ and $(\mathfrak{A}, \mathbf{a}) \sim^{\ell}_{\mathbf{GF},\Sigma} (\mathfrak{B}, \mathbf{b})$, respectively.

The *guarded quantifier rank* $\mathrm{gr}(\varphi)$ of a formula φ in **GF** is the number of nestings of guarded quantifiers in it. We say that $(\mathfrak{A}, \mathbf{a})$ and $(\mathfrak{B}, \mathbf{b})$ are $\mathbf{GF}^{\ell}(\Sigma)$*-equivalent* (resp. **oGF**), in symbols $(\mathfrak{A}, \mathbf{a}) \equiv^{\ell}_{\mathbf{GF},\Sigma} (\mathfrak{B}, \mathbf{b})$ (resp. **oGF**), if $\mathfrak{A} \models \varphi(\mathbf{a}) \Leftrightarrow \mathfrak{B} \models \varphi(\mathbf{b})$ for all formulas $\varphi(\mathbf{x})$ in $\mathbf{GF}(\Sigma)$ (resp. **oGF**) of guarded quantifier rank at most ℓ.

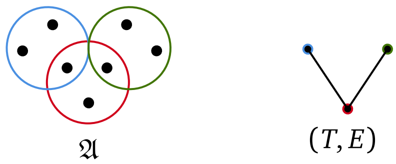
1.49. Lemma ([GO14, HLPW20]). *For any $\mathcal{L} \in \{\mathbf{GF}, \mathbf{oGF}\}$, pointed models $(\mathfrak{A}, \mathbf{a}), (\mathfrak{B}, \mathbf{b})$, signature Σ and $\ell \geqslant 0$,*

1. $(\mathfrak{A}, \mathbf{a}) \sim^{\ell}_{\mathcal{L},\Sigma} (\mathfrak{B}, \mathbf{b}) \Leftrightarrow (\mathfrak{A}, \mathbf{a}) \equiv^{\ell}_{\mathcal{L},\Sigma} (\mathfrak{B}, \mathbf{b})$*,*
2. $(\mathfrak{A}, \mathbf{a}) \sim_{\mathcal{L},\Sigma} (\mathfrak{B}, \mathbf{b}) \Rightarrow (\mathfrak{A}, \mathbf{a}) \equiv_{\mathcal{L},\Sigma} (\mathfrak{B}, \mathbf{b})$,
3. $(\mathfrak{A}, \mathbf{a}) \sim_{\mathcal{L},\Sigma} (\mathfrak{B}, \mathbf{b}) \Leftarrow (\mathfrak{A}, \mathbf{a}) \equiv_{\mathcal{L},\Sigma} (\mathfrak{B}, \mathbf{b})$ *if $\mathfrak{A}$ and $\mathfrak{B}$ are ω-saturated.*

We introduce guarded tree decompositions. They already appear *e.g.* in [G99]. A model that admits a guarded tree decomposition is a model that 'can be seen as a tree' in the **GF** context where, in particular, relations are not necessary binary.

1.50. *Definition.* A *guarded tree decomposition* of a model $\mathfrak{A}$ is a triple (T, E, bag) with (T, E) an undirected tree and bag a function that assigns to every $t \in T$ a guarded set $\mathrm{bag}(t)$ in $\mathfrak{A}$ such that $\mathfrak{A} = \bigcup_{t \in T} \mathfrak{A}_{|\mathrm{bag}(t)}$ and $\{t \in T \mid a \in \mathrm{bag}(t)\}$

is connected in (T, E), for every $a \in \mathsf{dom}(\mathfrak{A})$. We say that $\mathfrak{A}$ is *guarded tree decomposable* if there exists a guarded tree decomposition of $\mathfrak{A}$.



Guarded tree decomposition of some example model $\mathfrak{A}$*, with colors representing the* bag *mapping.*

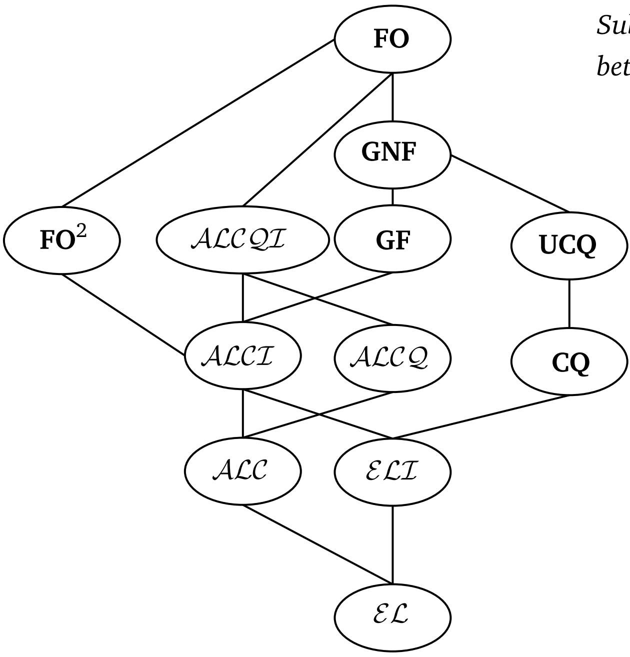
Using guarded tree decompositions one can formulate a variation of the tree model property for the Guarded Fragment.

1.51. Proposition (Tree model property, [G99]). *For every* **GF***-ontology* $\mathcal{O}$ *and* **GF***-formula* $\varphi(\mathbf{x})$ *such that* $\mathcal{O} \not\models \varphi$ *there exists a guarded tree-decomposable model* $\mathfrak{A}$ *of* $\mathcal{O}$ *such that* $\mathfrak{A} \models \neg\varphi(\mathbf{a})$ *for a tuple* **a** *with* $[\mathbf{a}] \subseteq \mathsf{bag}(r)$.

The extension **GNF** (for *Guarded Negation Fragment*) of **GF** makes minor apparitions throughout the thesis. Unlike **GF**, it subsumes **UCQ**.

1.52. *Definition.*

1. $(x = y)$ is a **GNF**-formula for any $x, y \in \mathsf{var}$.
2. $R(\mathbf{x})$ is a **GNF**-formula for any $n \geqslant 1$, $R \in \mathsf{rel}_n$ and $\mathbf{x} \in \mathsf{var}^n$.
3. If $\varphi(\mathbf{x}, \mathbf{y})$ are **GNF**-formulas, then
   - $(\varphi \wedge \psi)$ is a **GNF**-formula,
   - $(\varphi \vee \psi)$ is a **GNF**-formula,
   - $\exists \mathbf{y} \varphi$ is a **GNF**-formula,
   - $\alpha(\mathbf{x}, \mathbf{y}) \wedge \neg\varphi$ is a **GNF**-formula, where $\alpha(\mathbf{x}, \mathbf{y})$ is any atomic **FO**-formula such that every variable in $[\mathbf{x}] \cup [\mathbf{y}]$ occurs in it.



Subsumption relations between languages.

<table>
<tr><th></th><th>ARBITRARY</th><th>ROOTED</th></tr>
<tr><td>FO</td><td colspan="2">Undecidable</td></tr>
<tr><td>FO^2</td><td colspan="2">Undecidable [Ro07]</td></tr>
<tr><td>GNF</td><td colspan="2">2EXP [BCS15]</td></tr>
<tr><td>GF</td><td colspan="2">2EXP [BGO14]</td></tr>
<tr><td>$\mathcal{ALCQI}$</td><td rowspan="2">2EXP [Lu07]</td><td rowspan="2">CONEXP [Lu08]</td></tr>
<tr><td>$\mathcal{ALCI}$</td></tr>
<tr><td>$\mathcal{ALCQ}$</td><td colspan="2" rowspan="2">EXP [Lu08]</td></tr>
<tr><td>$\mathcal{ALC}$</td></tr>
<tr><td>$\emptyset$</td><td colspan="2">NP</td></tr>
</table>

Combined complexity of (**U**)**CQ**-*evaluation on* $\mathcal{L}$-*knowledge bases.*

<table>
<tr><th></th><th>COMBINED</th><th>DATA</th></tr>
<tr><td>$\mathcal{ALC}$</td><td>EXP [Sc91]</td><td>NP</td></tr>
<tr><td>$\mathcal{ALCIO}$</td><td>EXP [ABM99]</td><td>NP</td></tr>
<tr><td>$\mathcal{ALCQI}$</td><td>EXP</td><td>NP</td></tr>
<tr><td>GF</td><td>2EXP [G99]</td><td>NP</td></tr>
<tr><td>GNF</td><td>2EXP [BCS15]</td><td>NP</td></tr>
<tr><td>FO^2</td><td>NEXP [GKV97]</td><td>NP</td></tr>
<tr><td>FO</td><td colspan="2">Undecidable</td></tr>
</table>

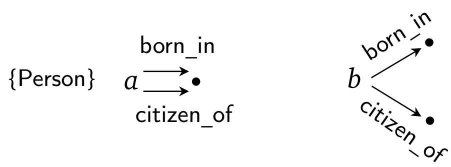
Combined and data complexity of satisfiability.

§ 1.6. The separability problem

We define the separability problem in its different dimensions (projective, weak, strong, restricted, full) and enunciate some of its most fundamental properties.

1.53. *Definition.* Let $\mathcal{L}_O$ be a fragment of **FO**. We call *labeled $\mathcal{L}_O$-knowledge base* any triple $(\mathcal{K}, E^+, E^-)$ with $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ an $\mathcal{L}_O$-knowledge base such that $\mathcal{O}$ is a constantless $\mathcal{L}_O$-ontology and $E^+, E^- \subseteq \mathsf{cons}(\mathcal{D})^n$ (for some $n \geqslant 1$) non-empty sets of *positive* and *negative* examples. An **FO**-formula $\varphi(\mathbf{x})$ with n free variables *weakly separates* $(\mathcal{K}, E^+, E^-)$ if $\mathcal{K} \models \varphi(\mathbf{a})$ and $\mathcal{K} \not\models \varphi(\mathbf{b})$ for all $\mathbf{a} \in E^+, \mathbf{b} \in E^-$, and *strongly separates* $(\mathcal{K}, E^+, E^-)$ if $\mathcal{K} \models \varphi(\mathbf{a})$ and $\mathcal{K} \models \neg\varphi(\mathbf{b})$ for all $\mathbf{a} \in E^+, \mathbf{b} \in E^-$. Let $\mathcal{L}_S$ be a fragment of **FO** and $\Sigma \subseteq \mathsf{rel}$. A labeled **FO**-knowledge base is *projectively $\mathcal{L}_S(\Sigma)$-separated* if it is separated by an $\mathcal{L}_S$-formula φ such that $\mathsf{sig}(\varphi) \cap \mathsf{sig}(\mathcal{K}) \subseteq \Sigma$. It is *non-projectively $\mathcal{L}_S(\Sigma)$-separated* if it is separated by an $\mathcal{L}_S$-formula φ such that $\mathsf{sig}(\varphi) \subseteq \Sigma$. As convention, unless specified otherwise, separability will be assumed to be weak, non-projective and *full* (*i.e.* such that $\Sigma = \mathsf{sig}(\mathcal{K})$). We may then write "$\mathcal{L}$-separable" in place of "weakly non-projectively $\mathcal{L}(\mathsf{sig}(\mathcal{K}))$-separable".

1.54. *Example.* Let $\mathcal{K}_1 = (\emptyset, \mathcal{D})$ where $\mathcal{D} = \big\{\mathsf{born\_in}(a,c), \mathsf{citizen\_of}(a,c), \mathsf{born\_in}(b,c_1), \mathsf{citizen\_of}(b,c_2), \mathsf{Person}(a)\big\}$. Then $(\mathcal{K}_1, \{a\}, \{b\})$ is weakly separated by $\mathsf{Person}(x)$. Now let $\mathcal{O} = \big\{\forall x(\exists y(\mathsf{citizen\_of}(x,y)) \to \mathsf{Person}(x))\big\}$ and $\mathcal{K}_2 = (\mathcal{O}, \mathcal{D})$. Then $\mathcal{K}_2 \models \mathsf{Person}(b)$, so $\mathsf{Person}(x)$ no longer separates. However, $(\mathcal{K}_2, \{a\}, \{b\})$ is separated by the formula $\exists y(\mathsf{born\_in}(x,y) \wedge \mathsf{citizen\_of}(x,y))$. Thus $(\mathcal{K}_2, \{a\}, \{b\})$ is weakly non-projectively $\mathcal{L}$-separable for $\mathcal{L} = \mathbf{CQ}$ and $\mathcal{L} = \mathbf{GF}$ and $\mathcal{L} = \mathbf{FO}^2$. It is also weakly non-projectively $\mathcal{ALCI}$-separated, although unnaturally, by $\forall\mathsf{born\_in}.\mathsf{Person} \to \exists\mathsf{citizen\_of}.\mathsf{Person}$.



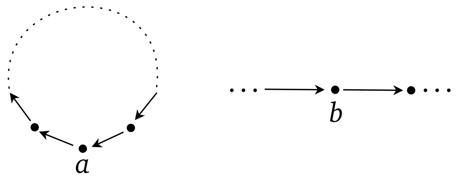
A typical witness for the distinction between projective and non-projective separability is given by the following example, in the context of $\mathcal{ALCI}$ and weak separation.

1.55. *Example.* Let $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ with

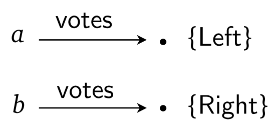
$$\mathcal{D} = \{R(a,a_1), R(a_1,a_2), \ldots, R(a_{n-1},a), R(b,b_1)\}$$
$$\mathcal{O} = \{\top \sqsubseteq \exists R.\top \sqcap \exists R^-.\top\},$$

Then $(\mathcal{K}, \{a\}, \{b\})$ is

1. non-projectively **CQ**($\{R\}$)-separated by $\exists y_1 \ldots \exists y_n R(x, y_1) \wedge \cdots \wedge R(y_{n-1}, x)$,
2. projectively $\mathcal{ALCI}(\{R\})$-separated by $A \to \exists R^n.A$ for any $A \in \mathsf{rel}_1 \setminus \mathsf{sig}(\mathcal{K})$,
3. not non-projectively $\mathcal{ALCI}(\{R\})$-separable,
4. not non-projectively **GF**($\{R\}$)-separable,
5. not non-projectively $\mathbf{FO}^2(\{R\})$-separable.



1.56. *Example.* Let $\mathcal{D} = \{\mathsf{votes}(a, c_1), \mathsf{votes}(b, c_2), \mathsf{Left}(c_1), \mathsf{Right}(c_2)\}$ and $\mathcal{K}_1 = (\emptyset, \mathcal{D})$. Then $(\mathcal{K}_1, \{a\}, \{b\})$ is weakly separated by the $\mathcal{EL}$-concept $\exists\mathsf{votes}.\mathsf{Left}$, but it is not strongly **FO**-separable. Now let $\mathcal{K}_2 = (\mathcal{O}, \mathcal{D})$ with $\mathcal{O} = \{\exists\mathsf{votes}.\mathsf{Left} \sqsubseteq \neg\exists\mathsf{votes}.\mathsf{Right}\}$. Then $\exists\mathsf{votes}.\mathsf{Left}$ strongly separates $(\mathcal{K}_2, \{a\}, \{b\})$.



1.57. As illustrated by Example 1.56, 'negative information' introduced by the ontology is crucial for strong separability because of the open-world semantics and since the database cannot contain negative information. In fact, *labeled KBs with an empty ontology are never strongly separable*. In a sense, weak separability tends to be too credulous if the data is incomplete regarding positive information, see Example 1.54, while strong separability tends to be too skeptical if the data is incomplete regarding negative information, as shown by Example 1.56.

1.58. Strong separability is always trivially monotone in the ontology (and also the database) in the sense that for any $\mathcal{L}_S$ and all labeled KBs $(\mathcal{K}_i, E^+, E^-)$ with $\mathcal{K}_i = (\mathcal{O}_i, \mathcal{D}_i)$ for $i = 1, 2$, if $\mathcal{O}_1 \subseteq \mathcal{O}_2$, $\mathcal{D}_1 \subseteq \mathcal{D}_2$, and $(\mathcal{K}_1, E^+, E^-)$ is $\mathcal{L}_S$-separable, then $(\mathcal{K}_2, E^+, E^-)$ is $\mathcal{L}_S$-separable.

1.59. No labeled $\mathcal{ELI}$-knowledge base is (projectively) strongly **FO**-separable. Intuitively, $\mathcal{ELI}$-ontologies only contain "positive" information and can thus not entail anything "negative". Rigorously, it immediately follows (even under UNA) *e.g.* from the existence of the model $\mathfrak{A}$ defined by $\mathsf{dom}(\mathfrak{A}) = \{x_c\}_{c \in \mathsf{cons}(\mathcal{D})}$, $c^{\mathfrak{A}} = \{x_c\}$ for all $c \in \mathsf{cons}(\mathcal{D})$ and $R^{\mathfrak{A}} = \mathsf{dom}(\mathfrak{A})^n$ for all $R \in \mathsf{rel}_n$, $n \geqslant 1$. Then, $\mathfrak{A} \models (\mathcal{O}, \mathcal{D})$ for any $\mathcal{ELI}$-ontology $\mathcal{O}$, and $\mathfrak{A} \models \varphi(\mathbf{a}^{\mathfrak{A}}) \iff \mathfrak{A} \models \varphi(\mathbf{b}^{\mathfrak{A}})$ for any **FO**-formula φ and $\mathbf{a}, \mathbf{b} \subseteq \mathsf{cons}(\mathcal{D})$ of same arity.

Unlike weak separability, strong separability is not influenced by symbols that do not occur in the knowledge base. Intuitively, given a knowledge base $\mathcal{K}$, any 'positive occurrence' of an external symbol that is satisfied by a model of $\mathcal{K}$ is also negated by another model of $\mathcal{K}$. Then, for any separating formula containing such a symbol, the same formula 'without it' is still separating. The following applies to any language $\mathcal{L}$ that occurs in this thesis. From now on we omit the mention of (non-)projectivity and simply speak of strong separability.

1.60. Proposition. *Any labeled* **FO**-*knowledge base* $(\mathcal{K}, E^+, E^-)$ *is strongly projectively* $\mathcal{L}(\Sigma)$*-separable iff it is strongly non-projectively* $\mathcal{L}(\Sigma)$*-separable, for* $\Sigma \subseteq \mathsf{sig}(\mathcal{K})$.

Proof. Assume $\varphi(\mathbf{x})$ strongly separates $(\mathcal{K}, E^+, E^-)$ and $R \in \mathsf{sig}(\varphi) \setminus \mathsf{sig}(\mathcal{K})$. Replace every occurrence of any formula of the form $R(\mathbf{y})$ in $\varphi(\mathbf{x})$ by $\bigwedge_{y \in [\mathbf{y}]}(y = y)$. For **CQ**, **UCQ**, $\mathbf{FO}^2$, and **FO**, we show the resulting formula φ' is as required and strongly separates $(\mathcal{K}, E^+, E^-)$. For any model $\mathfrak{A}$, let $\mathfrak{A}'$ denote the model obtained from $\mathfrak{A}$ by replacing $R^{\mathfrak{A}}$ with $\mathsf{dom}(\mathfrak{A})$, for any $R \in \mathsf{rel}_n$ and $n \geqslant 1$. Then, for any tuple $\mathbf{x}$ in $\mathsf{dom}(\mathfrak{A})$, $(\mathfrak{A}', \mathbf{x}) \simeq_{\mathsf{rel} \setminus \{R\}} (\mathfrak{A}, \mathbf{x})$ (in particular $\mathfrak{A}' \models \mathcal{K}$) and $\mathfrak{A}' \models \varphi'(\mathbf{x})$ iff $\mathfrak{A}' \models \varphi(\mathbf{x})$. It is then immediate that φ' separates. Let $\mathbf{a} \in E^+$ and $\mathbf{b} \in E^-$. Let $\mathfrak{A} \models \mathcal{K}$. Then $\mathfrak{A} \models \varphi(\mathbf{a}^{\mathfrak{A}}) \Rightarrow \mathfrak{A}' \models \varphi(\mathbf{a}^{\mathfrak{A}'}) \Rightarrow \mathfrak{A}' \models \varphi'(\mathbf{a}^{\mathfrak{A}'}) \Rightarrow \mathfrak{A} \models \varphi'(\mathbf{a}^{\mathfrak{A}})$. On the other hand, let $\mathfrak{A} \models \mathcal{K}$ such that $\mathfrak{A} \not\models \varphi(\mathbf{b}^{\mathfrak{A}})$. Then $\mathfrak{A} \not\models \varphi(\mathbf{b}^{\mathfrak{A}}) \Rightarrow \mathfrak{A}' \not\models \varphi(\mathbf{b}^{\mathfrak{A}'}) \Rightarrow \mathfrak{A}' \not\models \varphi'(\mathbf{b}^{\mathfrak{A}'})$. In the guarded fragments, if R occurs as a guard in φ, then the resulting formula might not be guarded. In this case we replace every subformula of the form $\exists \mathbf{y}(R(\mathbf{x}, \mathbf{y}) \wedge \psi)$ in $\varphi(\mathbf{x})$ by the conjunction of all $\neg(x = x)$ with x in $\mathbf{x}$ and every occurrence of $R(\mathbf{y})$ in $\varphi(\mathbf{x})$ in a non-guard position by the conjunction of $\neg(y = y)$ with y in $\mathbf{y}$. Finally, in $\mathbf{DL}_{\mathcal{IQ}}$, assume that the concept C strongly separates $(\mathcal{K}, E^+, E^-)$ and $X \in \mathsf{sig}(C) \setminus \mathsf{sig}(\mathcal{K})$. If X is a concept name, then replace every occurrence of X in C by $\bot$ and if X is a role name, then replace every subconcept of the form $\geqslant nX.D$ or $\geqslant nX^-.D$ in C by $\bot$. The resulting concept strongly separates. A similar proof as above can be given, where the witness model $\mathfrak{A}'$ is instead defined with $R^{\mathfrak{A}'} = \emptyset$. ⊣

1.61. *Definition.* Let **FO**-fragments $\mathcal{L}_0, \mathcal{L}_S$ be given.

1. Let $\mathsf{sep}_w(\mathcal{L}_O, \mathcal{L}_S)$ (resp. $\mathsf{sep}^p_w(\mathcal{L}_O, \mathcal{L}_S)$) denote the set of all weakly non-projectively (resp. projectively) $\mathcal{L}_S$-separable labeled $\mathcal{L}$-knowledge bases. We call (resp. *projective*) *full weak* $(\mathcal{L}_0, \mathcal{L}_S)$-separability the problem to decide $\mathsf{sep}_w(\mathcal{L}_O, \mathcal{L}_S)$ (resp. $\mathsf{sep}^p_w(\mathcal{L}_O, \mathcal{L}_S)$).
2. Let $\mathsf{sep}_{sig,w}(\mathcal{L}_O, \mathcal{L}_S)$ (resp. $\mathsf{sep}^p_{sig,w}(\mathcal{L}_O, \mathcal{L}_S)$) denote the set of all $(\mathcal{K}, E^+, E^-, \Sigma)$ such that $(\mathcal{K}, E^+, E^-)$ is a weakly non-projectively (resp. projectively) $\mathcal{L}_S(\Sigma)$-separable $\mathcal{L}_O$-knowledge base. We call (resp. *projective*) *restricted weak* $(\mathcal{L}_0, \mathcal{L}_S)$-separability the associated decision problem.

3. Define their strong counterparts by replacing w with s and removing p.

1.62. *Reduction to the single example case.* Weak separability is polynomial-time Turing reducible to its subcase where E^- is a singleton. If for every $\mathbf{b} \in E^-$ there is a formula $\varphi_{\mathbf{b}}$ separating $(\mathcal{K}, E^+, \{\mathbf{b}\})$, then $\bigwedge_{\mathbf{b} \in E^-} \varphi_{\mathbf{b}}$ separates $(\mathcal{K}, E^+, E^-)$. On the other hand, if φ separates $(\mathcal{K}, E^+, E^-)$ then φ obviously separates $(\mathcal{K}, E^+, \{\mathbf{b}\})$ for every $\mathbf{b} \in E^-$. The same argument works between strong separability and its subcase where both E^+ and E^- are singletons. Throughout the thesis we most frequently assume E^- (resp. E^+ and E^-) is a singleton. Most of the complexity bounds we obtain are proved for that specific case. For each of them it is easy to argue that they hold in the general case too, using the reduction above.

1.63. *The satisfiability lower bound.* For any **FO**-fragments $\mathcal{L}_O, \mathcal{L}_S$, weak (projective) and strong $(\mathcal{L}_O, \mathcal{L}_S)$-separability are computationally at least as hard as satisfiability of $\mathcal{L}$-knowledge bases. Let $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ be an $\mathcal{L}$-knowledge base. Suppose wlog $A, B \in \mathsf{rel}_1$ and $a, b \in \mathsf{cons}$ do not occur in $\mathcal{K}$. Then the following equivalences hold.

$$
\begin{aligned}
& \mathcal{K} \text{ is satisfiable by a model of cardinality} \geqslant 2. \\
\Leftrightarrow\ & \big((\mathcal{O}, \mathcal{D} \cup \{A(a), B(b)\}), \{a\}, \{b\}\big) \text{ is weakly } \mathcal{L}_S\text{-separable (by } A(x)). \\
\Leftrightarrow\ & \big((\mathcal{O}, \mathcal{D} \cup \{A(a), A(b)\}), \{a\}, \{b\}\big) \text{ is not strongly } \mathcal{L}_S\text{-separable.}
\end{aligned}
$$

As immediate consequences, if the ontology language has an undecidable satisfiability problem (*e.g.* **FO**), then any kind of separability problem is undecidable; if the ontology language has an untractable satisfiability problem (*e.g.* any language other than $\mathcal{EL}$ among the ones we consider), then any kind of separability problem is untractable.

Chapter **2**

Full weak separability

We now study, for several pairs of languages $(\mathcal{L}_O, \mathcal{L}_S)$, the problem of weakly separating labeled $\mathcal{L}_O$-knowledge bases $(\mathcal{K}, E^+, E^-)$ with $\mathcal{L}_S(\Sigma)$-formulas (or, in the projective case, $\mathcal{L}_S(\Sigma \cup \Sigma'))$ for some set Σ' of fresh relation symbols) where Σ is the full signature of $\mathcal{K}$. We mostly focus on the case where $\mathcal{L}_O = \mathcal{L}_S$, hence the denomination of each section simply by the ontology language.

The reader is referred to Section 1.6 for definitions of (full) (weak) $(\mathcal{L}_O, \mathcal{L}_S)$-separability, projectivity, etc.

This chapter is comprised of 6 sections.

(2.1) On $(\mathbf{FO}, \mathbf{FO})$-separability, where, by model-theoretically characterising the problem, we show that it is unaffected by projectivity, that **UCQ** can separate as well as **FO**, and that weakly separating with **FO**-formulas is, as a decision problem, linked to query evaluation.

(2.2) On $(\mathbf{FO}^2, \mathbf{FO}^2)$-separability, where, without model-theoretically characterising the problem, we show that projectivity does not bring enough separating power to $\mathbf{FO}^2$ to match **FO** and that the problem is undecidable with both as separation languages.

(2.3) On projective $(\mathcal{L}, \mathcal{L})$-separability for $\mathcal{L} \in \mathbf{DL}_{\mathcal{IQ}}$, where we find a uniform model-theoretic characterisation of separability over those languages and show that, as with **FO**, the decision problem is connected to query evaluation, which gives us tight complexity bounds for each language.

(2.4) On non-projective $(\mathcal{ALCI}, \mathcal{ALCI})$-separability, where, by model-theoretically characterising the problem, we show that, even though it does not coincide with its projective counterpart, it has the same complexity (NEXP-complete).

(2.5) On $(\mathbf{GF}, \mathbf{GF})$-separability, where we notice that **GF** behaves similarly to $\mathcal{ALCI}$, which implies that, both in the projective and non-projective case, we can establish characterisations and a link with query evaluation. Complexity bounds ensue.

(2.6) On $(\mathcal{ALC}, \mathcal{EL}(\mathcal{I}))$-separability, where we prove its undecidability via reduction from a query entailment problem.

Below is a summary of this chapter's complexity results for full signature weak $(\mathcal{L}, \mathcal{L})$-separability, where $\mathcal{L}$ is displayed in the left column. Separability is complete for the complexity classes contained in each cell. A grey cell means the problem is still open, although we get some upper bounds for free, as the data complexity of any problem is trivially bounded upwards by its combined complexity. With $\mathcal{ALCQ}$ and $\mathcal{ALCQI}$, our results only hold under the Unique Name Assumption (UNA) [BHLS17], *i.e.* the assumption that for all $c_1, c_2 \in \mathsf{cons}$ distinct and any model $\mathfrak{A}$, $c_1^{\mathfrak{A}} \neq c_2^{\mathfrak{A}}$.

	Combined		Data	
	Proj	Non-proj	Proj	Non-proj
FO	U			
FO^2	U	U	U	U
GNF	2Exp		$\leqslant$ 2Exp	
GF	2Exp	2Exp	$\leqslant$ 2Exp	$\leqslant$ 2Exp
$\mathcal{ALC}$	NExp	?	PSpace	?
$\mathcal{ALCI}$	NExp	NExp	NExp	NExp
$\mathcal{ALCQ}$ (UNA)	NExp	?	$\leqslant$ NExp	?
$\mathcal{ALCQI}$ (UNA)	Exp	?	$\leqslant$ Exp	?

§ 2.1. FO-ONTOLOGIES

We study full signature weak (**FO**, **FO**)-separability. With **FO**-ontologies, the complexity side of things is already clear: as stated in Remark 1.63, undecidability of **FO** implies that (**FO**, $\mathcal{L}_S$)-separability is undecidable for any language $\mathcal{L}_S$, both in the projective and non-projective case. By model-theoretically characterising separability (Thm. 2.1), we find that for all **FO**-ontologies, **UCQ** is already as powerful as **FO**. More precisely, we can determine an exact UCQ that separates the examples whenever they are **FO**-separable. That UCQ does not use any symbol outside of the knowledge base, which implies that projectivity makes no difference for separation with **FO**-formulas (Cor. 2.2). The existence of such a UCQ also induces a mutual polynomial reduction between the decision problems of separability and query evaluation (Cor. 2.4). That mutual reduction provides us with some complexity bounds 'for free' (Cor. 2.6).

We now introduce the central characterisation theorem for **FO**. Recall that we consider separability of labeled knowledge bases with only one negative example, without loss of generality (§1.6). Recall that we do not consider constants in **FO**-formulas and do not require homomorphisms between models to preserve constants (see Definition 1.6). Recall that for any pointed database $(\mathcal{D}, \mathbf{a})$ we write $\mathcal{D}_{\mathbf{a}}$ for the "connected component of $\mathbf{a}$" in $\mathcal{D}$ and $q(\mathcal{D}, \mathbf{a})$ for the query induced by $(\mathcal{D}, \mathbf{a})$. Precise definitions are given in Remarks 1.13 and 1.10.

2.1. Theorem. *Let $(\mathcal{K}, E^+, \{\mathbf{b}\})$ be a labeled* **FO***-knowledge base, where $\mathcal{K} = (\mathcal{O}, \mathcal{D})$. The following conditions are equivalent.*

1. *$(\mathcal{K}, E^+, \{\mathbf{b}\})$ is weakly projectively* **FO***-separable.*
2. *$(\mathcal{D}_{\mathbf{a}}, \mathbf{a}) \not\to (\mathfrak{A}, \mathbf{b})$ for some model $\mathfrak{A}$ of $\mathcal{K}$ and all $\mathbf{a} \in E^+$.*
3. *$\mathcal{K} \not\models \bigvee_{\mathbf{a} \in E^+} q(\mathcal{D}_{\mathbf{a}}, \mathbf{a})(\mathbf{b})$.*
4. *the UCQ $\bigvee_{\mathbf{a} \in E^+} q(\mathcal{D}_{\mathbf{a}}, \mathbf{a})$ weakly separates $(\mathcal{K}, E^+, \{\mathbf{b}\})$.*
5. *$(\mathcal{K}, E^+, \{\mathbf{b}\})$ is weakly non-projectively* **UCQ***-separable.*

Proof.
(2) $\Rightarrow$ (3) is straightforward by 1.15. (3) $\Rightarrow$ (4) follows from the fact that $\mathcal{D} \models q(\mathcal{D}_{\mathbf{a}}, \mathbf{a})(\mathbf{a})$ for all $\mathbf{a} \in E^+$. (4) $\Rightarrow$ (5) is immediate as $\bigvee_{\mathbf{a} \in E^+} q(\mathcal{D}_{\mathbf{a}}, \mathbf{a})$ only contains symbols from $\mathrm{sig}(\mathcal{K})$. (4) $\Rightarrow$ (1) is trivial.

(1) $\Rightarrow$ (2). Suppose that $(\mathcal{K}, E^+, \{\mathbf{b}\})$ is separated by a constantless **FO**-formula φ. Then, there exists a model $\mathfrak{A}$ of $\mathcal{K}$ such that $\mathfrak{A} \not\models \varphi(\mathbf{b})$. Let $\mathbf{a} \in E^+$. Since $\mathfrak{A} \models \varphi(\mathbf{a})$ and since isomorphism implies logical equivalence, there is no model $\mathfrak{B}$ of $\mathcal{K}$ such that $(\mathfrak{B}, \mathbf{a}) \simeq_{\mathsf{rel}} (\mathfrak{A}, \mathbf{b})$. Then, to prove (1) $\Rightarrow$ (2), we suppose

$\neg$(2) and define a model $\mathfrak{B}$ of $\mathcal{K}$ such that $(\mathfrak{B},\mathbf{a}) \simeq_{\mathsf{rel}} (\mathfrak{A},\mathbf{b})$, contradicting (1). Suppose there exists $h : (\mathcal{D}_{\mathbf{a}},\mathbf{a}) \to (\mathfrak{A},\mathbf{b})$. Let $\mathfrak{B}$ be the model satisfying the following.

$$\mathfrak{B} : \begin{cases} \mathsf{dom}(\mathfrak{B}) = \mathsf{dom}(\mathfrak{A}) & \\ \quad R^{\mathfrak{B}} = R^{\mathfrak{A}} & \text{for every } R \in \mathsf{rel} \\ \quad c^{\mathfrak{B}} = c^{\mathfrak{A}} & \text{for every } c \in \mathsf{cons} \setminus \mathsf{cons}(\mathcal{D}_{\mathbf{a}}) \\ \quad c^{\mathfrak{B}} = h(c^{\mathfrak{A}}) & \text{for every } c \in \mathsf{cons}(\mathcal{D}_{\mathbf{a}}) \end{cases}$$

Note that the construction of $\mathfrak{B}$ relies on not making the UNA. Then, the identity map clearly witnesses $(\mathfrak{B},\mathbf{a}) \simeq_{\mathsf{rel}} (\mathfrak{A},\mathbf{b})$. It only remains to check that $\mathfrak{B} \models \mathcal{K}$. We get $\mathfrak{B} \models \mathcal{O}$ from $\mathfrak{A} \models \mathcal{O}$ and the isomorphism. To prove $\mathfrak{B} \models \mathcal{D}$, suppose $R(c_1,\dots,c_n) \in \mathcal{D}$. Then $\mathfrak{A} \models R(c_1^{\mathfrak{A}},\dots,c_n^{\mathfrak{A}})$. By definition of $\mathcal{D}_{\mathbf{a}}$, either $c_i \notin \mathsf{cons}(\mathcal{D}_{\mathbf{a}})$ for all i, or $c_i \in \mathsf{dom}(\mathcal{D}_{\mathbf{a}})$ for all i. Then by definition of $\mathfrak{B}$ and the fact that h is a homomorphism, it directly follows in each of the two cases that $\mathfrak{B} \models R(c_1^{\mathfrak{B}},\dots,c_n^{\mathfrak{B}})$. ⊣

The (1) $\Longleftrightarrow$ (5) equivalence immediately yields the following.

2.2. Corollary. *For all* **FO***-fragments* $\mathcal{L}_S,\mathcal{L}'_S$ *containing* **UCQ***,*
$\mathsf{sep}_w(\mathbf{FO},\mathcal{L}_S) = \mathsf{sep}^p_w(\mathbf{FO},\mathcal{L}_S) = \mathsf{sep}^p_w(\mathbf{FO},\mathcal{L}'_S) = \mathsf{sep}_w(\mathbf{FO},\mathcal{L}'_S)$.

As $q(\mathcal{D}_{\mathbf{a}},\mathbf{a})$ is rooted for any $\mathbf{a} \in E^+$, we also obtain the following corollary.

2.3. *Definition.* An **FO**-fragment $\mathcal{L}$ has the *relativization property* if for every $\mathcal{L}$-sentence φ and $A \in \mathsf{rel}_1 \setminus \mathsf{sig}(\varphi)$, there exists a sentence φ' such that for every model $\mathfrak{A}$ with $A^{\mathfrak{A}} \neq \emptyset$, $\mathfrak{A} \models \varphi'$ iff $\mathfrak{A}_{|A} \models \varphi$ where $\mathfrak{A}_{|A}$ is the $A^{\mathfrak{A}}$*-reduct* of $\mathfrak{A}$, *i.e.* the restriction of $\mathfrak{A}$ to domain $A^{\mathfrak{A}}$. For any set S of $\mathcal{L}$-sentences we write $S_{|A} = \{\varphi_{|A} : \varphi \in S\}$.

2.4. Proposition. *For all* **FO***-fragments* $\mathcal{L}_O,\mathcal{L}_S$ *such that* $\mathcal{L}_S$ *contains* **UCQ** *and* $\mathcal{L}_O$ *has the relativization property, weak (projective)* $(\mathcal{L}_O,\mathcal{L}_S)$*-separability with one negative example is mutually polynomial-time reducible with the complement of rooted* **UCQ***-evaluation on* $\mathcal{L}$*-knowledge bases.*

Proof. The reduction from separability to evaluation is immediate from (1) $\Longleftrightarrow$ (5) $\Longleftrightarrow$ (3) in Theorem 2.1. We show the converse direction. Let $\mathcal{K} = (\mathcal{O},\mathcal{D})$ be an $\mathcal{L}$-knowledge base, $q(\mathbf{x}) = \bigvee_{i\in I} q_i(\mathbf{x})$ a rooted UCQ with $\mathbf{x} = (x_1,\dots,x_n)$ and $\mathbf{a} \in \mathsf{cons}(\mathcal{D})^n$. For some $A \in \mathsf{rel}_1 \setminus (\mathsf{sig}(\mathcal{K}) \cup \mathsf{sig}(q))$, consider the relativization $\mathcal{O}_{|A}$ of $\mathcal{O}$ to A and $\mathcal{D}^{+A} = \mathcal{D} \cup \{A(c) \mid c \in \mathsf{cons}(\mathcal{D})\}$. Let $\mathcal{D}' = \mathcal{D}^{+A} \uplus \biguplus_{i\in I} \mathcal{D}_i$, where $(\mathcal{D}_i,[x_1],\dots,[x_n])$ is the pointed database associated with q_i (Rem 1.13). For the disjoint union, we write $[x_1]^i,\dots,[x_n]^i$ for the copy of $[x_1],\dots,[x_n]$ in the disjoint $\mathcal{D}_i$. Let $E^+ = \{([x_1]^i,\dots,[x_n]^i) \mid i \in I\}$. Let $\mathcal{K}' = (\mathcal{O}_{|A},\mathcal{D}')$. We show that $\mathcal{K} \not\models q(\mathbf{a})$ iff $(\mathcal{K}', E^+, \{\mathbf{a}\})$ is $\mathcal{L}_S$-separable. By Theorem 2.1, $(\mathcal{K}', E^+, \{\mathbf{a}\})$ is $\mathcal{L}_S$-separable iff $\mathcal{K}' \not\models \bigvee_{\mathbf{p}\in E^+} q(\mathcal{D}'_{\mathbf{p}}, \mathbf{p})(\mathbf{a})$. However, for all $\mathbf{p} \in E^+$ there exists $i \in I$ such that $q(\mathcal{D}'_{\mathbf{p}}, \mathbf{p}) = q(\mathcal{D}'_{([x_1]^i,\ldots,[x_n]^i)}, ([x_1]^i, \ldots, [x_n]^i))$. By the fact that $\mathcal{D}'$ is a disjoint union, $\mathcal{D}'_{([x_1]^i,\ldots,[x_n]^i)} = \mathcal{D}_i$. Then, modulo renaming of answer variables, $q(\mathcal{D}'_{([x_1]^i,\ldots,[x_n]^i)}, ([x_1]^i, \ldots, [x_n]^i)) = q(\mathcal{D}'_i, ([x_1]^i, \ldots, [x_n]^i)) = q_i$. Therefore, $(\mathcal{K}', E^+, \{\mathbf{a}\})$ is $\mathcal{L}_S$-separable iff $\mathcal{K}' \not\models q(\mathbf{a})$, so we ultimately need to show that $\mathcal{K} \not\models q(\mathbf{a})$ iff $\mathcal{K}' \not\models q(\mathbf{a})$.

($\Leftarrow$) Suppose there exists $\mathfrak{A} \models \mathcal{K}'$ such that $\mathfrak{A} \not\models q(\mathbf{a}^{\mathfrak{A}})$. Then $\mathfrak{A} \models \mathcal{O}_{|A}$ implies $\mathfrak{A}_{|A} \models \mathcal{O}$ and $\mathfrak{A} \models \mathcal{D}^{+A}$ implies $\mathfrak{A}_{|A} \models \mathcal{D}$, so $\mathfrak{A}_{|A} \models \mathcal{K}$. Then, since $\mathfrak{A}_{|A}$ is a submodel of $\mathfrak{A}$, the implication $\mathfrak{A} \not\models q(\mathbf{a}^{\mathfrak{A}}) \Rightarrow \mathfrak{A}_{|A} \not\models q(\mathbf{a}^{\mathfrak{A}})$ is immediate.

($\Rightarrow$) Conversely, assume that $\mathcal{K} \not\models q(\mathbf{a})$. Let $\mathfrak{A} \models \mathcal{K}$ with $\mathfrak{A} \not\models q(\mathbf{a}^{\mathfrak{A}})$. As $A \notin \mathsf{sig}(\mathcal{K})$ we may assume $A^{\mathfrak{A}} = \mathsf{dom}(\mathfrak{A})$. As $\mathcal{D}'$ is a disjoint union, we can then expand $\mathfrak{A}$ into a model $\mathfrak{B}$ of $\mathcal{K}'$ in which $q(\mathbf{a})$ is still not satisfied, with $\mathfrak{B} = \mathfrak{A} \uplus \biguplus_i \mathfrak{A}_{\mathcal{D}_i}$, where $\mathfrak{A}_{\mathcal{D}_i}$ is $\mathcal{D}_i$ seen as a model (Rem. 1.10). Let $c^{\mathfrak{B}} = c^{\mathfrak{A}}$ for $c \in \mathsf{cons}(\mathcal{D})$ and $c^{\mathfrak{B}} = c^{\mathfrak{A}_{\mathcal{D}_i}}$ for $c \in \mathsf{cons}(\mathcal{D}_i)$. Clearly $\mathfrak{B} \models \mathcal{D}'$. Then, $\mathfrak{A}_{|A} \models \mathcal{O}$ and $\mathfrak{A}_{|A} = \mathfrak{B}_{|A}$ since $A \notin \mathsf{sig}(q)$, so $\mathfrak{B}_{|A} \models \mathcal{O}$. Since q is rooted (hence has only one connected component) and $a_1, \ldots, a_n \in \mathsf{cons}(\mathcal{D}^{+A})$, we have $\mathfrak{B} \models q(\mathbf{a}^{\mathfrak{B}})$ iff $\mathfrak{B}_{|A} \models q(\mathbf{a}^{\mathfrak{B}})$. By definition $\mathfrak{B}_{|A} \models q(\mathbf{a}^{\mathfrak{B}})$ iff $\mathfrak{A} \models q(\mathbf{a}^{\mathfrak{A}})$, which concludes the proof. ⊣

2.5. *Remark.* We use $\mathcal{O}_{|A}$ and $\mathcal{D}^{+A}$ because $(\mathcal{O}, \mathcal{D}_i)$ could be unsatisfiable. By relativizing, we make sure that the ontology axioms only apply to constants in $\mathcal{D}^{+A}$.

The Guarded Negation Fragment **GNF** (§1.52) of first-order logic contains **UCQ**, so Corollary 2.4 implies the following.

2.6. Corollary. *For any* **FO**-*fragment* $\mathcal{L}$ *containing* **UCQ**, *in combined complexity, weak (projective)* (**GNF**, $\mathcal{L}$)*-separability is* 2EXP*-complete, and weak (projective)* $(\emptyset, \mathcal{L})$*-separability is* CONP*-complete.*

Proof. Immediate from Corollary 2.4, as (1) UCQ-evaluation on **GNF**-knowledge bases is in 2EXP (thus rooted UCQ-evaluation is) and **GNF**-satisfiability is 2EXP-hard [BCS15]: recall that satisfiability is reducible to separability (§1.6), (2) rooted UCQ-evaluation on knowledge bases with empty ontologies is NP-complete. ⊣

The (1) $\Leftrightarrow$ (3) equivalence from Theorem 2.1 also provides the following corollary for free.

2.7. *Definition.* We say $(\mathcal{L}_O, \mathcal{L}_S)$-separability is *anti-monotone in the ontology* if for all $\mathcal{L}_O$-ontologies $\mathcal{O}_1, \mathcal{O}_2$, any database $\mathcal{D}$ and all sets E^+, E^- of examples, it holds that if $\mathcal{O}_1 \subseteq \mathcal{O}_2$, then $((\mathcal{O}_2, \mathcal{D}), E^+, E^-)$ $\mathcal{L}_S$-separable implies $((\mathcal{O}_1, \mathcal{D}), E^+, E^-)$ $\mathcal{L}_S$-separable.

2.8. Corollary. *For any* **FO**-*fragment* $\mathcal{L}$ *containing* **UCQ**, *(projective)* $(\mathbf{FO}, \mathcal{L})$-*separability is anti-monotone in the ontology.*

§ 2.2. $\mathbf{FO}^2$-ONTOLOGIES

As opposed to **FO**, the satisfiability problem in $\mathbf{FO}^2$ has been shown to be decidable in [Mo75], and NEXP-complete in [GKV97]. Separability for $\mathbf{FO}^2$-knowledge bases is then potentially decidable. We show that full weak $(\mathbf{FO}^2, \mathbf{FO}^2)$ and $(\mathbf{FO}^2, \mathbf{FO})$-separability are still undecidable, both in the projective and non-projective case (Thm. 2.12). This is done via reduction from a tiling problem, without using any model-theoretic characterisation of separability. Recall that for $(\mathbf{FO}^2, \mathbf{FO})$ there is no difference between the projective and non-projective case (Cor. 2.2). We show that is not true with $\mathbf{FO}^2$ as a separating language (Ex. 2.10). However, we show that even with the help of projectivity, $\mathbf{FO}^2$ still has strictly less separating power than **FO** (Thm. 2.9). Rather than showing it through a counterexample, we use model-theoretic arguments to show that it not only applies to $\mathbf{FO}^2$ but to any fragment of **FO** with the finite model property, the relativization property and on which evaluating UCQs is not finitely controllable (Lem. 2.11).

2.2.1. *Separating power*

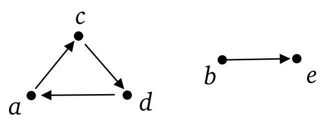
Results relative to the separating power of $\mathbf{FO}^2$ are contained in the following theorem, which contrasts with Corollary 2.2 in two ways: not only does projectivity make a difference for $\mathbf{FO}^2$, but it still does not allow $\mathbf{FO}^2$ to match **FO** in separating power.

2.9. Theorem. $\mathsf{sep}_w(\mathbf{FO}^2, \mathbf{FO}^2) \subsetneq \mathsf{sep}^p_w(\mathbf{FO}^2, \mathbf{FO}^2) \subsetneq \mathsf{sep}_w(\mathbf{FO}^2, \mathbf{FO})$.

Proof. Strictness of the first inclusion is an immediate consequence of Example 2.10 below, as $(\mathcal{K}, \{a\}, \{b\})$ is not non-projectively $\mathbf{FO}^2$-separable. As for the second inclusion, $\mathsf{sep}^p_w(\mathbf{FO}^2, \mathbf{FO}^2) \subseteq \mathsf{sep}_w(\mathbf{FO}^2, \mathbf{FO})$ follows from $\mathsf{sep}^p_w(\mathbf{FO}^2, \mathbf{FO}) = \mathsf{sep}_w(\mathbf{FO}^2, \mathbf{FO})$ (Cor. 2.2) and $\mathsf{sep}^p_w(\mathbf{FO}^2, \mathbf{FO}^2) \neq \mathsf{sep}^p_w(\mathbf{FO}^2, \mathbf{FO})$ follows from Lemma 2.11, as $\mathbf{FO}^2$ has the finite model property [Mo75] and the relativization property, but rooted UCQ evaluation on $\mathbf{FO}^2$-knowledge bases is not finitely controllable (Ex. 2.15). ⊣

The following example implies strictness of the first inclusion.

2.10. *Example.* Let $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ with $\mathcal{O}$ consisting of $\forall x \exists y \exists z(R(x,y) \wedge R(z,x))$ and $\forall x \forall y(R(x,y) \rightarrow \neg R(y,x))$, and $\mathcal{D} = \{R(a,c), R(c,d), R(d,a), R(b,e)\}$.



Database $\mathcal{D}$.

The labeled $\mathbf{FO}^2$-knowledge base $(\mathcal{K}, \{a\}, \{b\})$ is projectively $\mathcal{ALCI}$ (and thus $\mathbf{FO}^2$)-separable by $C = A \rightarrow \exists R.\exists R.\exists R.A$ that uses the concept name A as a helper symbol, but not non-projectively $\mathbf{FO}^2$-separable, since every non-trivial $\mathbf{FO}^2$-formula $\varphi(x)$ with $\mathrm{sig}(\varphi) = \{R\}$ is equivalent to $x = x$ or $\neg(x = x)$ w.r.t. $\mathcal{O}$: the formula $\exists y R(x,y)$ or $\exists y R(y,x)$ is already entailed by the ontology. For every non-trivial $\mathbf{FO}^2$-formula $\varphi(x)$ with $\mathrm{sig}(\varphi) = \{R\}$ that is equivalent to $x = x$ or $\neg(x = x)$ w.r.t. $\mathcal{O}$ and whose variables are w.l.o.g amongst $\{x, y\}$, the formulas $\exists x \varphi(x)$ and $\exists y \varphi(x)$ are also equivalent to $x = x$ or $\neg(x = x)$ w.r.t. $\mathcal{O}$.

To illustrate the role of the second sentence in $\mathcal{O}$, let $\mathcal{O}^-$ be $\mathcal{O}$ without it. Then $((\mathcal{O}^-, \mathcal{D}), \{a\}, \{b\})$ is separated by the $\mathbf{FO}^2$-sentence obtained from the separating $\mathcal{ALCI}$-concept C above by replacing each occurrence of $A(x)$ in the first-order translation $C^\dagger$ by $\exists y(R(x,y) \wedge x \neq y \wedge R(y,y))$.
The following lemma implies strictness of the second inclusion.

2.11. Lemma. *Let $\mathcal{L}$ be a fragment of* **FO** *such that*

1. *$\mathcal{L}$ has the relativization property,*
2. *$\mathcal{L}$ has the finite model property,*
3. $\mathsf{sep}^p_w(\mathcal{L}, \mathbf{FO}) = \mathsf{sep}^p_w(\mathcal{L}, \mathcal{L})$.

Then rooted UCQ-evaluation on $\mathcal{L}$-knowledge bases is finitely controllable.

Proof. Assume that evaluating rooted UCQs on $\mathcal{L}$-knowledge bases is not finitely controllable, *i.e.* there is an $\mathcal{L}$-knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{D})$, a rooted UCQ $q(\mathbf{x}) = \bigvee_{i \in I} q_i(\mathbf{x})$, $\mathbf{x} = (x_1, \ldots, x_n)$, and $\mathbf{a} \in \mathrm{cons}(\mathcal{D})^{|\mathbf{x}|}$ such that $\mathcal{K} \not\models q(\mathbf{a})$, but $\mathfrak{B} \models q(\mathbf{a}^{\mathfrak{B}})$ for all finite models $\mathfrak{B}$ of $\mathcal{K}$. Consider, for some fresh concept name A, the labeled knowledge base $(\mathcal{K}', E^+, \{\mathbf{a}\})$ defined in Corollary 2.4. It was proved in that same corollary that $q(\mathbf{x})$ separates $(\mathcal{K}', E^+, \{\mathbf{a}\})$. Suppose there is an $\mathcal{L}$-formula $\varphi(\mathbf{x})$ that separates $(\mathcal{K}', E^+, \{\mathbf{a}\})$. Since $\mathcal{L}$ has the finite model property, there exists a finite model $\mathfrak{A}_f$ of $\mathcal{K}'$ such that $\mathfrak{A}_f \models \neg\varphi(\mathbf{a}^{\mathfrak{A}_f})$. As $\mathfrak{B} \models q(\mathbf{a}^{\mathfrak{B}})$ for all finite models $\mathfrak{B}$ of $(\mathcal{O}, \mathcal{D})$ thus all finite models of $(\mathcal{O}_{|A}, \mathcal{D}^{+A})$, there exists $i \in I$ with $\mathfrak{A}_f \models q_i(\mathbf{a}^{\mathfrak{A}_f})$. That is witnessed by a homomorphism $h : (\mathcal{D}_i, ([x_1]^i, \ldots, [x_n]^i)) \rightarrow (\mathfrak{A}_f, \mathbf{a}^{\mathfrak{A}_f})$. Let $\mathfrak{A}'_f$ be a model which coincides with $\mathfrak{A}_f$ except that the constants c in $\mathcal{D}_i$ are interpreted as $h(c)$. Then $\mathfrak{A}'_f \models \mathcal{K}'$

and $\mathfrak{A}'_f \models \neg\varphi(([x_1]^i, \ldots, [x_n]^i)^{\mathfrak{A}_f})$, which contradicts the assumption that $\varphi(\mathbf{x})$ separates $(\mathcal{K}', E^+, \{\mathbf{a}\})$. ⊣

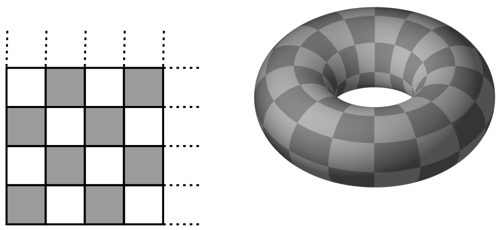
2.2.2. *Complexity*

This subsection is dedicated to proving the following theorem.

2.12. Theorem. *Full weak (projective)* $(\mathbf{FO}^2, \mathcal{L}_S)$*-separability is undecidable for any* **FO***-fragment* $\mathcal{L}_S$ *containing* $\mathbf{FO}^2$.

The proof is by reduction from tiling problems introduced next.

2.13. *Definition.* We call *square tiling system* any triple $\tau = (T, H, V)$ with T a finite set (of tiles) and $H, V \subseteq T \times T$. An $\mathbb{N}^2$*-solution* to τ is a function $\sigma : \mathbb{N}^2 \to T$ such that $(\sigma(i,j), \sigma(i+1,j)) \in H$ and $(\sigma(i,j), \sigma(i,j+1)) \in V$ for all $i, j \geqslant 0$. An $\mathbb{N}^2$-solution σ is *periodic* if there exist $h, v \geqslant 1$ such that $\sigma(i,j) = \sigma(i+h,j) = s(i,j+v)$, for all $i, j \geqslant 0$. A periodic $\mathbb{N}^2$-solution can be thought as a torus tiling with square tiles, *i.e.* assuming that the domain of σ is $\mathbb{Z}/h\mathbb{Z} \times \mathbb{Z}/v\mathbb{Z}$.



A periodic $\mathbb{N}^2$*-solution for* $T = \{t_1, t_2\}, H = V = \{(t_1, t_2), (t_2, t_1)\}$
and its toric interpretation.

It is well-known that the problem of deciding whether a given square tiling system admits an $\mathbb{N}^2$*-solution* (resp. periodic) is undecidable [B66]. However, we are going to exploit a stronger undecidability result due to [GK72] (see also [BGG97, Thm 3.1.7] for a new proof). Recall that two sets A, B are *recursively inseparable* if there is no decidable set that contains A and is disjoint from B. The result is as follows.

2.14. Theorem. *The set of square tiling systems that admit no* $\mathbb{N}^2$-solution *is recursively inseparable from the set of square tiling systems that admit a periodic* $\mathbb{N}^2$-solution.

We are now ready to prove Theorem 2.12.

Proof of Theorem 2.12. Given a square tiling system $\tau = (T, H, V)$, we construct a labeled $\mathbf{FO}^2$-knowledge base $(\mathcal{K}_\tau, \{a\}, \{b\})$ with $\mathcal{K}_\tau = (\mathcal{O}_\tau, \mathcal{D})$ as follows.

$$\mathcal{O}_\tau = \Big\{ \forall x\, (B(x) \to (\exists y\, (R_v(x,y) \wedge B(y))) \wedge (\exists y\, (R_h(x,y) \wedge B(y)))), \quad (2.1)$$

$$\forall xy\, (B(x) \wedge B(y) \to U(x,y)), \quad (2.2)$$

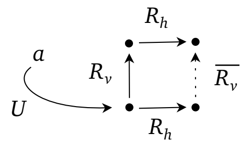
$$\forall xy\, (\neg R_v(x,y) \to \overline{R}_v(x,y)), \quad (2.3)$$

$$\forall x \bigvee_{t \in T} (A_t(x) \wedge \bigwedge_{t' \in T \setminus \{t\}} \neg A_{t'}(x)), \quad (2.4)$$

$$\forall xy\, (R_v(x,y) \to \bigvee_{(t,t') \in V} A_t(x) \wedge A_{t'}(y)), \quad (2.5)$$

$$\forall xy\, (R_h(x,y) \to \bigvee_{(t,t') \in H} A_t(x) \wedge A_{t'}(y)) \Big\} \quad (2.6)$$

$$\mathcal{D} = \big\{ U(a, a_1), R_v(a_1, a_2), R_h(a_2, a_3), R_h(a_1, a_4), \overline{R}_v(a_4, a_3), B(b) \big\}$$



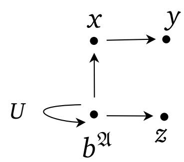
Connected component of a in $\mathcal{D}$.

By Theorem 2.14, it suffices to prove that for any **FO**-fragment $\mathcal{L}_S \supseteq \mathbf{FO}^2$,

1. If $(\mathcal{K}_\tau, \{a\}, \{b\})$ is (projectively) $\mathcal{L}_S$-separable, then τ admits a solution.
2. If τ admits a periodic solution, then $(\mathcal{K}_\tau, \{a\}, \{b\})$ is non-projectively $\mathcal{L}_S$-separable.

Indeed, if that is the case, then $\{\tau \mid (\mathcal{K}_\tau, \{a\}, \{b\}) \text{ separable}\}$ contains the set of tiling systems admitting a periodic solution and is contained in the set of tiling systems admitting a solution. That set is reduced to the set of separable $\mathcal{L}$-labeled knowledge bases by the computable function $\tau \mapsto (\mathcal{K}_\tau, \{a\}, \{b\})$, hence the undecidability result.

Proof of (1). For Point 1, suppose that $(\mathcal{K}_\tau, \{a\}, \{b\})$ is projectively or non-projectively $\mathcal{L}$-separable. Then, $(\mathcal{K}_\tau, \{a\}, \{b\})$ is **FO**-separable. By Theorem 2.1, it suffices to verify that if $\mathcal{K}_\tau \not\models q(\mathcal{D}_a, a)(b)$, then (T, H, V) admits a solution. Let $\mathfrak{A}$ be a model witnessing $\mathcal{K}_\tau \not\models q(\mathcal{D}_a, a)(b)$. From axioms (2.1)-(2.3) and $B(b) \in \mathcal{D}$, it follows that $\mathfrak{A}$ contains the submodel depicted below (modulo homomorphism). Then, $\mathfrak{A} \not\models q(\mathcal{D}_a, a)(b^{\mathfrak{A}})$ implies that y is an R_v-successor of z, otherwise by axiom (2.3) we have $\mathfrak{A} \models \overline{R}_v(z, y)$ and thus $\mathfrak{A} \models q(\mathcal{D}_a, a)(b^{\mathfrak{A}})$. Repeating this argument we can show $\mathfrak{A}$ contains (the homomorphic image of) an infinite R_h/R_v grid : every point in the grid (starting with the square of bottom-left corner $b^{\mathfrak{A}}$) is a U-successor of $b^{\mathfrak{A}}$ as it satisfies B (axiom (2.2)), thus is the bottom-left corner of an R_h/R_v square, by the same argument we used on $b^{\mathfrak{A}}$. Since $\mathfrak{A}$ is a model of formula (2.4) every element in the grid is labeled with A_t for a unique element $t \in T$. Finally, since $\mathfrak{A}$ is a model of formulas (2.5) and (2.6), the relations H and V are respected along R_h and R_v, respectively.



Proof of (2). Suppose that (T, H, V) admits a periodic solution σ with periods $h, v \geqslant 1$. We show that $(\mathcal{K}_\tau, \{a\}, \{b\})$ is non-projectively **FO**^{2}-separable under the assumption that $\mathcal{K}_\tau$ mentions a binary relation symbol S. This is without loss of generality, as we can include $\forall xy\, S(x,y) \to S(x,y)$ in $\mathcal{O}_\tau$. Let π be a bijection from $[h] \times [v]$ to $[hv]$ and let C_{ij} be the $\mathcal{ALCI}$-concept (corresponding to an **FO**^{2}-formula) expressing that there is an S-path of length $\pi(i,j)$[1]. Consider the following **FO**^{2}-formula, also written as an $\mathcal{ALCI}$-concept.

$$\varphi_{hv}(x) = \exists U. \bigsqcap_{i \in [h]} \bigsqcap_{j \in [v]} (\forall R_v . \forall R_h . C_{ij} \to \exists R_h . \exists \overline{R}_v . C_{ij})$$

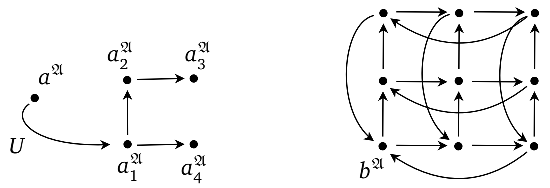
It should be clear that $\mathcal{K}_\tau \models \varphi_{hv}(a)$ since already $q(\mathcal{D}_a, a)(x) \models \varphi_{hv}(x)$. To see that $\mathcal{K}_\tau \not\models \varphi_{hv}(b)$, we construct a (finite) model $\mathfrak{A}$ witnessing that. Informally, $\mathfrak{A}$ consists of two disconnected parts. One part is $\mathcal{D}$ viewed as a model; the other is an $h \times v$-torus over binary symbols R_v, R_h in which each element has an outgoing S-path. More precisely, the torus has domain $[h] \times [v]$ and each element (i,j) is labeled with the unary symbol $A_{\sigma(i,j)}$ and has an outgoing S-path of length $\pi(i,j)$. In more details,

$$\begin{aligned}
B^{\mathfrak{A}} &= [h] \times [v] \\
R_v^{\mathfrak{A}} &= \big\{(a_1, a_2)\big\} \cup \\
&\quad \big\{((i,j),(i, j \oplus_v 1)) \mid i \in [h], j \in [v]\big\} \\
R_h^{\mathfrak{A}} &= \big\{(a_2, a_3), (a_1, a_4)\big\} \cup \\
&\quad \big\{((i,j),(i \oplus_h 1, j)) \mid i \in [h], j \in [v]\big\} \\
A_t^{\mathfrak{A}} &= \big\{(i,j) \in [h] \times [v] \mid \sigma(i,j) = t\big\} && \text{for all } t \in T \\
U^{\mathfrak{A}} &= \big\{(a, a_1)\big\} \cup ([h] \times [v]) \times ([h] \times [v]) \\
b^{\mathfrak{A}} &= (0,0), a^{\mathfrak{A}} = a, a_i^{\mathfrak{A}} = a_i && \text{for } i \in \{1,2,3,4\}
\end{aligned}$$

and $S^{\mathfrak{A}}$ is as described above and $\overline{R}_v^{\mathfrak{A}}$ is the complement of $R_v^{\mathfrak{A}}$.

[1] In the projective case, we could simply take fresh concepts C_{ij} instead of those S-paths.

It is readily checked that $\mathfrak{A}$ is a model of $\mathcal{K}$. Suppose that $\mathfrak{A} \models \varphi_{hv}(b^{\mathfrak{A}})$, let (i_0, j_0) be the U-successor of $b^{\mathfrak{A}}$ that witnesses the conjunction in φ_{hv}, and let $i = i_0 \oplus_h 1$ and $j = j_0 \oplus_v 1$. By construction of $\mathfrak{A}$, it holds that $\mathfrak{A} \models \forall R_v.\forall R_h.C_{ij}(i_0, j_0)$ and $\mathfrak{A} \not\models \exists R_h.\exists \overline{R}_v.C_{ij}(i_0, j_0)$, in contradiction to the implication in φ_{hv}. Hence, $\mathfrak{A} \not\models \varphi_{hv}(b^{\mathfrak{A}})$. ⊣



A depiction of $\mathfrak{A}$ with $h = v = 3$ (S- and U-relations omitted on the right).

2.15. The knowledge base used in the previous proof can be used to show that evaluating rooted CQs (and therefore arbitrary (U)CQs) on $\mathbf{FO}^2$-knowledge bases is not finitely controllable. Let τ be any square tiling system that admits an $\mathbb{N}^2$-solution but no periodic one. Recall $\mathcal{O}_\tau$ and $\mathcal{D}$ from the proof of Theorem 2.12. As mentioned in the proof, τ admitting a solution implies $(\mathcal{O}_\tau, \mathcal{D}) \not\models q(\mathcal{D}_a, a)(b)$, thus $(\mathcal{O}_\tau, \{B(b)\}) \not\models q(\mathcal{D}_a, a)(b)$ since $B(b) \in \mathcal{D}$. but $(\mathcal{O}_\tau, \{B(b)\}) \models_{\mathsf{fin}} q(\mathcal{D}_a, a)(b)$.

§ 2.3. PROJECTIVE CASE IN $\mathbf{DL}_{\mathcal{IQ}}$

We consider $(\mathcal{L}, \mathcal{L})$-separability for $\mathcal{L} \in \mathbf{DL}_{\mathcal{IQ}}$, where $\mathbf{DL}_{\mathcal{IQ}}$ denotes the set of extensions of $\mathcal{ALC}$ using the constructors $\mathcal{I}, \mathcal{Q}$. With bisimulations, it is possible to characterise separability in a rather uniform fashion over these languages, thanks to them admitting only slightly different notions of bisimulation. As a first step, a bisimulation-based characterisation (Thm. 2.16) is established. From that first one follows another characterisation, based on what we define as "$\mathcal{L}$-simulations" (Lem. 2.18). Finally, we reach the final (homomorphism-based) characterisation in Theorem 2.28, with the crucial help of projectivity (Lem. 2.27). One can recognize a similar characterisation to the one shown in Theorem 2.1 for **FO**. Then, as does its **FO** counterpart, Theorem 2.28 reveals a connection between the decision problems of separability and UCQ evaluation, from which we deduce complexity bounds for full signature weak projective $(\mathcal{L}, \mathcal{L})$-separability, $\mathcal{L} \in \mathbf{DL}_{\mathcal{IQ}}$. Below is a summary of this section's complexity results, in combined and data complexity. In any given cell, by a complexity class C we mean that full signature weak projective $(\mathcal{L}, \mathcal{L})$-separability is C-complete.

	Combined	Data
$\mathcal{ALC}$	NExp (Thm 2.54)	PSpace (Thm 2.64)
$\mathcal{ALCI}$	NExp (Cor 2.33)	NExp [JLPW20]
$\mathcal{ALCQ}$	NExp (Thm 2.68)	?
$\mathcal{ALCQI}$	Exp (Thm 2.35)	?

Our results on $\mathcal{ALCQ}$ and $\mathcal{ALCQI}$ rely on the Unique Name Assumption (UNA). For $\mathcal{ALC}, \mathcal{ALCI}$, the UNA has no impact as $\mathcal{K} \models C(a)$ with UNA iff $\mathcal{K} \models C(a)$ without UNA for any knowledge base $\mathcal{K}$, concept C and constant a.

2.3.1. *Intermediate characterisations*

Recall that all the DLs we consider each admit a notion of bisimulation that characterises logical equivalence (§1.4). We use that to give a first semantic characterisation of full weak projective $(\mathcal{L}, \mathcal{L})$-separability for each $\mathcal{L}$.

2.16. Theorem. *Let $(\mathcal{K}, E^+, \{b\})$ be a labeled $\mathcal{ALCQI}$-knowledge base with $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ and $\mathcal{L} \in \mathbf{DL}_{\mathcal{IQ}}$. The following conditions are equivalent.*

1. *There exists an $\mathcal{L}$-forest model $\mathfrak{A} \models \mathcal{K}$ of finite outdegree such that $(\mathfrak{B}, a^{\mathfrak{B}}) \not\sim_{\mathcal{L}} (\mathfrak{A}, b^{\mathfrak{A}})$ for all $\mathfrak{B} \models \mathcal{K}, a \in E^+$.*
2. *There exists $\mathfrak{A} \models \mathcal{K}$ of finite outdegree such that $(\mathfrak{B}, a^{\mathfrak{B}}) \not\sim_{\mathcal{L}} (\mathfrak{A}, b^{\mathfrak{A}})$ for all $\mathfrak{B} \models \mathcal{K}, a \in E^+$.*
3. *$(\mathcal{K}, E^+, \{b\})$ is weakly projectively $\mathcal{L}$-separable.*

Proof. (1) $\Rightarrow$ (2). Immediate.

(3) $\Rightarrow$ (1). Suppose there exists an $\mathcal{L}$-concept C such that $\mathcal{K} \models C(a)$ for all $a \in E^+$ and $\mathcal{K} \not\models C(b)$. Then there exists a model $\mathfrak{A}$ of $\mathcal{K}$ such that $\mathfrak{A} \not\models C(b^{\mathfrak{A}})$. By Lemma 1.42, $\mathfrak{A}$ can be assumed to be an $\mathcal{L}$-forest model of $\mathcal{D}$ of finite outdegree. For all $a \in E^+$ and any model $\mathfrak{B}$ of $\mathcal{K}$ we have $\mathfrak{B} \models C(a^{\mathfrak{B}})$. Then, for all $a \in E^+$, we have $(\mathfrak{B}, a^{\mathfrak{B}}) \not\equiv_{\mathcal{L}} (\mathfrak{A}, b^{\mathfrak{A}})$ and, by Lemma 1.25, $(\mathfrak{B}, a^{\mathfrak{B}}) \not\sim_{\mathcal{L}} (\mathfrak{A}, b^{\mathfrak{A}})$.

(2) $\Rightarrow$ (3). Assume there exists a model $\mathfrak{A}$ of $\mathcal{K}$ (of finite outdegree) such that for all $a \in E^+$ and all models $\mathfrak{B}$ of $\mathcal{K}$, $(\mathfrak{A}, b^{\mathfrak{A}}) \not\sim_{\mathcal{L}} (\mathfrak{B}, a^{\mathfrak{B}})$. We prove that there exists $\{C_1, \dots, C_n\} \subseteq \mathrm{tp}^{\mathfrak{A}}_{\mathcal{L}}(b^{\mathfrak{A}})$ such that $\mathcal{K} \models \bigsqcup_{1 \leqslant i \leqslant n} \neg C_i(a)$ for all $a \in E^+$. Then we are done as $\mathcal{K} \not\models \bigsqcup_{1 \leqslant i \leqslant n} \neg C_i(b)$. For a proof by contradiction, suppose this is not the case. Then, for some $a_0 \in E^+$, every finite subset $\{C_1, \dots, C_n\}$ of $\mathrm{tp}^{\mathfrak{A}}_{\mathcal{L}}(b^{\mathfrak{A}})$ satisfies $\mathcal{K} \not\models \bigsqcup_{1 \leqslant i \leqslant n} \neg C_i(a_0)$. Then, for all finite $\{C_1, \dots, C_n\} \subseteq \mathrm{tp}^{\mathfrak{A}}_{\mathcal{L}}(b^{\mathfrak{A}})$, $\big(\mathcal{O}, \mathcal{D} \cup \{C_1(a_0), \dots, C_n(a_0)\}\big)$ is satisfiable. By compactness (Thm 1.4), $\big(\mathcal{O}, \mathcal{D} \cup \{C(a_0) \mid C \in \mathrm{tp}^{\mathfrak{A}}_{\mathcal{L}}(b^{\mathfrak{A}})\}\big)$ is satisfiable. But any model $\mathfrak{B}$ of $\{C(a_0) \mid C \in \mathrm{tp}^{\mathfrak{A}}_{\mathcal{L}}(b^{\mathfrak{A}})\}$ satisfies $(\mathfrak{B}, a_0^{\mathfrak{B}}) \equiv_{\mathcal{L}} (\mathfrak{A}, b^{\mathfrak{A}})$. As $\mathfrak{A}$ has finite outdegree, this implies, by Lemma 1.25, $(\mathfrak{B}, a_0^{\mathfrak{B}}) \sim_{\mathcal{L}} (\mathfrak{A}, b^{\mathfrak{A}})$, which is the desired contradiction. ⊣

We now introduce simulations for an intermediate characterisation of separability that does not depend on the relation between two models $\mathfrak{A}$ and $\mathfrak{B}$ like in the previous theorem, but simply between the database and one model. From there it will be easy to move on to a homomorphism-based characterisation (and thus to a "query" one).

2.17. *Definition.* Let $S \subseteq \mathsf{cons}(\mathcal{D}) \times \mathfrak{A}$ for some database $\mathcal{D}$ and model $\mathfrak{A}$. Let Σ be a relational signature. We define the following conditions. Suppose $(c, x) \in S$, then

ATOMR	$A(c) \in \mathcal{D} \Rightarrow x \in A^{\mathfrak{A}}$ for all concept names $A \in \Sigma$.
FORTH	Same as in the definition of bisimulations, cf. Def 1.24.
QFORTH	Same as in the definition of bisimulations, cf. Def 1.24.
BISIM$_{\mathcal{L}}$	For all y, $(c, y) \in S \Rightarrow (\mathfrak{A}, x) \sim_{\mathcal{L}} (\mathfrak{A}, y)$.

For any $a \in \mathsf{cons}(\mathcal{D})$, let $\mathcal{D}_a^{\uparrow}$ be the directedly connected component of a in $\mathcal{D}$, and $\mathcal{K}_{\mathfrak{A},S} := \big(\mathcal{O}, \mathcal{D} \cup \{C(c) \mid C \in \mathsf{sub}(\mathcal{K}), c \in \mathsf{cons}(\mathcal{D}_a^{\uparrow}), (c, d) \in S, d \in C^{\mathfrak{A}}\}\big)$.

If S satisfies ATOMR, we say S is an

- $\mathcal{ALCI}(\Sigma)$-simulation if S satisfies FORTH over rel_2^- and BISIM$_{\mathcal{ALCI}}$,
- $\mathcal{ALCQI}(\Sigma)$-simulation if S satisfies QFORTH over rel_2^- and BISIM$_{\mathcal{ALCQI}}$.

If, additionally, $S \subseteq \mathsf{cons}(\mathcal{D}_a^{\uparrow}) \times \mathsf{dom}(\mathfrak{A})$ and $\mathcal{K}_{\mathfrak{A},S} \not\models \bot$, we say S is an

- $\mathcal{ALC}(\Sigma)$-simulation if S satisfies FORTH over rel_2 and BISIM$_{\mathcal{ALC}}$,
- $\mathcal{ALCQ}(\Sigma)$-simulation if S satisfies QFORTH over rel_2 and BISIM$_{\mathcal{ALCQ}}$.

If $\mathsf{rel}_1 \cup \mathsf{rel}_2 \subseteq \Sigma$, we write $\mathcal{L}$-simulation for $\mathcal{L}(\Sigma)$-simulation. If $(c, x) \in S$ for some such S with respect to $\mathcal{L}$, we write $(\mathcal{D}, c) \preceq_{\mathcal{L}} (\mathfrak{A}, x)$.

2.18. Lemma. *Let $\mathcal{L} \in \mathbf{DL}_{\mathcal{IQ}}$. Let $(\mathcal{K}, E^+, \{b\})$ be a labeled $\mathcal{L}$-knowledge base, with $\mathcal{K} = (\mathcal{O}, \mathcal{D})$. The following conditions are equivalent for any model $\mathfrak{A} \models \mathcal{K}$ of finite outdegree.*

1. *There exist $a \in E^+$, $\mathfrak{B} \models \mathcal{K}$ such that $(\mathfrak{B}, a^{\mathfrak{B}}) \sim_{\mathcal{L}} (\mathfrak{A}, b^{\mathfrak{A}})$.*
2. *There exists $a \in E^+$ such that $(\mathcal{D}, a) \preceq_{\mathcal{L}} (\mathfrak{A}, b^{\mathfrak{A}})$.*

2.19. *Why we need $\mathcal{K}_{\mathfrak{A},S}$.* To show why we need the additional condition involving $\mathcal{K}_{\mathfrak{A},S}$ when inverse roles are absent, we consider this example where that condition is not satisfied and show that, in that case, condition (1) of Lemma 2.18 does not hold. In other words, without that condition, simulations would not capture separability. Let $\mathcal{K} = (\mathcal{O}, \mathcal{D})$, where $\mathcal{O} = \{\top \sqsubseteq \forall R.A\}$ and $\mathcal{D} = \{R(c, a), B(b)\}$. Clearly, for every model $\mathfrak{A}$ of $\mathcal{K}$, the relation $\{(a, b^{\mathfrak{A}})\}$ between $\mathsf{cons}(\mathcal{D})$ and $\mathsf{dom}(\mathfrak{A})$ satisfies ATOMR, FORTH and BISIM$_{\mathcal{ALC}}$, since a has no atom and no successor in $\mathcal{D}$. However, $(\mathcal{K}, \{a\}, \{b\})$ is clearly weakly

$\mathcal{ALC}$-separable by A, *i.e.* condition (1) is not met. $\mathcal{K}_{A,S}$ is not satisfiable for any $\mathfrak{A}$ in which $b^{\mathfrak{A}} \notin A^{\mathfrak{A}}$ and any S that contains $(a, b^{\mathfrak{A}})$.

2.20. *Why we need* $\text{BISIM}_{\mathcal{L}}$. We illustrate the need for $\text{BISIM}_{\mathcal{L}}$ with $\mathcal{L} = \mathcal{ALCI}$ as example. Denote $\mathcal{ALCI}$-simulations without $\text{BISIM}_{\mathcal{ALCI}}$ by $\preceq^{-}_{\mathcal{ALCI}}$. Let $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ where $\mathcal{O} = \{\top \sqsubseteq \exists R.\top \sqcap \exists R^{-}.\top,\ B \sqsubseteq \forall R.\forall R.\forall R.\neg B\}$ and $\mathcal{D} = \{R(a, a'),\ R(a', a''),\ R(a'', a),\ B(b)\}$. Then $(\mathcal{D}, a) \preceq^{-}_{\mathcal{ALCI}} (\mathfrak{A}, b)$ for all $\mathfrak{A} \models \mathcal{K}$ but $(\mathcal{K}, \{a\}, \{b\})$ is still separable by $\neg B$. Indeed, $(\mathcal{D}, a) \not\preceq_{\mathcal{ALCI}} (\mathfrak{A}, b)$ as any simulation containing $(a, b^{\mathfrak{A}})$ would also contain (a, x) for some $R \circ R \circ R$-successor $x \in \mathsf{dom}(\mathfrak{A})$ of $b^{\mathfrak{A}}$. Then, as $b^{\mathfrak{A}} \in B^{\mathfrak{A}}$ we have $x \notin B^{\mathfrak{A}}$ by $\mathcal{O}$, contradicting $\text{BISIM}_{\mathcal{ALCI}}$ as $(\mathfrak{A}, x) \not\sim_{\mathcal{ALCI}} (\mathfrak{A}, b^{\mathfrak{A}})$.

Database $\mathcal{D}$, with bi-infinite R-chain (modulo bisimulation) enforced around b by $\mathcal{O}$.

2.21. *The impact of UNA in* $\mathcal{ALCQ}(\mathcal{I})$. Without the UNA, Lemma 2.18 fails if the separation language admits counting. Let $\mathcal{O} = \emptyset$, $\mathcal{D} = \{R(a, a_0), R(a, a_1), R(b, b_0)\}$ and make the UNA. Then there exists $\mathfrak{A} \models \mathcal{K}$ (the database seen as a model) such that $(\mathcal{D}_a, a) \not\preceq_{\mathcal{ALCQI}} (\mathfrak{A}, b^{\mathfrak{A}})$, but $(\mathcal{K}, \{a\}, \{b\})$ is not weakly $\mathcal{ALCQI}$-separable: for every $\mathfrak{A} \models \mathcal{K}$ there exists $\mathfrak{B} \models \mathcal{K}$ such that $(\mathfrak{B}, a^{\mathfrak{B}}) \sim_{\mathcal{ALCQI}} (\mathfrak{A}, b^{\mathfrak{A}})$. That is made possible by identifying $b_0^{\mathfrak{B}}$ and $b_1^{\mathfrak{B}}$.

We now prove Lemma 2.18 for each $\mathcal{L} \in \mathbf{DL}_{\mathcal{IQ}}$. The (1) $\Rightarrow$ (2) direction is straightforward in all cases: the restriction $S|_{\mathsf{cons}(\mathcal{D})}$ of any $\mathcal{ALCI}$-bisimulation (resp. $\mathcal{ALCQI}$) S between pointed models $(\mathfrak{B}, a^{\mathfrak{B}})$ and $(\mathfrak{A}, b^{\mathfrak{A}})$ defines an $\mathcal{ALCI}$-simulation (resp. $\mathcal{ALCQI}$) between $\mathcal{D}$ and $\mathfrak{A}$ containing $(a, b^{\mathfrak{A}})$. For $\mathcal{ALC}(\mathcal{Q})$, consider the restriction S_a of S to $\{c^{\mathfrak{B}} \mid c \in \mathsf{cons}(\mathcal{D}_a^{\uparrow})\}$ and it suffices to point out that $\mathcal{K}_{a,S_a}$ is satisfied by $\mathfrak{B}$. We then focus on (2) $\Rightarrow$ (1).

2.3.2. *Proof of Lemma 2.18 in* $\mathcal{ALCI}$

Assume $(\mathcal{D}, a) \preceq_{\mathcal{ALCI}} (\mathfrak{A}, b^{\mathfrak{A}})$ for some $a \in E^{+}$. We extend the model $\mathfrak{A}_{\mathcal{D}_a}$ induced by the maximal connected component of a in $\mathcal{D}$ (Rem. 1.10) into a model $\mathfrak{B}$ of $\mathcal{K}$ satisfying $(\mathfrak{B}, a^{\mathfrak{B}}) \sim_{\mathcal{ALCI}} (\mathfrak{A}, b^{\mathfrak{A}})$.

2.22. *Definition.* Let $S \subseteq \mathsf{cons}(\mathcal{D}) \times \mathsf{dom}(\mathfrak{A})$ be an $\mathcal{ALCI}$-simulation witnessing $(\mathcal{D}, a) \preceq_{\mathcal{ALCI}} (\mathfrak{A}, b^{\mathfrak{A}})$. For any $c \in \mathsf{cons}(\mathcal{D}_a)$, choose a unique $\delta \in \mathsf{dom}(\mathfrak{A})$ such that $(c, \delta) \in S$. Let d_c be arbitrary fresh domain elements for all $d \in \mathsf{dom}(\mathfrak{A})$ such that $d_c = c^{\mathfrak{A}}$ if $d = \delta$. We define a copy $\mathfrak{A}_c$ of $\mathfrak{A}$ for each $c \in \mathsf{cons}(\mathcal{D}_a)$, and a model $\mathfrak{A}'$ of $\mathcal{D}_a$, which informally consists of all the copies $\mathfrak{A}_c$ "plugged" onto

$\mathcal{D}_a$ at each c. The definitions are given below, for all concept and role names A, R.

$$\begin{aligned}
\mathsf{dom}(\mathfrak{A}_c) &= \{d_c \mid d \in \mathsf{dom}(\mathfrak{A})\} & \mathsf{dom}(\mathfrak{A}') &= \bigcup_{c \in \mathsf{cons}(\mathcal{D}_a)} \mathsf{dom}(\mathfrak{A}_c) \\
A^{\mathfrak{A}_c} &= \{d_c \mid d \in A^{\mathfrak{A}}\} & A^{\mathfrak{A}'} &= \bigcup_{c \in \mathsf{cons}(\mathcal{D}_a)} A^{\mathfrak{A}_c} \\
R^{\mathfrak{A}_c} &= \{(d_c, d'_c) \mid (d, d') \in R^{\mathfrak{A}}\} & R^{\mathfrak{A}'} &= \bigcup_{c \in \mathsf{cons}(\mathcal{D}_a)} R^{\mathfrak{A}_c} \\
& & & \quad \cup \{(d^{\mathfrak{A}}, e^{\mathfrak{A}}) \mid R(d, e) \in \mathcal{D}_a\}
\end{aligned}$$

Moreover, let $c^{\mathfrak{A}'} = c^{\mathfrak{A}}$ for all $c \in \mathsf{cons}(\mathcal{D}_a)$. Finally, let $\mathfrak{B}$ be the disjoint union of $\mathfrak{A}'$ and a model $\mathfrak{A}''$ of $(\mathcal{O}, \mathcal{D} \setminus \mathcal{D}_a)$, with $c^{\mathfrak{B}} = c^{\mathfrak{A}'}$ for all $c \in \mathsf{cons}(\mathcal{D}_a)$ and $c^{\mathfrak{B}} = c^{\mathfrak{A}''}$ for all $c \in \mathsf{cons}(\mathcal{D} \setminus \mathcal{D}_a)$.

As $\mathfrak{A}'' \models (\mathcal{O}, \mathcal{D} \setminus \mathcal{D}_a)$ and $\mathfrak{A}' \models \mathcal{D}_a$, we have $\mathfrak{B} \models \mathcal{K}$ iff $\mathfrak{A}' \models \mathcal{O}$. As $a^{\mathfrak{B}} \in \mathsf{dom}(\mathfrak{A}')$, we have $(\mathfrak{B}, a^{\mathfrak{B}}) \sim_{\mathcal{ALCI}} (\mathfrak{A}, b^{\mathfrak{A}})$ iff $(\mathfrak{A}', a^{\mathfrak{A}'}) \sim_{\mathcal{ALCI}} (\mathfrak{A}, b^{\mathfrak{A}})$. It is then sufficient to show there exists a bisimulation β between $\mathfrak{A}'$ and $\mathfrak{A}$ such that $(a^{\mathfrak{A}'}, b^{\mathfrak{A}}) \in \beta$ and such that for all $x \in \mathsf{dom}(\mathfrak{A}')$, there exists $y \in \mathsf{dom}(\mathfrak{A})$ with $(x, y) \in \beta$. That is done in the next Lemma. It implies $\mathfrak{A}' \models \mathcal{O}$ since $\mathfrak{A} \models \mathcal{O}$, hence $\mathfrak{B} \models \mathcal{K}$.

2.23. Lemma. *There exists an $\mathcal{ALCI}$-bisimulation β between $\mathfrak{A}'$ and $\mathfrak{A}$ such that $(a^{\mathfrak{A}'}, b^{\mathfrak{A}}) \in \beta$ and such that for all $x \in \mathsf{dom}(\mathfrak{A}')$, there exists $y \in \mathsf{dom}(\mathfrak{A})$ with $(x, y) \in \beta$.*

Proof. By definition of an $\mathcal{ALCI}$-simulation, if $(c, d) \in S$ and $(c, d') \in S$, then $(\mathfrak{A}, d) \sim_{\mathcal{ALCI}} (\mathfrak{A}, d')$. It is witnessed by a bisimulation in $\mathsf{cons}(\mathcal{D}) \times \mathsf{dom}(\mathfrak{A})$ that naturally induces a bisimulation in $\mathsf{dom}(\mathfrak{A}') \times \mathsf{dom}(\mathfrak{A})$, which we denote by $S_{c,d,d'}$. For any set of binary relations $\mathcal{R}$, let $\mathcal{R}_*$ be the set of finite compositions $R_1 \circ \cdots \circ R_n$ such that $\{R_1, \ldots, R_n\} \subseteq \mathcal{R}$. The empty composition is allowed and is defined to be the identity relation. Then, we define the following relations.

$$\begin{aligned}
\beta &= \beta''_* \circ \beta' \\
\beta' &= \big\{(d_c, d) \mid c \in \mathsf{cons}(\mathcal{D}_a), d \in \mathsf{dom}(\mathfrak{A})\big\} \\
\beta'' &= \bigcup \big\{S_{c,d,d'} \mid c \in \mathsf{cons}(\mathcal{D}_a), d, d' \in \mathsf{dom}(\mathfrak{A})\big\}
\end{aligned}$$

As each $S_{c,d,d'}$ is a bisimulation, β'' is also a bisimulation and so is β''_*. We now check that β is an $\mathcal{ALCI}$-bisimulation, *i.e.* satisfies ATOM, FORTH and BACK over every role name and their inverse. Let $(d_c, d') \in \beta$ for some $c \in \mathsf{cons}(\mathcal{D}_a)$ and $d \in \mathsf{dom}(\mathfrak{A})$. Then by definition of β, either $d = d'$ or $(d, d') \in \beta_1 \circ \cdots \circ \beta_n$ for some non-empty subset $\{\beta_1, \ldots \beta_n\} \subseteq \beta''$. We then suppose otherwise throughout the rest of the proof.

ATOM. Let A be a concept name. It follows from the definition of $\mathfrak{A}'$ that $d_c \in A^{\mathfrak{A}'}$ iff $d \in A^{\mathfrak{A}}$ and it follows from β''_* being a bisimulation that $d \in A^{\mathfrak{A}}$ iff $d' \in A^{\mathfrak{A}}$.

FORTH. Let $e_{c'}$ be an R-successor of d_c in $\mathfrak{A}'$ for some role name R.
(1) Suppose $d \neq c^{\mathfrak{A}'}$. Then all successors of d_c belong to $\mathfrak{A}_c$, so $c' = c$. Then, $(d,e) \in R^{\mathfrak{A}}$ by definition of $\mathfrak{A}_c$. Because it is a bisimulation, β''_* satisfies FORTH, so there exists $e' \in \mathsf{dom}(\mathfrak{A})$ such that $(d',e') \in R^{\mathfrak{A}}$ and $(e,e') \in \beta''_*$. Then, $(e,e') \in \beta$.
(2) Suppose $d_c = c^{\mathfrak{A}'}$. If $c' = c$, then the proof goes as before. If $c' \neq c$, then $e_{c'} \notin \mathsf{dom}(\mathfrak{A}_c)$, hence $e_{c'} = c'^{\mathfrak{A}'}$ by definition of $\mathfrak{A}'$. Furthermore, $(c,d) \in S$ by definition of $\mathfrak{A}_c$. Since S satisfies FORTH, there exists a successor f (which is not necessarily e) of d via R such that $(c',f) \in S$. Since β''_* satisfies FORTH, there exists $e' \in \mathsf{dom}(\mathfrak{A})$ such that $(d',e') \in R^{\mathfrak{A}}$ and $(f,e') \in \beta''_*$. But $(c',e) \in S$ by definition of $\mathfrak{A}_{c'}$, so $(e,f) \in S_{c',e,f} \subseteq \beta''$ since e and f have c' as a common predecessor via S. Then $(e,e') \in \beta''_*$ by composition and $(e_{c'},e) \in \beta'$ by definition of β', hence $(e_{c'},e') \in \beta$.

BACK. Let e' be an R-successor of d' in $\mathfrak{A}$ for some role name R. Since β''_* is a bisimulation, there exists $e \in \mathfrak{A}$ such that $(d,e) \in R^{\mathfrak{A}}$ and $(e,e') \in \beta''_*$. But then, $(e_c,d_c) \in R^{\mathfrak{A}_c}$ by definition of $\mathfrak{A}_c$. It thereby holds that $(e_c,e) \in \beta'$ and $(e,e') \in \beta''_*$, hence $(e_c,e') \in \beta$. ⊣

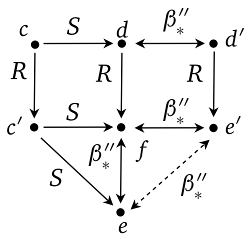


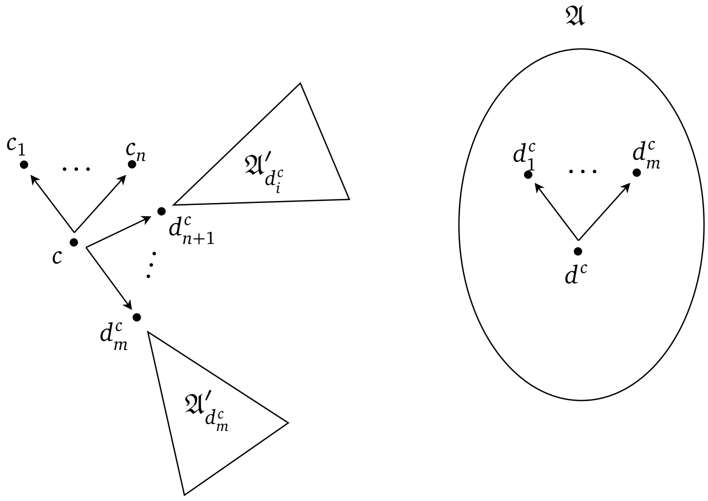
Illustration for FORTH (2).

2.3.3. *Proof of Lemma 2.18 in $\mathcal{ALCQI}$*

Assume that $(\mathcal{D},a) \preceq_{\mathcal{ALCQI}} (\mathfrak{A}, b^{\mathfrak{A}})$ for some $a \in E^+$. Let $S \subseteq \mathsf{cons}(\mathcal{D}) \times \mathsf{dom}(\mathfrak{A})$ be an $\mathcal{ALCQI}$-simulation between $\mathcal{D}$ and $\mathfrak{A}$ with $(a, b^{\mathfrak{A}}) \in S$. We construct a model $\mathfrak{B}$ of $\mathcal{K}$ such that $(\mathfrak{B}, a^{\mathfrak{B}}) \sim_{\mathcal{ALCQI}} (\mathfrak{A}, b^{\mathfrak{A}})$.

2.24. *Definition.* Take for every c in the maximally connected component $\mathcal{D}_a$ of a in $\mathcal{D}$ an element $d^c \in \mathfrak{A}$ such that $(c,d^c) \in S$. Let $c_1,\dots,c_n$ be the R-successors of c in $\mathcal{D}$ for some (possibly inverse) role R. Let $d^c_1,\dots,d^c_m$ be the R-successors of d^c in $\mathfrak{A}$. There exists a subset D of $\{d^c_1,\dots,d^c_m\}$ such that S contains a bijection $f : \{c_1,\dots,c_n\} \to D$. Assume without loss of generality that $f = \{(c_1,d_1),\dots,(c_n,d_n)\}$. Consider for $n < i \leqslant m$ the tree unfolding $\mathcal{T}_{\mathfrak{A}}^{d^c_i}$ of $\mathfrak{A}$ at

d_i^c (Def. 1.32). Remove from the domain of each $\mathscr{T}_{\mathfrak{A}}^{d_i^c}$ the subtree of root $d_i^c R^- d^c$ and denote the resulting model by $\mathfrak{A}'_{d_i^c}$. Expand $\mathcal{D}_a$ by adding c to $A^{\mathfrak{B}}$ for all concept names A with $d^c \in A^{\mathfrak{A}}$ and connecting for every $n < i \leqslant m$ a fresh copy of $\mathfrak{A}'_{d_i^c}$ to $\mathcal{D}_a$ by adding (c, d_i^c) to $R^{\mathfrak{B}}$. Let $\mathfrak{B}_0$ be the resulting model obtained by doing this for all $c \in \mathsf{cons}(\mathcal{D}_a)$. Let $\mathfrak{B} = \mathfrak{B}_0 \uplus \mathfrak{B}'$, where $\mathfrak{B}' \models (\mathcal{O}, \mathcal{D} \setminus \mathcal{D}_a)$.



"Plugging" the additional trees onto c, for one particular $c \in \mathsf{cons}(\mathcal{D}_a)$.

Then $\mathfrak{B}$ is as required: $(\mathfrak{B}, a^{\mathfrak{B}}) \sim_{\mathcal{ALCQI}} (\mathfrak{A}, b^{\mathfrak{A}})$ is proved in Lemma 2.25, and $\mathfrak{B} \models \mathcal{D}$ is immediate by definition. Also, $\mathfrak{B}' \models \mathcal{O}$ by definition. Then, $\mathfrak{B}_0 \models \mathcal{O}$ follows from $\mathfrak{A} \models \mathcal{O}$ and the fact that the bisimulation exhibited in Lemma 2.25 is defined on every element of $\mathsf{dom}(\mathfrak{B}_0)$.

2.25. Lemma. *There exists an $\mathcal{ALCQI}$-bisimulation β between $\mathfrak{B}_0$ and $\mathfrak{A}$ such that $(a^{\mathfrak{B}_0}, b^{\mathfrak{A}}) \in \beta$ and such that for all $x \in \mathsf{dom}(\mathfrak{B}_0)$, there exists $y \in \mathsf{dom}(\mathfrak{A})$ with $(x, y) \in \beta$.*

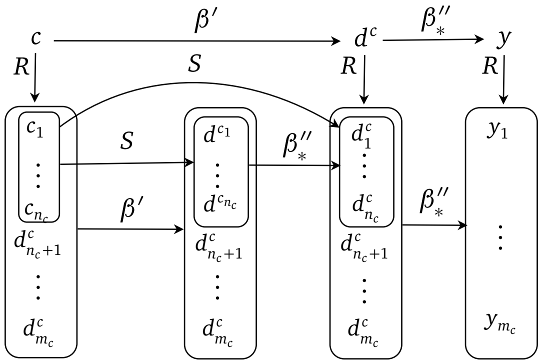
Proof. By definition of an $\mathcal{ALCQI}$-simulation, if $(c, d) \in S$ and $(c, d') \in S$, then $(\mathfrak{A}, d) \sim_{\mathcal{ALCQI}} (\mathfrak{A}, d')$. Let $S_{c,d,d'}$ be a bisimulation witnessing that. We define the following relations.

$$\beta = \beta''_* \circ \beta'$$
$$\beta' = \bigcup_{c \in \mathsf{cons}(\mathcal{D}_a)} \{(c, d^c)\} \cup \bigcup_{n < i \leqslant m} \{(p, \mathsf{tail}(p)) : p \in \mathsf{dom}(\mathfrak{A}'_{d_i^c})\}$$
$$\beta'' = \bigcup_{c \in \mathsf{cons}(\mathcal{D}_a)} \{S_{c,d,d'} : d, d' \in \mathsf{dom}(\mathfrak{A}), (c, d) \in S, (c, d') \in S\}$$

We show β is an $\mathcal{ALCQI}$-bisimulation. ATOM is immediate by S being an $\mathcal{ALCQI}$-simulation and by construction of $\mathfrak{B}_0$.

QFORTH. Suppose $(x, y) \in \beta$ and R is a (possibly inverse) role.

1. Suppose $x = c$ for some $c \in \mathsf{cons}(\mathcal{D}_a)$. We can write any finite set S_x of R-successors of x as $\{c_i : i \in I\} \cup \{d^c_j : j \in J\}$ with $I \subseteq \{1,\dots,n\}$ and $J \subseteq \{n+1,\dots,m\}$. By definition of β, $x = c$ implies $(d^c, y) \in \beta''_*$. We want to show β contains a bijection between S_x and some set of R-successors of y. By definition, β' contains a bijection onto the set $\{d^{c_i} : i \in I \cup J\}$, defined by $c_i \mapsto d^{c_i}$ for all $i \in I$ and $d^c_j \mapsto d^c_j$ (writing d^c_j on the right as the tail of the path comprised only of d^c_j) for all $j \in J$. Then, by definition of S we have $(c_i, d^{c_i}) \in S$ and $(c_i, d^c_i) \in S$. Since S is a simulation thus satisfies $\textsc{Bisim}_{\mathcal{ALCQI}}$, and by definition of β''_*, we get $(d^{c_i}, d^c_i) \in \beta''_*$ for all $i \in I$. Then β''_* contains the bijection defined by $d^{c_i} \mapsto d^c_i$ for all $i \in I$ and the identity map on all d^c_j, $j \in J$. Now, d^c_i is an R-successor of d^c for all $i \in I \cup J$ and since $(d^c, y) \in \beta''_*$ and β''_* is an $\mathcal{ALCQI}$-bisimulation, there exists a bijection $d^c_i \mapsto y_i$, $i \in I \cup J$ onto a set $\{y_i : i \in I \cup J\}$ of R-successors of y. Then, by composition of bijections contained in β''_* and definition of β''_*, the bijection $d^{c_i} \mapsto y_i$, $i \in I$ is also contained in β''_*. Then, the bijection defined by $c_i \mapsto y_i$ for all $i \in I$ and $d^c_j \mapsto y_j$ for all $j \in J$ is contained in $\beta''_* \circ \beta'$, *i.e.* in β.



An illustration of the argument for QFORTH.

2. Suppose now that $x \neq c$ for any $c \in \mathsf{cons}(\mathcal{D}_a)$. Then, $x \in \mathsf{dom}(\mathfrak{A}'_{d^c})$ for some $c \in \mathsf{cons}(\mathcal{D}_a)$. $(x, y) \in \beta$ implies $(x, \mathsf{tail}(x)) \in \beta'$ and $(\mathsf{tail}(x), y) \in \beta''_*$. Let $x_1, \dots, x_n$ be a finite set of R-successors of x in $\mathfrak{B}_0$. Suppose first that $x_i \neq c$ for all $i \leqslant n$ and $c \in \mathsf{cons}(\mathcal{D}_a)$. Then, β' contains a bijection $x_i \mapsto \mathsf{tail}(x_i)$ onto the set $\{\mathsf{tail}(x_i) : 1 \leqslant i \leqslant n\}$ of R-successors of $\mathsf{tail}(x)$. Then, because $(\mathsf{tail}(x), y) \in \beta''_*$, β''_* contains a bijection $\mathsf{tail}(x_i) \mapsto y_i$ for some set of R-successors $y_1, \dots, y_n$ of y. The composition of those two bijections is contained in $\beta''_* \circ \beta'$ and clearly the desired witness. Now, if $c \in \{x_1, \dots, x_n\}$, then $x = d^c_i$ (as a path) for some $i \in \{n+1, \dots, m\}$, so $\mathsf{tail}(x) = d^c_i$ and d^c is an R-successor of x. So the bijection, contained in β', defined on $\{x_1, \dots, x_n\} \setminus \{c\}$ by $x_i \mapsto \mathsf{tail}(x_i)$ and on $\{c\}$ by $c \mapsto d^c$ is a bijection onto a set of R-successors of $\mathsf{tail}(x)$. Then, the final witness bijection is simply given by composing the former with the bijection witnessing $(\mathsf{tail}(x), y) \in \beta''_*$, from $\{x_1, \dots, d^c, \dots, x_n\}$ onto a set of R-successors of y.

QBack. Suppose $(x,y) \in \beta$ and R is a (possibly inverse) role.

1. Suppose $x = c \in \mathsf{cons}(\mathcal{D}_a)$. Then, $(d^c, y) \in \beta''_*$. Let $y_1, \dots, y_n$ be a finite set of R-successors of y. By β''_* being an $\mathcal{ALCQI}$-bisimulation, there exists a bijection between $\{y_1, \dots, y_n\}$ and some set $\{d^c_1, \dots, d^c_n\}$ of R-successors of d^c. To define the desired bijection between $\{d^c_1, \dots, d^c_n\}$ and some set of R-successors of x, fix a given d^c_i. By construction of $\mathfrak{B}_0$, either $(c', d^c_i) \in S$ for some R-successor c' of c in $\mathcal{D}_a$, or $(x, d^c_i) \in R^{\mathfrak{B}_0}$. Let us build the desired bijection. In the first case, $(c', d^{c'}) \in S \cap \beta'$ so $(d^c_i, d^{c'}) \in \beta''_*$, so the bijection sends d^c_i onto c'. In the second case, the bijection sends d^c onto itself (seen as a path of one element). That last bijection is clearly contained in β', making the composition contained in β.

2. Suppose $x \neq c$ for any $c \in \mathsf{cons}(\mathcal{D}_a)$. Then, $x \in \mathsf{dom}(\mathfrak{A}'_{d^c})$ and $(x, \mathsf{tail}(x)) \in \beta'$ and $(\mathsf{tail}(x), y) \in \beta''_*$. Let $y_1, \dots, y_n$ be a finite set of R-successors of y. By β''_* being an $\mathcal{ALCQI}$-bisimulation, there exists a bijection between $\{y_1, \dots, y_n\}$ and some set of R-successors of $\mathsf{tail}(x)$. By definition of $\mathfrak{B}_0$, that set is either comprised of elements of the form $\mathsf{tail}(x')$ with x' a path in the same tree $\mathfrak{A}'_{d^c}$ as x, or of d^c. The remaining bijection contained in β' is then simply defined by sending $\mathsf{tail}(x')$ to x and d^c to c if needed. ⊣

2.3.4. *Proof of Lemma 2.18 in $\mathcal{ALC}(\mathcal{Q})$*

Proof for $\mathcal{ALC}$. (2) ⇒ (1). Let S be a witnessing simulation and $\mathfrak{B}_0 \models \mathcal{K}_{\mathfrak{A},S}$. Let $\mathfrak{A}_c$ be defined for each $c \in \mathsf{cons}(\mathcal{D}^{\uparrow}_a)$ as in the $\mathcal{ALCI}$ proof. Consider the model $\mathfrak{A}'$ defined as in the $\mathcal{ALCI}$ proof, but only using all $c \in \mathsf{cons}(\mathcal{D}^{\uparrow}_a)$. Let $\mathfrak{B}_1 = \mathfrak{A}' \cup \mathfrak{B}_0$ with $c^{\mathfrak{B}_1} = c^{\mathfrak{A}_c}$ if $c \in \mathsf{cons}(\mathcal{D}^{\uparrow}_a)$ and $c^{\mathfrak{B}_1} = c^{\mathfrak{B}_0}$ otherwise. Let $\mathfrak{B}$ be equal to $\mathfrak{B}_1$ in all aspects except that, for all $R \in \mathsf{rel}_2$, $R^{\mathfrak{B}} := R^{\mathfrak{B}_1} \cup \{(c^{\mathfrak{B}}, d^{\mathfrak{B}}) \mid R(c,d) \in \mathcal{D}\}$. Then $(\mathfrak{B}, a^{\mathfrak{B}}) \sim_{\mathcal{ALC}} (\mathfrak{A}, b^{\mathfrak{A}})$ is clear from $(\mathfrak{A}', a^{\mathfrak{B}}) \sim_{\mathcal{ALC}} (\mathfrak{A}, b^{\mathfrak{A}})$ and witnessed by the same bisimulation as in the $\mathcal{ALCI}$ case. It remains to prove $\mathfrak{B} \models \mathcal{K}$. That $\mathfrak{B} \models \mathcal{D}$ is clear. We then show that $\mathfrak{B} \models \mathcal{O}$. By assumption, $\mathfrak{B}_0 \models \mathcal{O}$. By the arguments from the $\mathcal{ALCI}$ proof, $\mathfrak{A}' \models \mathcal{O}$, so $\mathfrak{B}_1 \models \mathcal{O}$. Then, $\mathfrak{B} \not\models \mathcal{O}$ iff relations added on top of $\mathfrak{B}_1$ contradict $\mathcal{O}$. It can be quickly checked that any additional pair $(c^{\mathfrak{B}}, d^{\mathfrak{B}})$ in some $R^{\mathfrak{B}}$ is such that $c \notin \mathsf{cons}(\mathcal{D}^{\uparrow}_a)$ and $d \in \mathsf{cons}(\mathcal{D}^{\uparrow}_a)$. But since $\mathfrak{B}_0$ satisfies $\mathcal{K}_{\mathfrak{A},S}$, $c^{\mathfrak{B}_1}$ already has an R-successor satisfying the same $\mathcal{O}$-type as $d^{\mathfrak{A}'}$. Then, adding the pair $(c^{\mathfrak{B}}, d^{\mathfrak{B}})$ to the interpretation of R has no effect, *i.e.* $C^{\mathfrak{B}} = C^{\mathfrak{B}_1}$ for all $C \in \mathsf{sub}(\mathcal{O})$. ⊣

Proof for $\mathcal{ALCQ}$. (2) ⇒ (1). Consider the model $\mathfrak{B}_0$ defined as in the $\mathcal{ALCQI}$ proof but only with all $c \in \mathsf{cons}(\mathcal{D}^{\uparrow}_a)$. Let $\mathfrak{B}'$ be a model of $\mathcal{K}_{\mathfrak{A},S}$. We can assume without loss of generality that $\mathfrak{B}'$ an $\mathcal{ALCQ}$-forest (Lem. 1.40). Define $\mathfrak{B}''$ by removing, for any $c \in \mathsf{cons}(\mathcal{D}^{\uparrow}_a)$ that is a successor of some $d \notin \mathsf{cons}(\mathcal{D}^{\uparrow}_a)$, the subtree of root $c^{\mathfrak{B}'}$ from $\mathfrak{B}'$. Then, define $\mathfrak{B}$ as in the $\mathcal{ALC}$ case. Then $\mathfrak{B} \models \mathcal{K}$ (same argument as for $\mathcal{ALC}$) and $(\mathfrak{B}, a^{\mathfrak{B}}) \sim_{\mathcal{ALCQ}} (\mathfrak{A}, b^{\mathfrak{A}})$ (same argument as for $\mathcal{ALCQI}$). ⊣

2.3.5. *Final characterisation*

We now move on to the final characterisation of full weak $(\mathcal{L}, \mathcal{L})$-separability, based on homomorphisms from the database. It follows very easily from the simulation characterisation, but only with the crucial help of projectivity. Again, each language needs a slightly different variation of "homomorphism" here, mirroring their simulation counterparts; only with $\mathcal{ALCI}$ do we use "homomorphism" in the natural understanding of the term, aside from the preservation of constants.

We start by defining each language's homomorphism variant.

2.26. *Definition.* (1) By $(\mathcal{D}, a) \to (\mathfrak{A}, b^{\mathfrak{A}})$, we denote that there exists a homomorphism $h : (\mathfrak{A}_{\mathcal{D}}, a) \to (\mathfrak{A}, b^{\mathfrak{A}})$ where $\mathfrak{A}_{\mathcal{D}}$ is the canonical model induced by $\mathcal{D}$. (2) By $(\mathcal{D}, a) \to_i (\mathfrak{A}, b^{\mathfrak{A}})$ we additionally require that h is locally injective for all role names and their inverses, *i.e.* for all $c \in \mathsf{cons}(\mathcal{D})$ and all $R \in \mathsf{rel}_2^-$, the restriction of h to the set of R-successors of c in $\mathfrak{A}_{\mathcal{D}}$ is injective. (3) By $(\mathcal{D}, a) \to_r (\mathfrak{A}, b^{\mathfrak{A}})$ we denote that there is a homomorphism $h : (\mathfrak{A}_{\mathcal{D}_a^\uparrow}, a) \to (\mathfrak{A}, b^{\mathfrak{A}})$ such that the extended knowledge base $\mathcal{K}_{\mathfrak{A},h} := (\mathcal{O}, \mathcal{D}^+)$ is satisfiable, where $\mathcal{D}^+ = \mathcal{D} \cup \{C(c) \mid C \in \mathsf{sub}(\mathcal{O}), c \in \mathsf{cons}(\mathcal{D}_a^\uparrow), h(c) \in C^{\mathfrak{A}}\}$. (4) By $(\mathcal{D}, a) \to_{r,i} (\mathfrak{A}, b^{\mathfrak{A}})$, we require, in addition to (3), that h is locally injective for all role names (not necessarily inverse roles).

We can then connect the "simulation" characterisation and the "homomorphism" characterisation in the next Theorem, using the following Lemma. We rename homomorphism variants as such.

$$\begin{aligned} \to_{\mathcal{ALCI}} &:= \to \\ \to_{\mathcal{ALCQI}} &:= \to_i \\ \to_{\mathcal{ALC}} &:= \to_r \\ \to_{\mathcal{ALCQ}} &:= \to_{r,i} \end{aligned}$$

2.27. Lemma. *Let $\mathcal{L} \in \mathbf{DL}_{\mathcal{IQ}}$ and let $(\mathcal{K}, E^+, \{b\})$ be a labeled $\mathcal{L}$-knowledge base. The following conditions are equivalent.*

1. *There exists $\mathfrak{A} \models \mathcal{K}$ such that for all $a \in E^+$, $(\mathcal{D}, a) \not\precsim_{\mathcal{L}} (\mathfrak{A}, b^{\mathfrak{A}})$.*
2. *There exists $\mathfrak{A} \models \mathcal{K}$ such that for all $a \in E^+$, $(\mathcal{D}, a) \not\to_{\mathcal{L}} (\mathfrak{A}, b)$.*

Proof. (1) ⇒ (2) is trivial. For (2) ⇒ (1), assume $\mathfrak{A}$ is a model of $\mathcal{K}$ such that $(\mathcal{D}, a) \not\rightarrow_{\mathcal{L}} (\mathfrak{A}, b^{\mathfrak{A}})$ for all $a \in E^+$. Define an extension $\mathfrak{A}'$ of $\mathfrak{A}$ by taking for every $x \in \mathsf{dom}(\mathfrak{A})$ a fresh concept name C_x and setting $C_x^{\mathfrak{A}'} = \{x\}$ (crucial use of projectivity). Then $\mathfrak{A}'$ is as required for (1) since any two $\mathcal{L}$-bisimilar nodes in $\mathfrak{A}'$ are identical. ⊣

2.28. Theorem. *Let $(\mathcal{K}, E^+, \{b\})$ be an labeled $\mathcal{L}$-knowledge base, where $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ and $\mathcal{L} \in \mathbf{DL}_{\mathcal{IQ}}$. Then $(\mathcal{K}, E^+, \{b\})$ is weakly projectively $\mathcal{L}$-separable iff there exists $\mathfrak{A} \models \mathcal{K}$ such that, for all $a \in E^+$, $(\mathcal{D}, a) \not\rightarrow_{\mathcal{L}} (\mathfrak{A}, b^{\mathfrak{A}})$.*

Proof. Assume first that $(\mathcal{K}, E^+, \{b\})$ is weakly projectively $\mathcal{L}$-separable. By Theorem 2.16, there exists a model $\mathfrak{A}$ of $\mathcal{K}$ of finite outdegree such that for all $a \in E^+$ and all models $\mathfrak{B}$ of $\mathcal{K}$, $(\mathfrak{B}, a^{\mathfrak{B}}) \not\sim_{\mathcal{L}} (\mathfrak{A}, b^{\mathfrak{A}})$. By Lemma 2.18, $(\mathcal{D}, a) \not\lesssim_{\mathcal{L}} (\mathfrak{A}, b^{\mathfrak{A}})$ for all $a \in E^+$. By Lemma 2.27, there exists a model $\mathfrak{A}'$ of $\mathcal{K}$ such that $(\mathcal{D}, a) \not\rightarrow_{\mathcal{L}} (\mathfrak{A}', b^{\mathfrak{A}})$ for all $a \in E^+$, as required. Conversely, assume there exists a model $\mathfrak{A}$ of $\mathcal{K}$ such that $(\mathcal{D}, a) \not\rightarrow_{\mathcal{L}} (\mathfrak{A}, b^{\mathfrak{A}})$ for all $a \in E^+$. Clearly we may assume that $\mathfrak{A}$ has finite outdegree. By Lemma 2.27, there exists a model $\mathfrak{A}'$ of $\mathcal{K}$ of finite outdegree such that $(\mathcal{D}, a) \not\lesssim_{\mathcal{L}} (\mathfrak{A}', b^{\mathfrak{A}'})$ for all $a \in E^+$. By Lemma 2.18, for any model $\mathfrak{B}$ of $\mathcal{K}$ and all $a \in E^+$, $(\mathfrak{B}, a^{\mathfrak{B}}) \not\sim_{\mathcal{L}} (\mathfrak{A}', b^{\mathfrak{A}'})$. By Theorem 2.16, $(\mathcal{K}, E^+, \{b\})$ is $\mathcal{L}$-separable. ⊣

Since $\mathcal{ALCI}$ uses the usual definition of homomorphism, the above characterisation matches the one from Theorem 2.1 on **FO**-separability. Thus, if helper symbols are allowed, $\mathcal{ALCI}$ provides as much separating power as the much more expressive **FO** (as well as all **FO**-fragments containing **UCQ**), when the knowledge base is also in $\mathcal{ALCI}$.

2.29. Corollary. $\mathsf{sep}_w^p(\mathcal{ALCI}, \mathcal{ALCI}) = \mathsf{sep}_w^p(\mathcal{ALCI}, \mathcal{L})$ *for any* **FO***-fragment* $\mathcal{L}$ *containing* **UCQ**.

2.3.6. *Separability & query evaluation*

As in the **FO** case, being able to express separability in terms of homomorphisms from the database also allows us to express it in terms of query satisfaction, since any database can be seen as a conjunctive query and vice-versa. As each language has its own 'version' of homomorphism, it also has its corresponding version of query evaluation. Once again, only with $\mathcal{ALCI}$ are we dealing with the 'normal' version.

2.30. *Definition.* Let $\mathcal{K}$ be an $\mathcal{ALCQI}$-knowledge base and q a unary UCQ. We write

1. $\mathcal{K} \models^i q(a)$ if for all $\mathfrak{A} \models \mathcal{K}$ if there is a CQ p in q and a locally injective homomorphism from $p(x)$ to $\mathfrak{A}$ with $h(x) = a$.

2. $\mathcal{K} \models^r q(a)$ if for all $\mathfrak{A} \models \mathcal{K}$ there is a CQ p in q and a homomorphism from $p_x^{\uparrow}$ to $\mathfrak{A}$ with $h(x) = a$, with $p_x^{\uparrow}$ the restriction of p to the variables directedly reachable from x, such that the knowledge base $\mathcal{K}_{\mathfrak{A},h} = (\mathcal{O}, \mathcal{D}_p \cup \{C(y) \mid C \in \mathsf{sub}(\mathcal{O}), y \in \mathsf{var}(p_x^{\uparrow}), h(y) \in C^{\mathfrak{A}}\})$ is satisfiable, where $\mathcal{D}_p$ is the canonical database induced by p.

3. $\mathcal{K} \models^{r,i} q(a)$ if for all $\mathfrak{A} \models \mathcal{K}$ there is a CQ p in q and a locally injective (not necessarily w.r.t. inverse roles) homomorphism from $p_x^{\uparrow}$ to $\mathfrak{A}$ with $h(x) = a$, with $p_x^{\uparrow}$ the restriction of p to the variables reachable along role names (*i.e.* directedly) from x, such that $\mathcal{K}_{\mathfrak{A},h}$ is satisfiable.

Those variants naturally give rise to the decision problems we respectively call *locally injective, reachable*, and *locally injective reachable (U)CQ evaluation.* In the context of query evaluation, we then use the following notation for uniformity.

$$\begin{aligned} \models^{\mathcal{ALCI}} &\text{ for } \models \\ \models^{\mathcal{ALC}} &\text{ for } \models^r \\ \models^{\mathcal{ALCQI}} &\text{ for } \models^i \\ \models^{\mathcal{ALCQ}} &\text{ for } \models^{r,i} \end{aligned}$$

The following is then immediate from Lemma 2.27 and Definition 2.30.

2.31. Corollary. *Let $\mathcal{L} \in \mathbf{DL}_{\mathcal{IQ}}$. Let $(\mathcal{K}, E^+, \{b\})$ be a labeled $\mathcal{ALCQI}$-knowledge base, where $\mathcal{K} = (\mathcal{O}, \mathcal{D})$. Then $(\mathcal{K}, E^+, \{b\})$ is weakly projectively $\mathcal{L}$-separable iff $\mathcal{K} \not\models^{\mathcal{L}} \bigvee_{a \in E^+} q(\mathcal{D}_a, a)(b)$.*

2.3.7. *Preliminaries for complexity*

Corollary 2.31 directly implies that for each $\mathcal{L} \in \mathbf{DL}_{\mathcal{IQ}}$, the complexity of the complement of ($\mathcal{L}$'s variant of) rooted unary UCQ evaluation on $\mathcal{L}$-knowledge bases is an upper bound on the complexity of projective full weak $(\mathcal{L}, \mathcal{L})$-separability with one negative example. It was determined (in combined complexity) to be NExp-complete for $\mathcal{ALCI}$ in [Lu07, Lu08]. We determine the complexity for the other three variants. Surprisingly, the $\mathcal{ALCQI}$ variant turns out to be Exp-complete, that is no harder than satisfiability of $\mathcal{ALCQI}$-knowledge bases.

$\mathcal{ALCI}$	unary rooted UCQ eval.	[Lu08] NExp
$\mathcal{ALC}$	reach. unary rooted UCQ eval.	NExp
$\mathcal{ALCQI}$	loc. inj. unary rooted UCQ eval.	Exp
$\mathcal{ALCQ}$	reach. loc. inj. unary rooted UCQ eval.	NExp

The lower bounds are also provided by reduction from each variant of UCQ evaluation, using the same argument as in Corollary 2.4. Putting the two together we get the following.

2.32. Proposition. *There is a mutual polynomial-time reduction between full weak projective* $(\mathcal{L}, \mathcal{L})$*-separability with one negative example and the complement of the* $\mathcal{L}$*-variant of unary rooted UCQ evaluation.*

2.3.8. *Complexity for* $\mathcal{ALCI}$

2.33. Corollary. *Projective full weak* $(\mathcal{ALCI}, \mathcal{ALCI})$*-separability is* NExp*-complete in combined complexity.*

Proof. The result follows from mutual reduction (Cor. 2.31) of $(\mathcal{ALCI}, \mathbf{FO})$-separability (which we now know coincides with $(\mathcal{ALCI}, \mathcal{ALCI})$-separability) to the complement of unary rooted UCQ evaluation on $\mathcal{ALCI}$-knowledge bases. The latter is proved to be coNExp-complete in [Lu07, Lu08]. ⊣

Data complexity bounds are due to [JLPW20, Thm 19].

2.34. Theorem. *Projective full weak* $(\mathcal{ALCI}, \mathcal{ALCI})$*-separability is* NExp*-complete in data complexity.*

Proof sketch. The upper bound is immediate since data complexity is bounded by combined complexity. For the lower bound, it suffices to show that unary rooted UCQ evaluation on a fixed ontology and a database whose Gaifman graph is connected is coNExp-hard. That is done by reduction from a tiling problem, which happens to also be introduced in this thesis for another problem (Prop. 2.51 in the $\mathcal{ALC}$ subsection). ⊣

2.3.9. *Complexity for* $\mathcal{ALCQI}$

2.35. Theorem. *Projective full weak* $(\mathcal{ALCQI}, \mathcal{ALCQI})$*-separability is* Exp*-complete in combined complexity.*

Proof. The lower bound is obtained by a reduction to satisfiability of $\mathcal{ALCQI}$-knowledge bases (Rem. 1.63), which is EXP-complete. The upper bound is obtained as the $\mathcal{ALCI}$ one, via reduction from the corresponding – locally injective – rooted unary UCQ evaluation problem, defined in Definition 2.30. The difficult part is now to show that injective rooted unary UCQ evaluation is EXP-complete (Lem. 2.40). ⊣

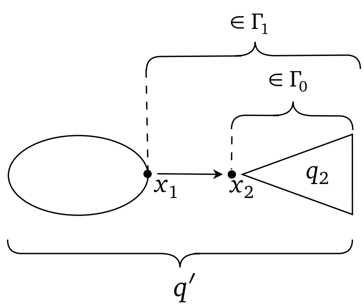
To obtain Lemma 2.40, we first characterize locally injective rooted unary UCQ evaluation in a way that can be efficiently checked. For that, we introduce the notion of *extended type assignment* (Definitions 2.36 and 2.38). Then, we show that locally injective rooted unary UCQ evaluation can be seen as the satisfaction by some extended type assignment of a semantic condition (Lem. 2.39). Finally, we show that condition can be checked in exponential time (Lem. 2.40).

2.36. *Definition.* Let an $\mathcal{ALCQI}$-knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{D})$, a unary rooted UCQ $q_0(x_0)$, and a candidate answer $a_0 \in \mathsf{cons}(\mathcal{D})$ be given. Let Γ denote the set of pairs $(R(x_1, x_2), q_2(x_2))$ such that there exists a CQ q' in q_0 with $R(x_1, x_2)$ an atom in q' and q_2 a set of atoms in q' such that $q' \setminus \{R(x_1, x_2)\}$ consists of two (potentially empty) disconnected components q_1 and q_2 of atoms where

1. x_1 does not occur in q_2 and x_2 do not occur in q_1
2. $x_0 \neq x_2$ and x_0 does not occur in q_2, and
3. $q_{r,q_2}(x_1) = R(x_1, x_2) \cup q_2$ is a tree-shaped unary CQ with root x_1.

Define the following sets of unary tree-shaped CQs.

$$\Gamma_0 = \{q_2(x_2) \mid (R(x_1, x_2), q_2) \in \Gamma\}$$
$$\Gamma_1 = \{q_{r,q_2}(x_1) \mid (R(x_1, x_2), q_2) \in \Gamma\}$$



2.37. $|\Gamma|$ is polynomial in the size of q_0, as it is bounded by the number of atoms $R(x_1, x_2)$ contained in any CQ of q_0.

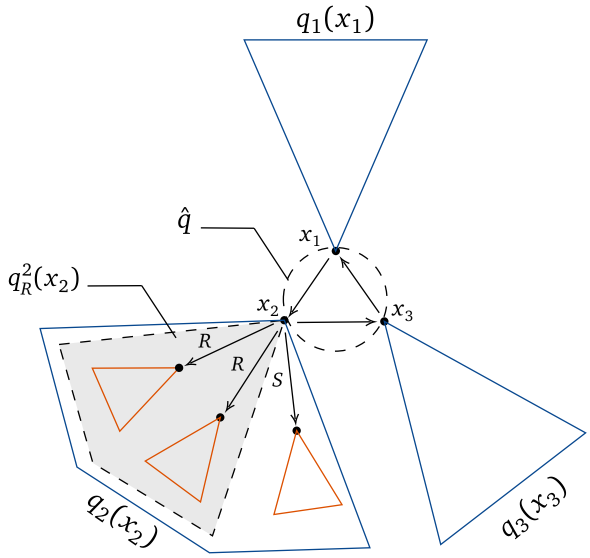
Let $\mathcal{K} = (\mathcal{O}, \mathcal{D})$, q_0 and a_0 be fixed for the remainder of this subsection.

2.38. *Definition.* Let $\mathfrak{A}$ be an $\mathcal{ALCQI}$-forest model of $\mathcal{K}$ and d an R-successor of $a^{\mathfrak{A}}$ for some $a \in \mathsf{cons}(\mathcal{D})$, such that $d \neq c^{\mathfrak{A}}$ for all $c \in \mathsf{cons}(\mathcal{D})$. A q_0-*extended* $\mathcal{O}$-*type* t is the union of an $\mathcal{O}$-type and a set that contains, for every $q(x) \in \Gamma_0$, either the expression $\mathsf{isat}(q(x))$ or $\neg\mathsf{isat}(q(x))$. We write $\top_{\mathcal{O}}^{q_0}$ for the set of q_0-extended $\mathcal{O}$-types. We say that $\mathsf{isat}(q(x))$ *is satisfied in* $(\mathfrak{A}, d)$ if $\mathfrak{A} \models^i q(d)$ and this is witnessed by a locally injective homomorphism h into $\mathsf{dom}(\mathfrak{A}) \setminus \{c^{\mathfrak{A}} : c \in \mathsf{cons}(\mathcal{D})\}$. As $\mathfrak{A}$ is a forest model of $\mathcal{D}$, h is locally injective iff it is injective (easy to check). A q_0-extended $\mathcal{O}$-type t *is satisfied in* $(\mathfrak{A}, d)$ if its concepts are satisfied in d and $\mathsf{isat}(q(x))$ is satisfied in $(\mathfrak{A}, d)$ iff $\mathsf{isat}(q(x)) \in t$, for all $q(x) \in \Gamma_0$.

A q_0-*extended* $\mathcal{O}$-*type assignment for* $\mathcal{D}$ is a function μ that assigns to every $a \in \mathsf{cons}(\mathcal{D})$ an $\mathcal{O}$-type $\mu(a)$ and to every triple (a, R, t) with $a \in \mathsf{cons}(\mathcal{D})$, R a role from $\mathcal{K}$, and t a q_0-extended $\mathcal{O}$-type, a natural number $\mu(a, R, t)$. We say μ is *small* if for every a and R we have $\sum_t \mu(a, R, t) \leqslant ||\mathcal{O}||$. We say μ is *realized by an* $\mathcal{ALCQI}$-*forest model* $\mathfrak{A}$ of $\mathcal{D}$ if for every $a \in \mathsf{cons}(\mathcal{D})$, a satisfies $\mu(a)$ in $\mathfrak{A}$ and the number of R-successors of a in $\mathfrak{A}$ outside $\mathsf{cons}(\mathcal{D})$ satisfying an q_0-extended $\mathcal{O}$-type t is $\mu(a, R, t)$. A q_0-extended $\mathcal{O}$-type assignment μ is $\mathcal{K}$-*realizable* if there exists an $\mathcal{ALCQI}$-forest model $\mathfrak{A}$ of $\mathcal{K}$ that realizes it.

A *forest decomposition of a CQ* q in q_0 is a partition $\hat{q} \cup q_1(x_1) \cup \cdots \cup q_n(x_n)$ of (the set of atoms in) q such that $q_i(x_i) \in \Gamma_1$. We assume that the variables $x_1, \ldots, x_n$ all occur in $\hat{q}$, which can be achieved by adding 'dummy atoms' of the form $\top(x_i)$. It can be verified that $\hat{q}$ and q_i share only the variable x_i, that x_0 occurs in $\hat{q}$, and that if x_0 occurs in q_i, then $x_i = x_0$. A *forest decomposition of the UCQ* q_0 is any forest decomposition of any of its CQs.

Given a forest decomposition $\hat{q} \cup q_1 \cup \cdots \cup q_n$ and x_i in $\hat{q}$ and a role R we obtain the tree-shaped CQ $q_R^j(x_j)$ as the conjunction of all queries of the form $R(x_j, y) \wedge q(y)$ in $\{q_1, \ldots, q_n\}$. For a q_0-extended $\mathcal{O}$-type assignment μ we write $\mathcal{K} \models_\mu^i q_R^j(a)$ if $\mathfrak{A} \models^i q_R^j(a^{\mathfrak{A}})$ for some model, or, equivalently, all models $\mathfrak{A}$ of $\mathcal{K}$ realizing μ – as μ fully determines at each R-successor of database elements whether $\mathsf{isat}(q)$ is satisfied for every $q \in \Gamma_0$ (with multiplicity) therefore fully determines whether q_r^j is locally injectively satisfied in a – and this is witnessed by an injective homomorphism from $q_R^j(x_j)$ into the interpretation induced by the subtree generated by $a^{\mathfrak{A}}$ in $\mathfrak{A}$ (so a is the only constant in $\mathcal{D}$ such that its interpretation in $\mathfrak{A}$ is in the range of h).



Example of a forest decomposition $\hat{q} \cup q_1 \cup q_2 \cup q_3$, with a detailed look at q_2. Elements of Γ_0 are displayed in orange and elements of Γ_1 in blue.

We say that a q_0-extended $\mathcal{O}$-type assignment μ for $\mathcal{D}$ *avoids* q_0 if for every forest decomposition $\hat{q} \cup q_1(x_1) \cup \cdots \cup q_n(x_n)$ of q_0 there is no locally injective homomorphism $h : \hat{q} \to \mathcal{D} \cup \{A(a) \mid A \in \mu(a), a \in \mathsf{cons}(\mathcal{D})\}$ such that $h(x_0) = a_0$ and $\mathcal{K} \models^i_\mu q^j_R(h(x_j))$ for all $R \in \mathsf{rel}^-_2$ and $1 \leqslant j \leqslant n$.

2.39. Lemma. *$\mathcal{K} \not\models^i q_0(a_0)$ iff there is a small $\mathcal{K}$-realizable extended $\mathcal{O}$-type assignment for $\mathcal{D}$ that avoids q_0.*

Proof.

($\Leftarrow$) We are given a $\mathcal{K}$-realizable small q_0-extended $\mathcal{O}$-type assignment μ for $\mathcal{D}$ that avoids q_0. We show that any $\mathcal{ALCQI}$-forest model $\mathfrak{A} \models \mathcal{K}$ that realizes μ satisfies $\mathfrak{A} \not\models^i q_0(a_0)$. Suppose the contrary for contradiction. Then there exists a locally injective homomorphism $h : q_0 \to \mathfrak{A}$ with $h(x_0) = a_0^{\mathfrak{A}}$. Since $\mathfrak{A}$ is a forest model, the preimages of h induce a forest decomposition $\hat{q}, q_1, \ldots, q_n$ of q_0, with $\hat{q}$ induced by the preimage of $\{c^{\mathfrak{A}} : c \in \mathsf{cons}(\mathcal{D})\}$ and the $q_i, 1 \leqslant i \leqslant n$, by the preimages of trees of root $c^{\mathfrak{A}}$ for each $c \in \mathsf{cons}(\mathcal{D})$. It is then immediate that h is a witness for the fact that μ does not avoid q_0: $\mathfrak{A} \models \mathcal{D} \cup \{A(a) \mid A \in \mu(a), a \in \mathsf{cons}(\mathcal{D})\}$ since $\mathfrak{A}$ realizes μ, and h is a locally injective homomorphism from $\hat{q}$ to $\mathfrak{A}$ thus from $\hat{q}$ to $\mathcal{D} \cup \{A(a) \mid A \in \mu(a), a \in \mathsf{cons}(\mathcal{D})\}$.

($\Rightarrow$) Let $\mathfrak{A} \models \mathcal{K}$ such that $\mathfrak{A} \not\models^i q_0(a_0)$. We can assume without loss of generality that $\mathfrak{A}$ is an $\mathcal{ALCQI}$-forest model and has outdegree $\leqslant ||\mathcal{O}||$ outside the database elements (same argument as Cor. 1.37). For each $a \in \mathsf{cons}(\mathcal{D})$, role R and q_0-extended $\mathcal{O}$-type t, define as $\mu(a)$ the extended $\mathcal{O}$-type satisfied by $a^{\mathfrak{A}}$ and as $\mu(a, R, t)$ the number of R-successors outside $\mathcal{D}$ that satisfy t. Then, μ is small and avoids q_0: suppose for contradiction that it does not. Let it be witnessed by some forest decomposition $\hat{q}, q_1(x_1), \ldots, q_n(x_n)$ of some CQ q of q_0 and some homomorphism h as in the definition of avoiding q_0. Then, $\mathcal{K} \models^i_\mu q^j_R(h(x_j))$ for all j, r. In particular, $\mathfrak{A} \models \mathcal{K}$ and $\mathfrak{A}$ realizes μ, so $\mathfrak{A} \models^i q^j_r(h(x_j))$, witnessed by some locally injective homomorphism $h_{j,r}$. Then, the locally injective homomorphism $h \cup \bigcup_{j,r} h_{j,r}$ witnesses $\mathfrak{A} \models^i q(a_0)$, which is contradictory. ⊣

2.40. Lemma. *Locally injective unary rooted UCQ evaluation on $\mathcal{ALCQI}$-knowledge bases is* Exp*-complete in combined complexity.*

Proof. After Lemma 2.39 it suffices to show that, in the size of $\mathcal{K}$ and q_0,

1. the number of small q_0-extended $\mathcal{O}$-type assignments for $\mathcal{D}$ is at most exponential (Lem. 2.41),
2. it can be checked in exponential time whether a small q_0-extended $\mathcal{O}$-type assignment is $\mathcal{K}$-realizable (Lem. 2.45),
3. it can be checked in exponential time whether a small $\mathcal{K}$-realizable q_0-extended $\mathcal{O}$-type assignment avoids q_0 (Lem. 2.46).

⊣

2.41. Lemma. *The number of small q_0-extended $\mathcal{O}$-type assignments for $\mathcal{D}$ is at most exponential in $||\mathcal{K}|| + ||q_0||$.*

Proof. There are at most exponentially many q_0-extended $\mathcal{O}$-types, since there are at most exponentially many $\mathcal{O}$-types and Γ (thus Γ_0) has polynomial size. It then suffices to show that the number of functions

$$f : \left\{ (a, R, t) \;\middle|\; \begin{array}{l} a \in \mathsf{cons}(\mathcal{D}) \\ R \in \mathsf{rel}_2 \cap \mathsf{sig}(\mathcal{K}) \\ t \ q_0\text{-extended } \mathcal{O}\text{-type} \end{array} \right\} \to \{0, \ldots, ||\mathcal{O}||\}$$

$$\text{such that } \Sigma_t f(a, R, t) \leqslant ||\mathcal{O}||$$

is at most exponential. As there are at most exponentially many q_0-extended $\mathcal{O}$-types, the sum restrictions imply that any such function f satisfies $f(a, R, t) \neq 0$ on only polynomially many t for each (a, R), therefore on only polynomially many triples (a, R, t), given that there are only polynomially many $a \in \mathsf{cons}(\mathcal{D})$ and $R \in \mathsf{rel}_2 \cap \mathsf{sig}(\mathcal{K})$. ⊣

To prove Lemma 2.45, we extrapolate the so-called 'elimination procedure' used to determine the complexity of satisfiability in $\mathcal{ALC}$ [BHLS17].

2.42. *Definition.* For any tree-shaped CQ $q(x)$ and role S, we call *S-subtree of $q(x)$* any CQ corresponding to a subtree of the graph associated with $q(x)$ that is rooted in an S-successor of x. Let $\mathsf{Tr}_{\mathcal{O}}^{q_0}$ be the set of all triples (t,R,t') where R is a (possibly inverse) role and t,t' q_0-extended $\mathcal{O}$-types.

We aim at eliminating all 'bad' triples from $\mathsf{Tr}_{\mathcal{O}}^{q_0}$, *i.e.* those whose number restrictions cannot be witnessed simultaneously. Since a 'good' triple can become bad after one of its bad witnesses is eliminated, we need to make successive elimination rounds until we reach a fixed point. Let E be the function performing one round of elimination.

2.43. *Definition.* For any subset T of $\mathsf{Tr}_{\mathcal{O}}^{q_0}$, let $E(T)$ be defined as follows. For each $(t,R,t') \in T$, consider all possible 'combinations of successor triples' for (t,R,t'), *i.e.* sets of the form $S = \{(t',R_1,t_1)^{\alpha_1},\ldots,(t',R_\ell,t_\ell)^{\alpha_\ell}\}$ where $\ell \geqslant 1, (t',R_i,t_i) \in T, \alpha_i \geqslant 1$ for all $i \in \{1,\ldots,\ell\}$, such that $\sum_{i=1}^{\ell} \alpha_i \leqslant ||\mathcal{O}||$ (each α_i will indicate the multiplicity of the i-th triple). Check for all such S whether it is a *witness for* (t,R,t') (defined below) until one is. In that case, move on to the next triple. If (t,R,t') has no witness set, remove the triple from T. Let $E(T)$ denote the set that remains after applying the procedure to every triple in T.

We now define what it means for S to be a witness for (t,R,t'), *i.e.* that S respects all number restrictions and isat conditions in t'. Formally, S is a witness for (t,R,t') if

1. for all concepts of the form $\leqslant nS.C$ in t', $\sum_{C\in t_i, S=R_i} \alpha_i + \mathbf{1}_{C\in t,\ S=R^-} \leqslant n$,
2. for all concepts of the form $\geqslant nS.C$ in t', $\sum_{C\in t_i, S=R_i} \alpha_i + \mathbf{1}_{C\in t,\ S=R^-} \geqslant n$,
3. for each role ρ and $q \in \Gamma_0$, there is a bijection

$$f : \{\rho\text{-subtrees of } q \mid \mathsf{isat}(q) \in t'\} \to \{(t',R_i,t_i) \in S \mid R_i = \rho\}$$

   such that $f(q') = (t,R_i,t_i)$ implies $\mathsf{isat}(q') \in t_i$.

For all $i \geqslant 1$ we write E^i for $E \circ \cdots \circ E$ i times. Then, let $m = \min\{i \mid E^i(\mathsf{Tr}_{q_0}^{\mathcal{O}}) = E^{i+1}(\mathsf{Tr}_{q_0}^{\mathcal{O}})\}$ and $\mathsf{Elim}(\mathsf{Tr}_{q_0}^{\mathcal{O}}) = E^m(\mathsf{Tr}_{q_0}^{\mathcal{O}})$.

The next step is to characterise realizability in an algorithmically checkable way (Lem. 2.44), in order to deduce its complexity in Lemma 2.45.

2.44. Lemma. *A small q_0-extended $\mathcal{O}$-type assignment μ is $\mathcal{K}$-realizable if and only if the following conditions are met.*

1. $A(a) \in \mathcal{D} \Rightarrow A \in \mu(a)$ *for all* $a \in \mathsf{cons}(\mathcal{D})$, $A \in \mathsf{rel}_1$.
2. *Let* $a \in \mathsf{cons}(\mathcal{D})$ *and* $R \in \mathsf{rel}_2$. *Write all* $\geqslant r$ *number restrictions in* $\mu(a)$ *as* $\{\geqslant \lambda_j R.C_j : j \in J\}$ *and all* $\geqslant R^-$ *number restrictions in* $\mu(a)$ *as* $\{\geqslant \lambda_j R^-.C_j : j \in J'\}$. *Then*

   for all $j \in J$, $|\{b : C_j \in \mu(b), R(a,b) \in \mathcal{D}\}| + \sum_{C_j \in t} \mu(a,R,t) \geqslant \lambda_j$, *for all* $j \in J'$, $|\{b : C_j \in \mu(b), R(b,a) \in \mathcal{D}\}| + \sum_{C_j \in t} \mu(a, R^-, t) \geqslant \lambda_j$.

Same for $\leqslant R$ *number restrictions, with all occurrences of* $\geqslant$ *replaced by* $\leqslant$.

3. $\mu(a,R,t) \neq 0 \Rightarrow (\mu(a),R,t) \in \mathsf{Elim}(\mathsf{Tr}^{q_0}_{\mathcal{O}})$ *for all* $R \in \mathsf{rel}_2^-$ *and* q_0*-extended* $\mathcal{O}$*-types* t.

Proof. ($\Rightarrow$) Suppose μ is small and $\mathcal{K}$-realizable by some $\mathfrak{A}$. Then (1) is satisfied immediately. For (2) and (3), take as witnesses the types of successors.

($\Leftarrow$) We build a model $\mathfrak{A}$, prove it is a model of $\mathcal{K}$ and that it realizes μ. We first define $\mathfrak{A}_0$ as

$$\begin{aligned} \mathsf{dom}(\mathfrak{A}_0) &= \mathsf{cons}(\mathcal{D}) \\ R^{\mathfrak{A}_0} &= \{(a,b) : R(a,b) \in \mathcal{D}\} && \text{for all } R \in \mathsf{rel}_2 \\ A^{\mathfrak{A}_0} &= \{a \in \mathsf{cons}(\mathcal{D}) : A \in \mu(a)\} && \text{for all } A \in \mathsf{rel}_1 \end{aligned}$$

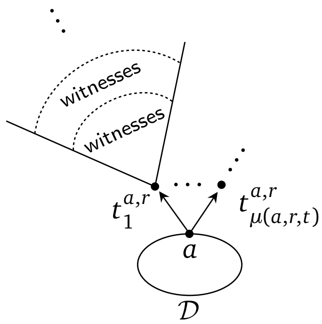
To construct $\mathfrak{A}_1$ from $\mathfrak{A}_0$, add $\mu(a,R,t)$ copies of t as R-successors of every $a \in \mathsf{dom}(\mathfrak{A}_0)$, for all $R \in \mathsf{rel}_2^-$ and q_0-extended $\mathcal{O}$-types t:

$$\mathsf{dom}(\mathfrak{A}_1) = \mathsf{dom}(\mathfrak{A}_0) \cup \left\{ t_i^{a,r} \;\middle|\; \begin{array}{l} a \in \mathsf{dom}(\mathfrak{A}_0), R \in \mathsf{rel}_2^- \\ t \ q_0\text{-extended type} \\ i \in \{1,\dots,\mu(a,R,t)\} \end{array} \right\}$$

$$R^{\mathfrak{A}_1} = R^{\mathfrak{A}_0} \cup \{(a, t_i^{a,s}) \mid t_i^{a,s} \in \mathsf{dom}(\mathfrak{A}_1), S = R\} \qquad \text{for all } R \in \mathsf{rel}_2^-$$

$$A^{\mathfrak{A}_1} = A^{\mathfrak{A}_0} \cup \{t_i^{a,r} \in \mathsf{dom}(\mathfrak{A}_1) \mid A \in t\} \qquad \text{for all } A \in \mathsf{rel}_1.$$

We only informally describe how to construct $\mathfrak{A}_k$ from $\mathfrak{A}_{k-1}$ for all $k \geqslant 2$, as a formal description would be too cumbersome. For each $x \in \mathsf{dom}(\mathfrak{A}_{k-1}) \setminus \mathsf{dom}(\mathfrak{A}_{k-2})$, let t' be the q_0-extended $\mathcal{O}$-type satisfied by x and t the one satisfied by its unique predecessor (as this is all in a tree), where R is the role linking the two. Then $(t,R,t') \in \mathsf{Elim}(\mathsf{Tr}^{q_0}_{\mathcal{O}})$ by an easy induction, as we assume condition (3). Consider a set $\{(t',R_1,t_1)^{\alpha_1},\dots,(t',R_\ell,t_\ell)^{\alpha_\ell}\}$ witnessing that $(t,R,t') \in \mathsf{Elim}(\mathsf{Tr}^{q_0}_{\mathcal{O}})$. To build $\mathfrak{A}_k$, we then add, for each $i \in \{1,\dots,\ell\}$, α_i copies of t_i as R_i-successors of t'. Naturally, any concept name A is satisfied in the copy of t_i iff $A \in t_i$. Finally, set $\mathfrak{A} = \bigcup_{k \geqslant 0} \mathfrak{A}_k$. We now show that $\mathfrak{A} \models \mathcal{K}$. It is easy to check that $t \in C^{\mathfrak{A}}$ iff $C \in t$, for any copy of t in $\mathfrak{A}$ and any $C \in \mathsf{sub}(\mathcal{O})$. Let $C \sqsubseteq D \in \mathcal{O}$. If a copy of t is in $C^{\mathfrak{A}}$ then $C \in t$. By maximality of types and consistency with respect to $\mathcal{K}$, it follows that $D \in t$, hence the copy of t is in $D^{\mathfrak{A}}$. Finally, the fact that $\mathfrak{A}$ realizes μ is immediate from its definition. ⊣



Construction of $\mathfrak{A}$.

2.45. Lemma. *$\mathcal{K}$-realizability of a small q_0-extended $\mathcal{O}$-type assignment for $\mathcal{D}$ is decidable in exponential time in the size of $\mathcal{K}, q_0$.*

Proof. Let μ be a small q_0-extended $\mathcal{O}$-type assignment, of bound $X \leqslant ||\mathcal{O}||$. We first need to determine $\mathsf{Elim}(\mathsf{Tr}_{\mathcal{O}}^{q_0})$. The smallest i such that $E^i(\mathsf{Tr}_{\mathcal{O}}^{q_0}) = E^{i+1}(\mathsf{Tr}_{\mathcal{O}}^{q_0})$ is bounded by $|\mathsf{Tr}_{\mathcal{O}}^{q_0}|$, which is polynomial in the total number of q_0-extended $\mathcal{O}$-types, therefore at most exponential. To determine $E^{i+1}(\mathsf{Tr}_{\mathcal{O}}^{q_0})$ from $E^i(\mathsf{Tr}_{\mathcal{O}}^{q_0})$, we check whether at most the total number of triples in $E^i(\mathsf{Tr}_{\mathcal{O}}^{q_0})$ (which is also at most exponential) is 'good' in $E^i(\mathsf{Tr}_{\mathcal{O}}^{q_0})$. To check that a given triple (t, R, t') is 'good' in $E^i(\mathsf{Tr}_{\mathcal{O}}^{q_0})$, we check at most all sets of triples (with multiplicity) of cardinality $\leqslant ||\mathcal{O}||$ in $E^i(\mathsf{Tr}_{\mathcal{O}}^{q_0})$. Checking that a given such set respects restrictions in t' takes polynomially many steps in the size of t' and the given set. We then check the conditions given by Lemma 2.44. Condition (1) can be checked in polynomial time. Condition (2) can be checked in exponential time: the procedure is the same as to check that a set of triples is 'good'. Condition (3) is checked by determining whether $(\mu(a), R, t) \in \mathsf{Tr}_{\mathcal{O}}^{q_0}$ for a number of triples $(\mu(a), R, t)$ at most as large as the size of μ (exponential in $\mathcal{K}, q_0$). ⊣

2.46. Lemma. *Whether a small q_0-extended $\mathcal{O}$-type assignment avoids q_0 is decidable in exponential time (in the size of $\mathcal{K}, q_0$).*

Proof. Assume μ is given. There are at most exponentially many forest decompositions $\hat{q} \cup q_1 \cup \cdots \cup q_n$ of q, because Γ_1 has polynomial size. Finding all homomorphisms $h : \hat{q} \to \mathcal{D} \cup \{A(a) : A \in \mu(a)\}$ can be done in exponential time. For each of them, we can then check that $h(x_0) = a_0$ and $\mathcal{K} \vDash_\mu^i q_R^j(h(x_j))$ for all $R \in \mathsf{rel}_2^-$ and $j \in \{1, \ldots, n\}$. For a given R, denote by S_R^j the set of R-subtrees in q_R^j. Note that $S_j^R \subseteq \Gamma_1$. Then, $\mathcal{K} \vDash_\mu^i q_R^j(h(x_j))$ iff for all $p \in S_R^j$, $\sum_{\mathsf{isat}(p) \in t} \mu(h(x_j), R, t) \geqslant 1$, which can be checked in polynomial time. ⊣

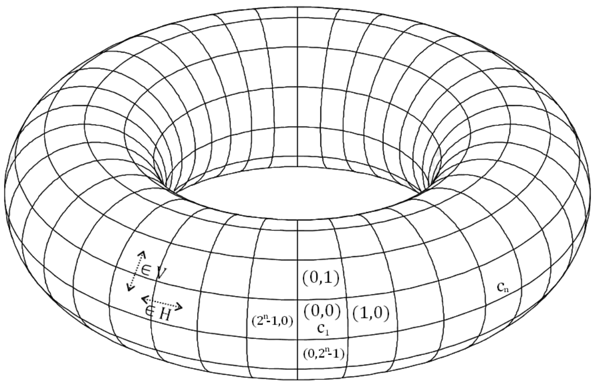
2.3.10. *Complexity for $\mathcal{ALC}$*

We show that full weak projective $(\mathcal{ALC}, \mathcal{ALC})$-separability is NExp-complete in combined complexity and PSpace-complete in data complexity.

For combined complexity, the proof is a simple transposition of the reductions used for $\mathcal{ALCI}$ in Corollary 2.33, but using the $\mathcal{ALC}$ (reachable) variant of unary rooted (U)CQ evaluation, on $\mathcal{ALC}$-knowledge bases. Unlike the $\mathcal{ALCI}$ variant's, the complexity of that decision problem is not known. We show that it is coNExp-complete. We prove the lower bound in Proposition 2.51, using a reduction from the tiling problem introduced in Definition 2.47 & Proposition 2.48, and the upper bound in Proposition 2.53.

2.47. *Definition.* Let an *initialized square tiling system* τ be a triple (T, H, V), where $T = \{0, \ldots, k\}$ is a finite set of *tile types* and $H, V \subseteq T \times T$ represent the *horizontal and vertical matching conditions*. An *initial condition* for τ takes the form $c = (c_0, \ldots, c_{n-1}) \in T^n$. A mapping $\sigma : \{0, \ldots, 2^n - 1\}^2 \to T$ is an *exponential toric solution* for τ given c if for all $x, y < 2^n$, the following holds (where $\oplus_i$ denotes addition modulo i).

1. If $\sigma(x, y) = t_1$ and $\sigma(x \oplus_{2^n} 1, y) = t_2$, then $(t_1, t_2) \in H$
2. If $\sigma(x, y) = t_1$ and $\sigma(x, y \oplus_{2^n} 1) = t_2$, then $(t_1, t_2) \in V$
3. $\sigma(i, 0) = c_i$ for all $i < n$.



The core of our lower bound proof follows from the next result, due to [Lu02, Cor. 4.15].

2.48. Proposition. *There exists an initialized square tiling system τ such that it is* NExp*-complete to decide, given an initial condition c, whether there exists an exponential toric solution for τ given c.*

Let $\tau = (T, H, V)$ be the tiling problem witnessing Proposition 2.48. Towards a reduction to reachable unary rooted CQ evaluation, we construct an $\mathcal{ALC}$-ontology $\mathcal{O}_{\tau,c}$ and a CQ $q_{\tau,c}$.

2.49. *Definition.* We define $\mathcal{O}_{\tau,c}$. It consists of 4 groups of axioms whose behaviour is informally described as follows.

1. Below each instance of a distinguished concept name A_0, there is a binary tree of depth $2n$ whose edges are represented by a role name S. The leaves of the tree correspond to the positions in the $2^n \times 2^n$-torus and the position of each leaf is represented in binary using the concept names $X_1, \ldots, X_n$ for the x-coordinate and $Y_1, \ldots, Y_n$ for the y-coordinate. We write $\neg^j C$ for $\neg\neg^{j-1}C$ and $\neg^0 C = C$ for every concept C and $j \geqslant 1$.

$$\begin{array}{rl} A_i \sqsubseteq \neg A_j \sqcap \exists S.A_{i0} \sqcap \exists S.A_{i1} & \text{for all distinct } i,j \in \{0,1\}^* \\ A_{i_1 \ldots i_{2n}} \sqsubseteq \bigsqcap_{j=1}^{n} \neg^{i_j} X_j \sqcap \bigsqcap_{j=n+1}^{2n} \neg^{i_j} Y_j & \text{for all } i_1, \ldots, i_{2n} \in \{0,1\} \end{array}$$

2. Each 'leaf' has three additional successors, all attached via the role name R, marked with the concept names H (for 'here'), U (for 'up'), and R (for 'right').

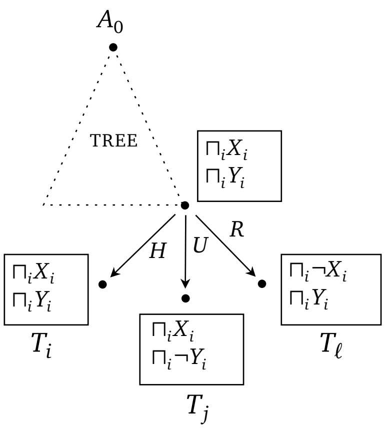
$$A_0 \sqsubseteq \forall S^{2n}.(\exists R.\top \sqcap \exists U.\top \sqcap \exists H.\top)$$

3. The three successors are also associated with torus positions represented via $X_1, \ldots, X_n$ and $Y_1, \ldots, Y_n$. If the position of the leaf that the successors are attached to is (i,j), then the H-successor also has position (i,h), the U-successor has position $(i, j \oplus_{2^n} 1)$ and the R-successor has position $(i \oplus_{2^n} 1, j)$. For every n-digit binary number $b = b_1 \ldots b_n$ we define $C_b^X = \bigsqcap_{i=1}^n \neg^{b_i} X_i$ and $C_b^Y = \bigsqcap_{i=1}^n \neg^{b_i} Y_i$.

$$\begin{array}{rl} C_b^X \sqsubseteq \forall R.C_{b \oplus_n 1}^X & \text{for all } b \leqslant 2^n \\ C_b^Y \sqsubseteq \forall U.C_{b \oplus_n 1}^Y & \text{for all } b \leqslant 2^n \\ (C_b^X \sqcap C_{b'}^Y) \sqsubseteq \forall H.(C_b^X \sqcap C_{b'}^Y) & \text{for all } b, b' \leqslant 2^n \end{array}$$

4. To each of the successor nodes is assigned a tile from T, where tile m is represented by concept name T_m. The assignment needs to be compatible with c and 'locally compatible' with the matching conditions, that is, if tiles i, j, ℓ are assigned to the H-, U-, and R-successor of the same tree leaf, then $(i,j) \in V$ and $(i,\ell) \in H$.

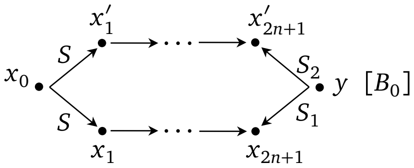
$$\begin{array}{rl} C_b^X \sqcap C_{0\ldots0}^Y \sqsubseteq \forall H.T_{c_b} & \text{for all } b \leqslant n \\ \exists H.T_m \sqsubseteq \forall R.\bigsqcup_{(m,m') \in H} T_{m'} & \text{for all } m \in T \\ \exists H.T_m \sqsubseteq \forall U.\bigsqcup_{(m,m') \in V} T_{m'} & \text{for all } m \in T \\ T_m \sqsubseteq \neg T_{m'} & \text{for all } m \neq m' \end{array}$$



Depiction of the binary tree described above, with a detailed look at the 3 successors of an example leaf of coordinates $\bigsqcap_i X_i$ and $\bigsqcap_i Y_i$, i.e. $(2^n, 2^n)$.

We then define the following CQ $q_{\tau,c}(x_0)$, represented below, by

$$\exists x_1 \ldots \exists x_{2n+1} \exists x'_1 \ldots \exists x'_{2n+1} \exists y \left(\begin{array}{l} S(x_0, x_1) \wedge S(x_0, x'_1) \wedge \\ \bigwedge_{i=1}^{2n} (S(x_i, x_{i+1}) \wedge S(x'_i, x'_{i+1})) \wedge \\ S_1(y, x_{2n+1}) \wedge S_2(y, x'_{2n+1}) \wedge B_0(y) \end{array} \right)$$



and further extend $\mathcal{O}_{\tau,c}$ with the following, for $1 \leqslant i \leqslant n$.

$$\begin{array}{l} B_0 \sqsubseteq (\forall S_1.X_i \sqcap \forall S_2.X_i) \sqcup (\forall S_1.\neg X_i \sqcap \forall S_2.\neg X_i) \\ B_0 \sqsubseteq (\forall S_1.Y_i \sqcap \forall S_2.Y_i) \sqcup (\forall S_1.\neg Y_i \sqcap \forall S_2.\neg Y_i) \\ B_0 \sqsubseteq \bigsqcup_{i,j \in T, i \neq j} (\forall S_1.T_i \sqcap \forall S_2.T_j) \end{array}$$

$\mathcal{O}_{\tau,c}$ and $q_{\tau,c}$ can be constructed in polynomial time, thus the next lemma suffices to obtain the lower bound.

2.50. Lemma. $(\mathcal{O}_{\tau,c}, \{A_0(a)\}) \models^r q_{\tau,c}(a)$ *iff* τ *has no exponential toric solution given* c.

Proof. ($\Leftarrow$) Let $\mathfrak{A} \models \mathcal{O}_{\tau,c}$ with $a \in A_0^{\mathfrak{A}}$. Then $\mathcal{O}_{\tau,c}$ generates a tree below a as described above. Since τ has no exponential toric solution given c, this tree must contain a tiling defect, *i.e.* there must be two elements d_1, d_2 reachable from a along an S-path of length $2n+1$ such that d_1, d_2

1. are associated with the same position, *i.e.* $d_1 \in X_i^{\mathfrak{A}}$ iff $d_2 \in X_i^{\mathfrak{A}}$ for $1 \leqslant i \leqslant n$ and $d_1 \in Y_i^{\mathfrak{A}}$ iff $d_2 \in Y_i^{\mathfrak{A}}$ for $1 \leqslant i \leqslant n$; and

2. are tiled differently, *i.e.* $d_1 \in T_i^{\mathfrak{A}}$ and $d_2 \in T_j^{\mathfrak{A}}$, $i \neq j$.

Clearly, there is a homomorphism h from $(q_{\tau,c})^{\uparrow}_{x_0}$ to $\mathfrak{A}$ with $h(x_{2n+1}) = d_1$ and $h(x'_{2n+1}) = d_2$. It is straightforward that, because d_1, d_2 satisfy Conditions 1 and 2 above and because of the axioms involving B_0, h satisfies the required satisfiability condition, *i.e.* that there exists a model of the knowledge base

$$\big(\mathcal{O}_{\tau,c}, \mathcal{D}_{q_{\tau,c}} \cup \{C(y) \mid C \in \mathsf{sub}(\mathcal{O}_{\tau,c}), y \in \mathsf{var}((q^{\uparrow}_{\tau,c})_{x_0}), h(y) \in C^{\mathfrak{A}}\}\big).$$

($\Rightarrow$) Assume that τ has a solution given c. We can then find a model $\mathfrak{A}$ of $\mathcal{O}_{\tau,c}$ with $a \in A_0^{\mathfrak{A}}$ such that the tree enforced by $\mathcal{O}_{\tau,c}$ below a represents that solution. In particular, there is no tiling defect. Consequently, all homomorphisms from $(q_{\tau,c})^{\uparrow}_{x_0}$ to $\mathfrak{A}$ violate the satisfiability condition. ⊣

The result ensues.

2.51. Proposition. *Reachable unary rooted (U)CQ evaluation on $\mathcal{ALC}$-knowledge bases is* coNExp*-hard in combined complexity.*

We next determine a coNExp upper bound. It requires the following lemma that can be easily checked and states that two models of an ontology $\mathcal{O}$ can be joined at a node to form another model of $\mathcal{O}$ if that node has the same $\mathcal{O}$-type in both models.

2.52. Lemma. *Let $\mathcal{O}$ be an $\mathcal{ALC}$-knowledge base, $\mathfrak{A}_1, \mathfrak{A}_2 \models \mathcal{O}$ and x such that* $\mathsf{dom}(\mathfrak{A}_1) \cap \mathsf{dom}(\mathfrak{A}_2) = \{x\}$ *and* $\mathsf{tp}_{\mathcal{O}}(\mathfrak{A}_1, x) = \mathsf{tp}_{\mathcal{O}}(\mathfrak{A}_2, x)$*. Then $\mathfrak{A}_1 \cup \mathfrak{A}_2 \models \mathcal{O}$.*

2.53. Proposition. *Reachable unary rooted (U)CQ evaluation on $\mathcal{ALC}$-knowledge bases is in* coNExp *in combined complexity.*

Proof. Let $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ be an $\mathcal{ALC}$-knowledge base, q a unary rooted UCQ, and $a \in \mathsf{cons}(\mathcal{D})$. Then $\mathcal{K} \models^r q(a)$ iff for every $\mathcal{ALC}$-forest model $\mathfrak{A} \models \mathcal{K}$ of outdegree at most $||\mathcal{O}||$, $\mathfrak{A} \models^r q(a)$. We omit the full proof as it mirrors the proofs of Lemmas 1.40 and 1.42: if $\mathfrak{A} \not\models^r q(a^{\mathfrak{A}})$, then by forest unfolding and "trimming" we obtain a forest model of bounded outdegree $\mathfrak{B}$ such that $\mathfrak{B} \not\models^r q(a^{\mathfrak{B}})$. A NExp algorithm for the complement of reachable unary rooted UCQ evaluation on $\mathcal{ALC}$-knowledge bases is now as follows. Let $\mathcal{K}$, q, and a be given as an input. We guess an initial piece of a forest model $\mathfrak{A}$ of $\mathcal{K}$ of outdegree at most $||\mathcal{O}||$ and depth at most $||q||$. The number of elements in such an initial piece is single exponential, bounded by $||\mathcal{D}|| + ||\mathcal{D}|| \cdot ||\mathcal{O}||^{||q||}$. Along with $\mathfrak{A}$, we guess an adornment $\mu : \mathsf{dom}(\mathfrak{A}) \to 2^{\mathsf{sub}(\mathcal{O})}$ that specifies which subconcepts of $\mathcal{O}$ are satisfied at each element in $\mathfrak{A}$. It is required that for all $d \in \mathsf{dom}(\mathfrak{A})$, $\bigsqcap \mu(d)$ is consistent w.r.t. $\mathcal{O}$, which can be checked in exponential time in $||\mathcal{K}|| + ||q||$. The adornment must also be compatible with $\mathfrak{A}$, that is

1. $d \in A^{\mathfrak{A}}$ iff $A \in \mu(d)$ for all concept names A;
2. if $(d, e) \in R^{\mathfrak{A}}$, $C \in \mu(e)$ and $\exists R.C \in \mathsf{sub}(\mathcal{O})$, then $\exists R.C \in \mu(d)$.

By Lemma 2.52, the adornment ensures that the guessed initial piece of $\mathfrak{A}$ can be extended to a full $\mathcal{ALC}$-forest model of $\mathcal{K}$. Since $q(x)$ is rooted, however, only the guessed initial piece of $\mathfrak{A}$ can be in the range of a homomorphism h from $q_x^\uparrow$ to $\mathfrak{A}$ that maps x to a database constant. The number of homomorphisms from $q_x^\uparrow$ to $\mathfrak{A}$ is single exponential, bounded by $|\mathsf{dom}(\mathfrak{A})|^{||q||}$, thus we can iterate through all candidates. For each h that turns out to be a homomorphism, we additionally verify that $\mathcal{K}_{\mathfrak{A},h}$ (which has polynomial size in K, q) is satisfiable in exponential time (cf. complexity of satisfiability in $\mathcal{ALC}$). We accept iff every homomorphism violates the satisfiability condition. $\dashv$

In conclusion,

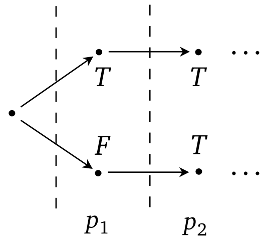
2.54. Theorem. *Full weak projective $(\mathcal{ALC}, \mathcal{ALC})$-separability is* NEXP*-complete in combined complexity.*

Data complexity, lower bound. We show that when the ontology is removed from the input, complexity drops to PSPACE-completeness. For the lower bound, it suffices to show that rooted UCQ evaluation on $\mathcal{ALC}$-knowledge bases $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ is PSPACE-hard in the size of 'database + query'. The reduction to simple (not necessarily reachable) rooted UCQ evaluation is sufficient, as it provides a query $q(x)$ such that $q_x^\uparrow = q$. The reduction is from validity of quantified Boolean formulas (QBF), well-known to be PSPACE-complete. We assume without loss of generality that the input QBF is of the form $\varphi = \forall p_1 \exists p_2 \forall p_3 \cdots \exists p_n \psi$ with $\psi = \psi_1 \wedge \cdots \wedge \psi_m$ in conjunctive normal form. We require the following definitions.

2.55. *Definition.* We call $\forall\exists$*-tree* any tree-shaped graph in which every node of even depth has two children, every node of odd depth has one child, and every node is labeled with exclusively T or F. In any graph, we call the *corresponding tuple of truth values* of a path of length n the tuple $(v_1, \ldots, v_n)$ with v_i being the label (T or F) assigned to the path's $i+1$-th node. Let φ*-forbidden tuples* be all tuples $(v_1, \ldots, v_n) \in \{T, F\}^n$ such that assigning v_i to p_i for all i makes ψ false. In any graph, we say a path is φ*-forbidden* if its corresponding tuple of truth values is φ-forbidden.

2.56. *Example.*

1. $\forall p_1 \exists p_2 (p_1 \vee p_2) \wedge (\neg p_1 \vee p_2)$ is valid. Forbidden tuples are (F,F) and (T,F). Validity is witnessed by the existence of a $\forall\exists$-tree T_φ, that has no forbidden path from its root.



2. $\forall p_1 \exists p_2 (p_1 \wedge p_2)$ is not valid. Any pair containing F is forbidden. In any $\forall\exists$-tree, there exists a path starting from the root that contains F.

2.57. Theorem. *Full weak projective $(\mathcal{ALC},\mathcal{ALC})$-separability is* PSPACE*-hard in data complexity.*

Proof. Let $\mathcal{K} = (\mathcal{O},\mathcal{D})$ with $\mathcal{D} = \{U(a_0)\}$ and

$$\mathcal{O} = \left\{ \begin{array}{lcl} U & \sqsubseteq & \exists R.(E \sqcap T) \sqcap \exists R.(E \sqcap F) \\ E & \sqsubseteq & \exists R.U \\ T & \equiv & \neg F \end{array} \right\}$$

We use $R^i(x,y)$, $i \geqslant 1$, as shorthand for $R(z_1,z_2),\dots,R(z_{i-1},z_i)$ with $z_1 = x$, $z_i = y$, and $z_2,\dots,z_{i-1}$ existentially quantified fresh variables. We define a rooted UCQ q_φ, constructible in polynomial time, such that φ is valid iff $\mathcal{K} \not\models q_\varphi(a_0)$.

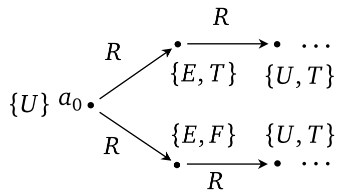
$$J_i^+ = \{j \in \{1,\dots,n\} \mid p_j \text{ occurs positively in } \psi_i\}$$
$$J_i^- = \{j \in \{1,\dots,n\} \mid p_j \text{ occurs negatively in } \psi_i\}$$
$$q_\varphi = \bigvee_{i=1}^{m} \Big[\exists x_1 \dots \exists x_n \; R(x_0,x_1) \wedge \dots \wedge R(x_{n-1},x_n) \wedge \bigwedge_{j \in J_i^+} F(x_j) \wedge \bigwedge_{j \in J_i^-} T(x_j)\Big]$$

We prove that φ is valid iff $\mathcal{K} \not\models q_\varphi(a_0)$. In CNF, for some $k_1,\dots,k_m \geq 1$, we can write $\varphi = \bigwedge_{1 \leqslant i \leqslant m} \bigvee_{1 \leqslant j \leqslant k_i} \ell_{i_j}$. Then, given a valuation χ, φ is false iff there exists at least one $i \in \{1,\dots,m\}$ such that for every $j \in \{1,\dots,k_i\}$ we have $V(p_{i_j}) = \overline{V}_{i_j}$, where for any propositional variable p, $V(p)$ is defined as T if $\chi(p) = 1$ and F otherwise. Therefore, φ is valid iff for every choice of p_1, there is a choice of p_2 such that for every [...], there is a choice of p_n such that those n choices make φ true, *i.e.* such that for all $i \in \{1,\dots,m\}$ there exists at least one $j \in \{1,\dots,k_i\}$ such that $V(p_{i_j}) \neq \overline{V}_{i_j}$. Using Definition 2.55, that statement can be rephrased as

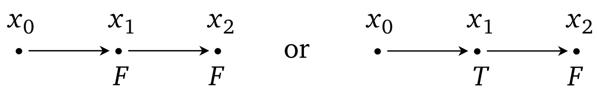
- φ is valid iff there exists a $\forall\exists$-tree with no φ-forbidden path from its root.

It can then be proved the righthand statement is equivalent to $\mathcal{K} \not\models q_\varphi(a_0)$. Suppose φ is valid. Then there exists a $\forall\exists$-tree T_φ with no forbidden φ-path from its root. By considering all edges of T_φ as R-edges, seeing the labels T/F as concept names and interpreting a_0 as the root, T_φ can be seen as a model. We then build a model $\mathfrak{A}_\varphi$ from T_φ by adding U to each node of even depth and E to each node of odd depth. It is then clear that $\mathfrak{A} \models \mathcal{K}$ and $\mathfrak{A}_\varphi \not\models q_\varphi(a_0^{\mathfrak{A}_\varphi})$. $\dashv$

2.58. In (1) of Example 2.56, T_φ induces a model $\mathfrak{A}_\varphi$ that contains the following submodel.



The induced query $q_\varphi(x_0)$ can be represented as below. Clearly, $\mathfrak{A}_\varphi \not\models q_\varphi(a_0^{\mathfrak{A}_\varphi})$.



Data complexity, upper bound. For the upper bound, it suffices to show that reachable rooted UCQ evaluation on $\mathcal{ALC}$-knowledge bases is in PSPACE when the ontology is fixed. We need the following definitions.

2.59. *Definition.* An *augmented database* is a database that may contain 'atoms' $\neg C(a)$, C an $\mathcal{EL}$-concept. An *augmented $\mathcal{ALC}$-knowledge base* is a pair $(\mathcal{O}, \mathcal{D})$ with $\mathcal{O}$ an $\mathcal{ALC}$-ontology and $\mathcal{D}$ an augmented database.

It has been shown in [Lu08, §5] that given an $\mathcal{ALC}$-knowledge base $(\mathcal{O}, \mathcal{D})$ and a CQ q, one can compute a sequence of augmented $\mathcal{ALC}$-knowledge bases $\mathcal{K}_1, \dots, \mathcal{K}_n$, $\mathcal{K}_i = (\mathcal{O}_i, \mathcal{D}_i)$ such that $\mathcal{K} \not\models q$ iff at least one $\mathcal{K}_i$ is satisfiable. The proof straightforwardly extends to reachable evaluation and an easy analysis shows that when q is rooted, then we can assume that $\mathcal{O}_i = \mathcal{O}$ for all i. Each database $\mathcal{D}_i$ is of size polynomial in $||\mathcal{D}|| + ||q||$ and the knowledge bases $\mathcal{K}_1, \dots, \mathcal{K}_n$, of which there are only single exponentially many in $||\mathcal{D}|| + ||q||$, can be enumerated using polynomial space. It thus suffices to show that for every fixed $\mathcal{ALC}$-ontology $\mathcal{O}$, given an augmented database $\mathcal{D}$, it can be decided in PSPACE whether the knowledge base $(\mathcal{O}, \mathcal{D})$ is satisfiable. We only sketch the procedure.

2.60. We can assume without loss of generality that $\neg\exists R.C(a) \in \mathcal{D}$ implies $\neg C(b) \in \mathcal{D}$ for all $R(a, b) \in \mathcal{D}$ and C. This is because that property can be checked in polynomial time and any $(\mathcal{O},\mathcal{D})$ that cannot be completed to satisfy it is already unsatisfiable. We can precompute in constant time the set S of $\mathcal{O}$-types. To check satisfiability of $(\mathcal{O},\mathcal{D})$, we first guess an assignment $\delta : \mathsf{cons}(\mathcal{D}) \to S$ of $\mathcal{O}$-types to constants in $\mathcal{D}$ such that

1. for all $R(a,b) \in \mathcal{D}$ where $R \in \mathsf{rel}_2$,
   - if $\exists R.C \in \mathsf{sub}(\mathcal{O})$, then $C \in \delta(b)$ implies $\exists R.C \in \delta(a)$, and
   - if $\forall R.C \in \mathsf{sub}(\mathcal{O})$, then $\forall R.C \in \delta(a)$ implies $C \in \delta(b)$,
2. for all $A(a) \in \mathcal{D}$ where $A \in \mathsf{rel}_1$, $A \in \delta(a)$.

It is then sufficient to check whether the concept $\bigsqcap_{C(a)\in\mathcal{D}} C \sqcap \bigsqcap \delta(a)$ is satisfiable w.r.t. $\mathcal{O}$, for every $a \in \mathsf{cons}(\mathcal{D})$.

2.61. Lemma. *If $\bigsqcap_{C(a)\in\mathcal{D}} C \sqcap \bigsqcap \delta(a)$ is satisfiable w.r.t. $\mathcal{O}$ for every $a \in \mathsf{cons}(\mathcal{D})$, then $(\mathcal{O},\mathcal{D})$ is satisfiable.*

Proof. Suppose that for all $c \in \mathsf{cons}(\mathcal{D})$ there is $\mathfrak{A}_c$ and some $x \in (\bigsqcap_{C(c)\in\mathcal{D}} C \sqcap \bigsqcap \delta(c))^{\mathfrak{A}_c}$. Then we can define a model $\mathfrak{A}$ by taking the disjoint union of all $\mathfrak{A}_c$ and connecting them at each c. We can assume without loss of generality that the models $\mathfrak{A}_c$ are all pairwise disjoint and that for each $c \in \mathsf{cons}(\mathcal{D})$, the witness x mentioned above is equal to $c^{\mathfrak{A}_c}$. Then define $\mathfrak{A}$ as follows.

$$\begin{aligned}
\mathsf{dom}(\mathfrak{A}) &= \bigcup_{c\in\mathsf{cons}(\mathcal{D})} \mathfrak{A}_c \\
R^{\mathfrak{A}} &= \bigcup_{c\in\mathsf{cons}(\mathcal{D})} R^{\mathfrak{A}_c} \cup \{(c^{\mathfrak{A}_c}, d^{\mathfrak{A}_d}) \mid R(c,d) \in \mathcal{D}\} && \text{for all } R \in \mathsf{rel}_2 \\
A^{\mathfrak{A}} &= \bigcup_{c\in\mathsf{cons}(\mathcal{D})} \mathfrak{A}_c && \text{for all } A \in \mathsf{rel}_1 \\
c^{\mathfrak{A}} &= c^{\mathfrak{A}_c} && \text{for all } c \in \mathsf{cons}(\mathcal{D})
\end{aligned}$$

Note that $C^{\mathfrak{A}} = \bigcup_{c\in\mathsf{cons}(\mathcal{D})} C^{\mathfrak{A}_c}$ for all $C \in \mathsf{sub}(\mathcal{O})$. It is clear if C is a concept name. Then assuming the induction hypothesis, let $x \in (\exists R.C)^{\mathfrak{A}}$ for some R, C. The only non-trivial case is, in the left to right inclusion, the case where $x = c^{\mathfrak{A}}$ and $x \in (\exists R.C)^{\mathfrak{A}}$ is witnessed by $d^{\mathfrak{A}}$ such that $R(c,d) \in \mathcal{D}$. Then $d \in C^{\mathfrak{A}}$ implies $d^{\mathfrak{A}} \in C^{\mathfrak{A}_d}$ by induction hypothesis, then $C \in \delta(d)$ since $C \in \mathsf{sub}(\mathcal{O})$ (therefore either $C \in \delta(d)$ or $\neg C \in \delta(d)$). From $C \in \delta(d)$ we get $\exists R.C \in \delta(c)$ by definition of δ, and finally $c^{\mathfrak{A}} \in C^{\mathfrak{A}_c}$. $\mathfrak{A} \models \mathcal{D}$ is clear. To show $\mathfrak{A} \models \mathcal{O}$, let $C \sqsubseteq D \in \mathcal{O}$. Since $\mathfrak{A}_c \models \mathcal{O}$ for all $c \in \mathsf{cons}(\mathcal{D})$ we have $C^{\mathfrak{A}} = \bigcup_{c\in\mathsf{cons}(\mathcal{D})} C^{\mathfrak{A}_c} \subseteq \bigcup_{c\in\mathsf{cons}(\mathcal{D})} D^{\mathfrak{A}_c} = D^{\mathfrak{A}}$. ⊣

To check satisfiability w.r.t. $\mathcal{O}$ in PSPACE, we adapt the well-known K-WORLD recursive procedure (*e.g.* [Sp93], p. 18) used to determine satisfiability of modal formulas (*i.e.* $\mathcal{ALC}$-concepts) w.r.t. an empty ontology.

2.62. *Definition.* Let Δ, Σ be sets of $\mathcal{ALC}$-concepts, and t an $\mathcal{O}$-type. Let $\textsc{k-world}(\Delta, \Sigma, t)$ be true iff

1. Δ is a maximally propositionally consistent subset of Σ, *i.e.*

$$\begin{aligned} C \in \Delta &\Rightarrow C \in \Sigma && \text{for all } C \\ C \in \Delta &\Leftrightarrow \neg C \notin \Delta && \text{for all } \neg C \in \Sigma \\ C_1 \sqcap C_2 \in \Delta &\Rightarrow C_1 \in \Delta \text{ and } C_2 \in \Delta && \text{for all } C_1 \sqcap C_2 \in \Sigma \end{aligned}$$

2. Δ is consistent with t, *i.e.*

$$\begin{aligned} C \in \Delta &\Rightarrow \neg C \notin t \\ C \in t &\Rightarrow \neg C \notin \Delta \end{aligned}$$

3. For all $\exists R.C \in \Delta$, there exists a set Δ_C and an $\mathcal{O}$-type t' such that $\textsc{k-world}(\Delta_C, \Sigma', t')$ is true, where Σ' is the closure under subconcepts and single negation of $\{D \mid \forall R.D \in \Delta\}$, and

$$\begin{aligned} &C \in \Delta_C \\ \forall R.D \in \Delta &\Rightarrow D \in \Delta_C && \text{for all } D \\ \forall R.D \in t &\Rightarrow D \in t' && \text{for all } D \end{aligned}$$

2.63. Lemma. $\textsc{k-world}(\Delta, \Sigma, t)$ *is true iff* Δ *is a maximally satisfiable subset of* Σ *in a pointed model satisfying* t.

Proof. ($\Rightarrow$) Suppose $\textsc{k-world}(\Delta, \Sigma, t)$ is true. At each recursive step of the algorithm (denote by $(\Delta_0, \Sigma_0, t_0)$ the triple being considered at that step) and each concept of the form $\exists R.C$ in Δ_0 pick a triple $(\Delta_{0,C}, \Sigma_0', t_0')$ witnessing condition (3) for $(\Delta_0, \Sigma_0, t_0)$. Let $\mathsf{dom}(\mathfrak{A})$ be the set of all such picked triples. For each $A \in \mathsf{rel}_1$ let $A^{\mathfrak{A}} = \{(\Delta, \Sigma, t) \in \mathsf{dom}(\mathfrak{A}) \mid A \in \Delta \cup t\}$. For each $R \in \mathsf{rel}_2$ let $R^{\mathfrak{A}}$ be all pairs of the form $((\Delta_0, \Sigma_0, t_0), (\Delta_{0,C}, \Sigma_0', t_0'))$. Then let $\mathfrak{B}$ be the model obtained by identifying each $(\Delta_0, \Sigma_0, t_0)$ with the root of a (tree) model $\mathfrak{A}_{t_0}$ of $\mathcal{O}$ witnessing the fact that t_0 is an $\mathcal{O}$-type. We can assume without loss of generality that all such trees are disjoint. Then, $\mathfrak{B}$ is well-defined and a (tree) model of $\mathcal{O}$: one easily checks (same idea as Lemma 2.52) that for all t, $C^{\mathfrak{A}_t} = C^{\mathfrak{B}} \cap \mathsf{dom}(\mathfrak{A}_t)$, which means that for every $C \sqsubseteq D \in \mathcal{O}$ we have $C^{\mathfrak{B}} = \bigcup_t C^{\mathfrak{A}_t} \subseteq \bigcup_t D^{\mathfrak{A}_t} = D^{\mathfrak{B}}$. It holds for all $C \in \Sigma$ that $\mathfrak{B} \models C$ iff $C \in \Delta$: the proof is exactly the same as for the case with no ontology ([Sp93]).

($\Leftarrow$) Suppose Δ is a maximally satisfiable subset of Σ in a pointed model satisfying t. Then (1) and (2) are straightforwardly true. For some pointed model $(\mathfrak{A}, a)$ and all $C \in \Sigma$, we have $\mathfrak{A} \models C(a)$ iff $C \in \Delta$ and $\mathsf{tp}_{\mathcal{O}}(\mathfrak{A}, a) = t$. Then, for every concept of the form $\exists R.C$ in Δ there exists a witnessing R-successor a' of a in $\mathfrak{A}$. That successor induces witnesses Δ_C and t' for condition (3), Δ_C

being the set of all $C \in \Sigma'$ such that $\mathfrak{A} \models C(a')$ and $t' = \mathsf{tp}_{\mathcal{O}}(\mathfrak{A}, a')$. All conditions are then satisfied. ⊣

The upper bound proof can now be concluded.

2.64. Theorem. *Full weak projective* $(\mathcal{ALC}, \mathcal{ALC})$*-separability is in* PSPACE *in data complexity.*

Proof. Let $D_a = \bigsqcap_{C(a) \in \mathcal{D}} C$. Then, the concept $\bigsqcap_{C(a) \in \mathcal{D}} C \sqcap \bigsqcap \delta(a)$ is satisfiable w.r.t $\mathcal{O}$ iff there exists $\Delta \subseteq \mathsf{sub}(D_a)$ containing D_a such that K-WORLD$(\Delta, \mathsf{sub}(D_a), \delta(a))$ is true. If the ontology is fixed, the problem to determine, given an $\mathcal{ALC}$-concept C and an $\mathcal{O}$-type t, whether there exist $\Delta \subseteq \mathsf{sub}(C)$ containing C and t such that K-WORLD$(\Delta, \mathsf{sub}(C), t)$ is true is in PSPACE in $||C||$. Since $|\Sigma'| < |\Sigma|$ at each level of recursion, the number of recursive calls is bounded by $|\mathsf{sub}(C)|$, hence polynomial in $||C||$. All subsets of $\mathsf{sub}(C)$ encountered in the algorithm can be represented using polynomial space in $||C||$. The $\mathcal{O}$-types are of constant size as $\mathcal{O}$ is fixed. The fact that we check (3) non-deterministically is unimpactful as NPSPACE = PSPACE [Sa70]. ⊣

Using the same argument as for the PSPACE upper bound for $\mathcal{ALC}$ we obtain (almost for free) an upper data complexity bound in the case when the ontology does not contain any quantifier. Let $\mathcal{ALC}^{\mathsf{prop}}$ be the propositional (or quantifier-free) fragment of $\mathcal{ALC}$.

2.65. Theorem. *For* $\mathcal{L} \in \mathbf{DL}_{\mathcal{IQ}}$*, full signature weak projective* $(\mathcal{ALC}^{\mathsf{prop}}, \mathcal{L})$*-separability is in* Σ_p^2 *in data complexity.*

Proof. It suffices to check, given $\mathcal{K}, q, a$, whether there exists $\mathfrak{A} \models \mathcal{K}$ such that $\mathfrak{A} \not\models^{\mathcal{L}} q(a)$. Because the ontology is propositional, it holds for any $\mathfrak{A} \models \mathcal{K}$ such that $\mathfrak{A} \not\models^{\mathcal{L}} q(a)$ that $\mathfrak{A}_{|\mathcal{D}} \models \mathcal{K}$ and $\mathfrak{A}_{|\mathcal{D}} \not\models^{\mathcal{L}} q(a)$, where $\mathfrak{A}_{|\mathcal{D}}$ is the restriction of $\mathfrak{A}$ to $\{c^{\mathfrak{A}} \mid c \in \mathsf{cons}(\mathcal{D})\}$. Then it suffices to guess a model $\mathfrak{A}$ of $\mathcal{D}$, check that $\mathfrak{A}_{|\mathcal{D}} \models \mathcal{O}$ (polynomial time), coguess mappings $f : q \to \mathfrak{A}$ and check whether f is (resp. injectively) homomorphic (polynomial time). If $\mathcal{L} = \mathcal{ALC}(\mathcal{Q})$, we also need to check unsatisfiability of $\mathcal{K}_{f,\mathfrak{A}}$. That is in CONP in the size of its (extended) database, which only extends $\mathcal{D}$ with concepts in $\mathsf{sub}(\mathcal{O})$, *i.e.* only adds constant size. Then, satisfiability of $\mathcal{K}_{f,\mathfrak{A}}$ is also in NP in $||\mathcal{D}||$, which concludes the proof. ⊣

We can get a Σ_p^2 lower bound by reduction from the problem to decide, given an undirected graph G and $k \geqslant 1$, whether there exists a 2-coloring of G that does not contain a monochromatic k-clique [Ru86].

2.66. Theorem. *For $\mathcal{L} \in \mathbf{DL}_{\mathcal{IQ}}$, full signature weak projective $(\mathcal{ALC}^{\mathsf{prop}}, \mathcal{L})$-separability is Σ^2_p-hard in data complexity.*

Proof. Let a graph $G = (V, E)$ and $k \geqslant 1$ be given. Then, G induces the following knowledge base and UCQ.

$$\begin{aligned}
\mathcal{O} &= \{C \sqsubseteq C_1 \sqcup C_2, C_1 \sqsubseteq \neg C_2\} \\
\mathcal{D}_G &= \{R(a_0, a_v), R(a_v, a_0) \mid v \in V\} \\
&\quad \cup \{R(a_u, a_v), R(a_v, a_u) \mid (u, v) \in E\} \\
q_G(x) &= \exists y_1 \ldots \exists y_k \bigwedge_{1 \leqslant i,j \leqslant k} (C_1(y_i) \wedge R(x, y_i) \wedge R(x, y_i) \wedge R(y_i, y_j)) \\
&\qquad \vee \bigwedge_{1 \leqslant i,j \leqslant k} (C_2(y_i) \wedge R(x, y_i) \wedge R(x, y_i) \wedge R(y_i, y_j))
\end{aligned}$$

Clearly, $(\mathcal{O}, \mathcal{D}_G) \models q_G(a_0)$ iff all 2-colorings of G admit a monochromatic k-clique. ⊣

2.3.11. *Complexity for $\mathcal{ALCQ}$*

Following the same arguments as for other languages, to determine the complexity of weak projective $(\mathcal{ALCQ}, \mathcal{ALCQ})$-separability, it is sufficient to determine the complexity of locally injective reachable unary rooted UCQ evaluation. In contrast to the $\mathcal{ALCQI}$ case, local injectivity here only concerns role names, but not inverse roles. This has no impact on the upper bound part, which can be proved in a similar fashion to the $\mathcal{ALC}$ one (Prop. 2.53). Because of number restrictions, it is however not as simple to extend the initial part as "plugging" models of the guessed $\mathcal{O}$-types onto each element. One needs to add exactly the "missing" number of each $\mathcal{O}$-type, similarly to what is done in subsection 2.3.3.

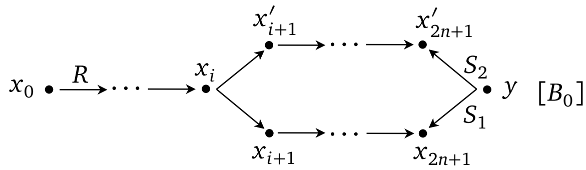
2.67. Lemma. *Locally injective reachable unary rooted UCQ evaluation on $\mathcal{ALCQ}$-knowledge bases is* coNEXP*-complete in combined complexity.*

Proof. For the lower bound, it suffices to slightly tweak the proof for the $\mathcal{ALC}$ lower bound (Prop. 2.53). The difference is that the CQ q branches directly at the 'root' while the corresponding 'real branching' in the tree might occur on a deeper level. That is not an issue in $\mathcal{ALC}$ as the homomorphism does not need to be locally injective. In the $\mathcal{ALCQ}$ case, the $(\Leftarrow)$ direction in Lemma 2.50 cannot be transposed: τ can have no solution given c while there might exist a model of $(\mathcal{O}, \{A_0(a)\})$ into which q, as defined there, can only be "reachably", but not locally injectively, embedded. A solution is to replace q with a UCQ $q_0 \vee \cdots \vee q_{2n-1}$ that contains one CQ for each possible level on which the branching could occur.

In detail, for all i, we define q_i (represented below) as follows.

$$\exists x_1 \ldots \exists x_{2n+1} \exists x'_{i+1} \ldots \exists x'_{2n+1} \exists y \begin{pmatrix} R(x_0, x_1) \wedge \cdots \wedge R(x_{2n}, x_{2n+1}) \wedge R(x_i, x'_{i+1}) \\ \wedge R(x'_{i+1}, x'_{i+2}) \wedge \cdots \wedge R(x_{2n}, x_{2n+1}) \\ \wedge S_1(y, x_{2n+1}) \wedge S_2(y, x'_{2n+1}) \wedge B_0(y) \end{pmatrix}$$

⊣



The lower bound follows.

2.68. Theorem. *Weak projective $(\mathcal{ALCQ}, \mathcal{ALCQ})$-separability is* NExp*-complete in combined complexity.*

§ 2.4. Non-projective case in $\mathcal{ALCI}$

Example 1.55, seen in the Introduction, shows that there exist labeled $\mathcal{ALCI}$-knowledge bases that are weakly projectively $\mathcal{ALCI}$-separable but not non-projectively so. That is also clearly true for all $\mathcal{L} \in \mathbf{DL}_{\mathcal{IOQ}}$, but we choose to start the investigation into non-projective separability with $\mathcal{ALCI}$, as it has the simplest characterisation in the projective case. Tweaking the model-theoretic characterisation of projective separability, we can establish a characterisation of non-projective $(\mathcal{ALCI}, \mathcal{ALCI})$-separability (Thm. 2.74). It follows from that characterisation that we can easily reduce the projective problem to the non-projective one in polynomial time (Cor. 2.78). We then deduce that weak non-projective $(\mathcal{ALCI}, \mathcal{ALCI})$-separability is, as the projective one, NExp-complete in combined complexity (Thm. 2.80) and in data complexity.

2.4.1. *Semantic characterisation*

Analogs of the bisimulation and simulation-based characterisations from the projective case are easily obtained in the non-projective case. They also, as in the projective case, pave the way for a (partially) "homomorphism-based" final characterisation.

2.69. Lemma. *Let $(\mathcal{K}, E^+, \{b\})$ be a labeled $\mathcal{ALCI}$-knowledge base. The following conditions are equivalent.*

1. $(\mathcal{K}, E^+, \{b\})$ *is weakly non-projectively* $\mathcal{ALCI}$*-separable.*
2. *There exists an* $\mathcal{ALCI}$*-forest model of finite outdegree* $\mathfrak{A} \models \mathcal{K}$ *such that* $(\mathfrak{B}, a^{\mathfrak{B}}) \not\sim_{\mathcal{ALCI},\mathsf{sig}(\mathcal{K})} (\mathfrak{A}, b^{\mathfrak{A}})$ *for all* $\mathfrak{B} \models \mathcal{K}, a \in E^+$.
3. *There exists an* $\mathcal{ALCI}$*-forest model of finite outdegree* $\mathfrak{A} \models \mathcal{K}$ *such that* $(\mathcal{D}, a) \not\preceq_{\mathcal{ALCI},\mathsf{sig}(\mathcal{K})} (\mathfrak{A}, b^{\mathfrak{A}})$ *for all* $a \in E^+$.

Proof. A simple transposition of the proofs of Theorem 2.16 (for (1) $\Longleftrightarrow$ (2)) and Lemma 2.18 (for (2) $\Longleftrightarrow$ (3)) from (bi)simulations respecting the full signature to (bi)simulations respecting $\mathsf{sig}(\mathcal{K})$ only. ⊣

We now introduce the notion of *incompleteness of a* $\mathsf{cl}(\mathcal{K})$*-type*, essential to formulate the characterisation of non-projective separability.

2.70. *Definition.* Let $\mathcal{K}$ be an $\mathcal{ALCI}$-knowledge base. Let $\mathsf{cl}(\mathcal{K})$ denote the closure under single negation of $\mathsf{sub}(\mathcal{K})$ and $\{\exists R.\top, \exists R^-.\top : R \in \mathsf{rel}_2 \cap \mathsf{sig}(\mathcal{K})\}$. We call $\mathsf{cl}(\mathcal{K})$*-type* any set $t \subseteq \mathsf{cl}(\mathcal{K})$ such that there exists a model $\mathfrak{A} \models \mathcal{K}$ and $a \in \mathsf{dom}(\mathfrak{A})$ with $\mathsf{tp}^{cl}_{\mathcal{K}}(\mathfrak{A}, a) = t$, where $\mathsf{tp}^{cl}_{\mathcal{K}}(\mathfrak{A}, a) = \{C \in \mathsf{cl}(\mathcal{K}) \mid a \in C^{\mathfrak{A}}\}$ is the $\mathsf{cl}(\mathcal{K})$*-type of* a *in* $\mathfrak{A}$, *i.e.* a *realizes* t *in* $\mathfrak{A}$. A $\mathsf{cl}(\mathcal{K})$-type t is *connected* if $\exists R.\top \in t$ for some $R \in \mathsf{rel}_2^-$. A $\mathsf{cl}(\mathcal{K})$-type t is $\mathcal{ALCI}$*-complete* if for any two pointed models $(\mathfrak{A}_1, b_1)$ and $(\mathfrak{A}_2, b_2)$ of $\mathcal{K}$, $t = \mathsf{tp}^{cl}_{\mathcal{K}}(\mathfrak{A}_1, b_1) = \mathsf{tp}^{cl}_{\mathcal{K}}(\mathfrak{A}_2, b_2)$ implies $(\mathfrak{A}_1, b_1) \sim_{\mathcal{ALCI},\mathsf{sig}(\mathcal{K})} (\mathfrak{A}_2, b_2)$. A type t is *realizable in* $(\mathcal{K}, b)$, where $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ and $b \in \mathsf{cons}(\mathcal{D})$, if $b^{\mathfrak{A}}$ realizes t in some model $\mathfrak{A}$ of $\mathcal{K}$. Two $\mathsf{cl}(\mathcal{K})$-types t_1 and t_2 are R*-coherent* if there exists $\mathfrak{A} \models \mathcal{K}$ and $d_1, d_2 \in \mathsf{dom}(\mathfrak{A})$ respectively realizing t_1 and t_2, such that $(d_1, d_2) \in R^{\mathfrak{A}}$. We write $t_1 \rightsquigarrow_R t_2$ in this case.

2.71. *Definition.* We say a sequence $\sigma = t_0 R_0 \ldots R_n t_{n+1}$ of $\mathsf{cl}(\mathcal{K})$-types $t_0, \ldots, t_{n+1}$ and $\mathsf{sig}(\mathcal{K})$-roles $R_0, \ldots, R_n$ *witnesses* $\mathcal{ALCI}$*-incompleteness of a* $\mathsf{cl}(\mathcal{K})$*-type* t if $t = t_0$, $n \geqslant 1$, and

1. $t_i \rightsquigarrow_{R_{i+1}} t_{i+1}$ for $i \leqslant n$;
2. there exists a model $\mathfrak{A}$ of $\mathcal{K}$ and nodes $d_{n-1}, d_n \in \mathsf{dom}(\mathfrak{A})$ with $(d_{n-1}, d_n) \in R^{\mathfrak{A}}_{n-1}$ such that d_{n-1} and d_n realize t_{n-1} and t_n in $\mathfrak{A}$, respectively, and there does not exist d_{n+1} in $\mathfrak{A}$ realizing t_{n+1} with $(d_n, d_{n+1}) \in R^{\mathfrak{A}}_n$.

The following lemma can be considered as part of Lemma 2.73, which describes how incompleteness will be used in Theorem 2.74.

2.72. Lemma. *There exists a tree-shaped model* $\mathfrak{A}_t$ *of* $\mathcal{O}$ *whose root* c *realizes* t *such that if a node* $e \in \mathsf{dom}(\mathfrak{A}_t)$ *realizes any* $\mathsf{cl}(\mathcal{K})$*-type* t_1 *and is of depth* $k \geqslant 0$*, then for every* $\mathsf{cl}(\mathcal{K})$*-type* t_2 *with* $t_1 \rightsquigarrow_R t_2$ *for some* $\mathsf{sig}(\mathcal{K})$*-role* R *there exists* e' *realizing* t_2 *of depth* $k+1$ *with* $(e, e') \in R^{\mathfrak{A}_t}$.

Proof. Define $\mathfrak{A}_t = \bigcup_{i \geqslant 0} \mathfrak{A}^i_t$ where $\mathsf{dom}(\mathfrak{A}^0_t) = \{t\}$. For all $i \geqslant 1$, role name R,

$t_0 \in \mathsf{dom}(\mathfrak{A}_t^i) \setminus \mathsf{dom}(\mathfrak{A}_t^{i-1})$ and t_1 such that $t_0 \rightsquigarrow_R t_1$, define $\mathfrak{A}_t^{i+1}$ by adding a fresh copy of t_1 as R-successor to t_0. For any concept name A, assume that $A^{\mathfrak{A}_t^i}$ consists of all elements such that their associated $\mathsf{cl}(\mathcal{K})$-type contains A. Then, $\mathfrak{A}_t \models \mathcal{O}$ follows from the fact that $C \in t_0$ iff $t_0 \in C^{\mathfrak{A}}$ for any $C \in \mathsf{cl}(\mathcal{K})$ where t_0 denotes any fresh copy of t_0 as a type. If C is atomic, that equivalence is clear by definition. If $C = \exists R.C'$ for some role R and concept C', suppose $C \in t_0$. Then, since t_0 is realizable by some x in some $\mathfrak{A}$ there exists x' such that $(x, x') \in R^{\mathfrak{A}}$ and $t \rightsquigarrow_R \mathsf{tp}_{\mathcal{K}}^{cl}(\mathfrak{A}, x)$ so there exists an R-successor of (any element representing) t_0 in $\mathfrak{A}_t$ that is a fresh copy of $\mathsf{tp}_{\mathcal{K}}^{cl}(\mathfrak{A}, x)$. By induction, $C' \in \mathsf{tp}_{\mathcal{K}}^{cl}(\mathfrak{A}, x)$ implies that this fresh copy is in $(C')^{\mathfrak{A}}$ which concludes the argument. The converse direction is straightforward. ⊣

2.73. Lemma. *The following conditions are equivalent for any* $\mathsf{cl}(\mathcal{K})$*-type* t.

1. *t is not $\mathcal{ALCI}$-complete;*
2. *There is a sequence witnessing $\mathcal{ALCI}$-incompleteness of t;*
3. *There is a sequence witnessing $\mathcal{ALCI}$-incompleteness of t, of length* $\leqslant 2^{||\mathcal{K}||}+1$.

Proof.

(1) ⇒ (2). If t is not $\mathcal{ALCI}$-complete, then there exists a model $\mathfrak{A}_t'$ of $\mathcal{O}$ realizing t in its root c' such that $(\mathfrak{A}_t, c) \not\sim_{\mathcal{ALCI},\mathsf{sig}(\mathcal{K})} (\mathfrak{A}_t', c')$. But then there is a sequence σ of the form $t_0 R_0 \ldots R_n t_{n+1}$ (possibly with $n = 0$) witnessing that absence of bisimulation that is realized in $\mathfrak{A}_t$ starting from c. We need $n \geqslant 1$ to fit Definition 2.71. To obtain a sequence σ with $n \geqslant 1$ assume that there exist a role R, a $\mathsf{cl}(\mathcal{K})$-type t' and a node $d \in \mathsf{dom}(\mathfrak{A}_t)$ such that $(c, d) \in R^{\mathfrak{A}_t}$ and d realizes t' in $\mathfrak{A}_t$, but there exists no such d' in $\mathfrak{A}_t'$ with $(c', d') \in R^{\mathfrak{A}_t'}$ and d' realizing t' in $\mathfrak{A}_t'$. If no such R, t', d exist then clearly already $n \geqslant 1$. Now observe that $\exists R.\top \in t$. Thus there exists d' realizing a $\mathsf{cl}(\mathcal{K})$-type t'' in $\mathfrak{A}'$ such that $(c', d') \in R^{\mathfrak{A}_t'}$. Then t and t'' are R-coherent, so by definition of $\mathfrak{A}_t$ there exists an R-successor of c in $\mathfrak{A}_t$ of type t''. Then the sequence $tRt''R^- tRt'$ is as required.

(2) ⇒ (3). Suppose there exists σ witnessing $\mathcal{ALCI}$-incompleteness of t with length $> 2^{||\mathcal{K}||}+1$. Define the sequence σ' obtained by substituting to exhaustion all subsequences in σ of the form $tR\sigma_1 R' t' R'' \sigma_2$ by $tR''\sigma_2$ whenever $t = t'$, where R, R', R'' are (possibly inverse) roles and σ_1, σ_2 are subsequences (with σ_2 being possibly empty). The total number of $\mathsf{cl}(\mathcal{K})$-types is bounded by $2^{||\mathcal{K}||}$. Then σ' has length $\leqslant 2^{||\mathcal{K}||} + 1$ since all its types are distinct, except possibly its first and last. Condition (1) in Definition 2.71 is trivially satisfied by σ' as it is by the larger sequence σ. Condition (2) is also satisfied by σ' as it is satisfied by σ and as the last two types in σ' are the same as in σ.

(3) ⇒ (1). Straightforward from the definition. ⊣

2.74. Theorem. *A labeled $\mathcal{ALCI}$-knowledge base $(\mathcal{K}, E^+, \{b\})$ is weakly non-projectively $\mathcal{ALCI}$-separable iff there exists a model $\mathfrak{A}$ of $\mathcal{K}$ such that, for all $a \in E^+$,*

1. $(\mathcal{D}_a, a) \not\to (\mathfrak{A}, b^{\mathfrak{A}})$
2. *if* $\mathsf{tp}^{cl}_{\mathcal{K}}(\mathfrak{A}, b^{\mathfrak{A}})$ *is connected and $\mathcal{ALCI}$-complete, it is not realizable in* $(\mathcal{K}, a)$.

Proof. ($\Rightarrow$) Assume $(\mathcal{K}, E^+, \{b\})$ is $\mathcal{ALCI}$-separable. By Lemma 2.69, there exists a forest model $\mathfrak{A}$ of $\mathcal{K}$ of finite outdegree such that $(\mathcal{D}_a, a) \not\preceq_{\mathcal{ALCI},\mathsf{sig}(\mathcal{K})} (\mathfrak{A}, b^{\mathfrak{A}})$ for all $a \in E^+$. To show that Condition 1 holds, assume that there exists $a \in E^+$ and a homomorphism h from $\mathcal{D}_a$ to $\mathfrak{A}$ mapping a to $b^{\mathfrak{A}}$. As h is clearly an $\mathcal{ALCI}(\mathsf{sig}(\mathcal{K}))$-embedding, we have derived a contradiction. To show that Condition 2 holds, assume that $\mathsf{tp}^{cl}_{\mathcal{K}}(\mathfrak{A}, b^{\mathfrak{A}})$ is $\mathcal{ALCI}$-complete and that $\mathsf{tp}^{cl}_{\mathcal{K}}(\mathfrak{A}, b^{\mathfrak{A}})$ is realized at $a^{\mathfrak{B}}$ in a model $\mathfrak{B}$ of $\mathcal{K}$. By completeness, $(\mathfrak{B}, a^{\mathfrak{B}}) \sim_{\mathcal{ALCI},\mathsf{sig}(\mathcal{K})} (\mathfrak{A}, b^{\mathfrak{A}})$.

($\Leftarrow$) Assume Conditions 1 and 2 hold for a model $\mathfrak{A}$ of $\mathcal{K}$. We may assume that $\mathfrak{A}$ is a forest model and of finite outdegree. If $\mathsf{tp}^{cl}_{\mathcal{K}}(\mathfrak{A}, b^{\mathfrak{A}})$ is connected and $\mathcal{ALCI}$-complete, then by Condition 2 $\neg(\bigsqcap_{C \in \mathsf{tp}^{cl}_{\mathcal{K}}(\mathfrak{A}, b^{\mathfrak{A}})} C)$ separates $(\mathcal{K}, E^+, \{b\})$ and we are done. If $\mathsf{tp}^{cl}_{\mathcal{K}}(\mathfrak{A}, b^{\mathfrak{A}})$ is not connected, then it follows from $(\mathcal{D}_a, a) \not\to (\mathfrak{A}, b^{\mathfrak{A}})$ that either there exists A with $A(a) \in \mathcal{D}$ and $b^{\mathfrak{A}} \notin A^{\mathfrak{A}}$ or there exists R with $R(a, c) \in \mathcal{D}$ for some c. In both cases $\mathsf{tp}^{cl}_{\mathcal{K}}(\mathfrak{A}, b^{\mathfrak{A}})$ is not realizable in $(\mathcal{K}, a)$. Thus $\neg(\bigsqcap_{C \in \mathsf{tp}^{cl}_{\mathcal{K}}(\mathfrak{A}, b^{\mathfrak{A}})} C)$ separates $(\mathcal{K}, E^+, \{b\})$ and we are done. Assume now that $\mathsf{tp}^{cl}_{\mathcal{K}}(\mathfrak{A}, b^{\mathfrak{A}})$ is connected and not $\mathcal{ALCI}$-complete. For a model $\mathfrak{C}$ of $\mathcal{K}$ and $\ell \geqslant 0$ we denote by $\mathfrak{C}^{\leqslant \ell}_{\mathcal{D},b}$ the submodel of $\mathfrak{C}$ induced by all nodes reachable from some $c^{\mathfrak{C}}$, $c \in \mathsf{dom}(\mathcal{D}_b)$, in at most ℓ steps. We construct for any $\ell \geqslant 0$ a model $\mathfrak{C}$ of $\mathcal{K}$ such that

1. $(\mathfrak{C}^{\leqslant \ell}_{\mathcal{D},b}, b^{\mathfrak{C}}) \to (\mathfrak{A}, b^{\mathfrak{A}})$;
2. $(\mathfrak{C}, d_1) \not\sim_{\mathcal{ALCI},\mathsf{sig}(\mathcal{K})} (\mathfrak{C}, d_2)$ for any two distinct $d_1, d_2 \in \mathsf{dom}(\mathfrak{C}^{\leqslant \ell}_{\mathcal{D},b})$.

By Lemma 2.75, the theorem is proved if such a $\mathfrak{C}$ can be constructed. By Lemma 2.76, such a model can indeed be constructed. ⊣

2.75. Lemma. *If (1) and (2) hold for $\ell \geqslant |\mathcal{D}|$ and $(\mathcal{D}_a, a) \not\to (\mathfrak{A}, b^{\mathfrak{A}})$ for $a \in E^+$, then $(\mathcal{D}_a, a) \not\preceq_{\mathsf{sig}(\mathcal{K})} (\mathfrak{C}, b^{\mathfrak{A}})$.*

Proof. Let $\ell \geqslant |\mathcal{D}|$. Assume that there exists an $\mathcal{ALCI}(\mathsf{sig}(\mathcal{K}))$-simulation S between $(\mathcal{D}_a, a)$ and $(\mathfrak{C}, b^{\mathfrak{C}})$ for some $a \in E^+$. As there is no homomorphism from $\mathcal{D}_a$ to $\mathfrak{A}$ mapping a to $b^{\mathfrak{A}}$, by Condition (1) there is no homomorphism from $\mathcal{D}_a$ to $\mathfrak{C}^{\leqslant \ell}_{\mathcal{D},b}$ mapping a to $b^{\mathfrak{C}}$. Then there exist e, d, d' with $d \neq d'$ and $(e, d), (e, d') \in S$ such that $\mathsf{dist}(b^{\mathfrak{C}}, d), \mathsf{dist}(b^{\mathfrak{C}}, d') \leqslant |\mathcal{D}_a|$ (otherwise S would be a homomorphism). Then $(\mathfrak{C}, d) \sim_{\mathcal{ALCI},\mathsf{sig}(\mathcal{K})} (\mathfrak{C}, d')$ and we have derived a contradiction to Condition (2) for $\mathfrak{C}$. ⊣

2.76. Lemma. *For any $\ell \geqslant 0$, there exists a model $\mathfrak{C}$ of $\mathcal{K}$ such that*

1. $(\mathfrak{C}^{\leqslant \ell}_{\mathcal{D},b}, b^{\mathfrak{C}}) \to (\mathfrak{A}, b^{\mathfrak{A}})$;
2. $(\mathfrak{C}, d_1) \not\sim_{\mathcal{ALCI},\mathsf{sig}(\mathcal{K})} (\mathfrak{C}, d_2)$ *for any two distinct* $d_1, d_2 \in \mathsf{dom}(\mathfrak{C}^{\leqslant \ell}_{\mathcal{D},b})$.

Proof. Take a sequence $\sigma = t^\sigma_0 R^\sigma_0 \dots R^\sigma_{m_\sigma} t^\sigma_{m_\sigma+1}$ that witnesses $\mathcal{ALCI}$-incompleteness of $t^\sigma_0 := \mathsf{tp}^{cl}_{\mathcal{K}}(\mathfrak{A}, b^{\mathfrak{A}})$, where $1 \leqslant m_\sigma \leqslant L_{\mathcal{K}} := 2^{||\mathcal{K}||} + 1$. Then there exists $d \in \mathsf{dom}(\mathfrak{A})$ such that $(b^{\mathfrak{A}}, d) \in (R^\sigma_0)^{\mathfrak{A}}$. By unfolding $\mathfrak{A}$ at all $c^{\mathfrak{A}}$, $c \in \mathsf{dom}(\mathcal{D}_b)$, we obtain a model of $\mathcal{K}$ having exactly the same properties as $\mathfrak{A}$ except that in addition in the tree-shaped models $\mathfrak{A}_c$ hooked to $c^{\mathfrak{A}}$ all nodes of any depth k have an R-successor in $\mathfrak{A}_c$ of depth $k+1$, for some $R \in \mathsf{sig}(\mathcal{K})$. We denote this model again by $\mathfrak{A}$. Denote by L the set of all nodes in $\mathfrak{A}$ that have depth exactly ℓ in some $\mathfrak{A}_c$, $c \in \mathsf{dom}(\mathcal{D}_b)$. We obtain $\mathfrak{C}$ by keeping only $\mathfrak{A}^{\leqslant \ell}_{\mathcal{D},b}$ and then attaching to every $d \in L$ a tree model $\mathfrak{F}_d$ (but not at its root) such that in the resulting model no node in L is $\mathcal{ALCI}(\mathsf{sig}(\mathcal{K}))$-bisimilar to any other node in $\mathfrak{A}^{\leqslant \ell}_{\mathcal{D},b}$. It then directly follows that $\mathfrak{C}$ satisfies Conditions (1) and (2):

1. $(\mathfrak{C}^{\leqslant \ell}_{\mathcal{D},b}, b^{\mathfrak{C}}) \to (\mathfrak{A}, b^{\mathfrak{A}})$ iff $(\mathfrak{A}^{\leqslant \ell}_{\mathcal{D},b}, b^{\mathfrak{C}}) \to (\mathfrak{A}, b^{\mathfrak{A}})$ which is trivially true.
2. Suppose for contradiction that some bisimulation S is witnessed by d_1, d_2. Then it is clear by 'following' the bisimulation that there exists $d \in L$ that contradicts the assumption.

We set

$$C_0 = \bigsqcap_{C \in t^\sigma_{m_\sigma - 1}} C, \quad C_1 = \bigsqcap_{C \in t^\sigma_{m_\sigma}} C, \quad C_2 = \bigsqcap_{C \in t^\sigma_{m_\sigma + 1}} C, \quad S = R^\sigma_{m_\sigma - 1}, \quad T = R^\sigma_{m_\sigma}$$

Take for any $d \in L$ a number N_d such that

$$\begin{cases} N_d > |\mathcal{D}| + 2\ell + 2(L_{\mathcal{K}} + 1) \\ |N_d - N_{d'}| > 2(L_{\mathcal{K}} + 1) \text{ for } d \neq d' \end{cases}$$

Now fix $d \in L$ and let $t_0 = \mathsf{tp}^{cl}_{\mathcal{K}}(\mathfrak{A}, d)$. By first walking from d to $b^{\mathfrak{A}}$ we find a sequence $t_0 R_0 \cdots R_{n_d} t_{n_d+1}$ that witnesses $\mathcal{ALCI}$-incompleteness of t_0 and ends with $t^\sigma_{m_\sigma-1} S t^\sigma_{m_\sigma} T t^\sigma_{m_\sigma+1}$: the concatenation of σ with the sequence of types leading from d to $b^{\mathfrak{A}}$. By Lemma 2.73 we may assume that $n_d \leqslant L_{\mathcal{K}}$. Let $D = \exists \Sigma^{L_{\mathcal{K}}}.(C_1 \sqcap \neg \exists T.C_2)$, where $\exists \Sigma^k.C$ stands for the disjunction of all $\exists \rho.C$ with ρ a path $R_1 \cdots R_m$ of $\Sigma := \mathsf{sig}(\mathcal{K})$, roles $R_1, \dots, R_m$ and $m \leqslant k$. To construct $\mathfrak{F}_d$, consider the tree model $\mathfrak{A}_{c_0}$ of $\mathcal{O}$ whose root c_0 realizes t_{n_d} such that if a node $e \in \mathsf{dom}(\mathfrak{A}_{c_0})$ realizes any $\mathsf{cl}(\mathcal{K})$-type t and is of depth $k \geqslant 0$, then for every $\mathsf{cl}(\mathcal{K})$-type t' with $t \rightsquigarrow_R t'$ for some $\mathsf{sig}(\mathcal{K})$-role R there exists e' realizing t' of depth $k+1$ with $(e, e') \in R^{\mathfrak{A}_{c_0}}$, except if $k \leqslant N_d + L_{\mathcal{K}} + 1$, $t = t_{n_d}$, $R = T$, and

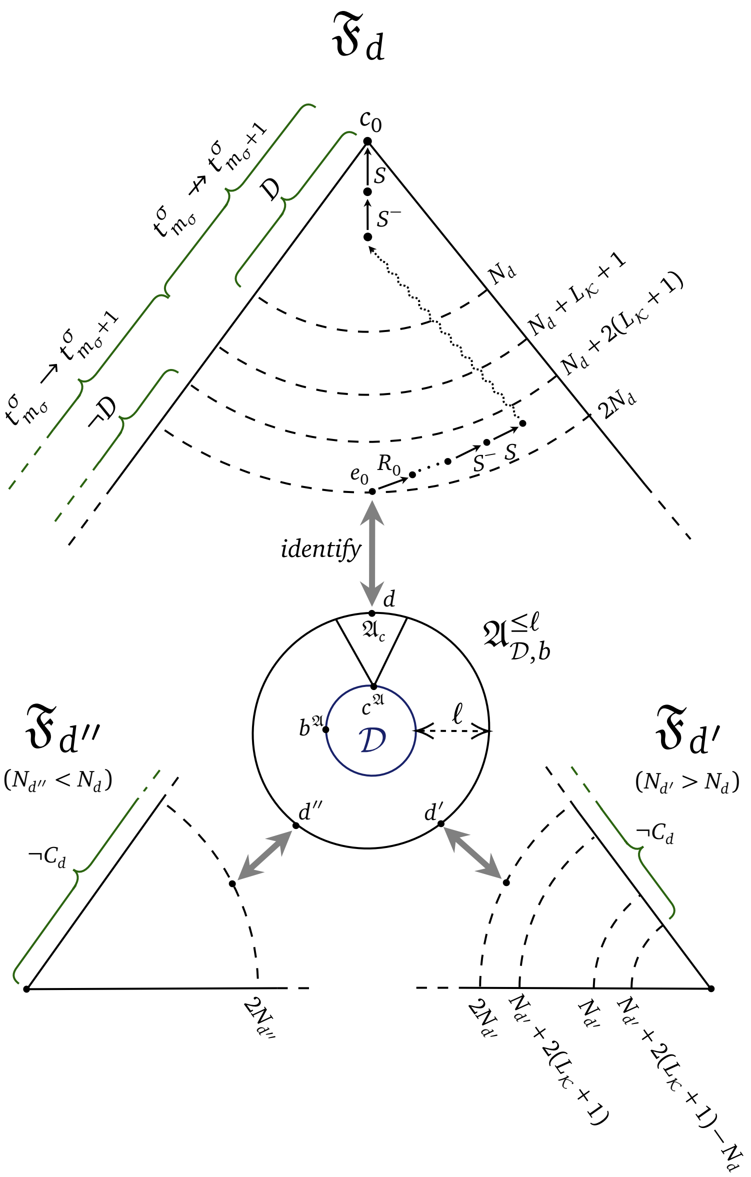
$t' = t_{n_d+1}$. Observe that $\mathfrak{A}_{c_0}$ satisfies

$$\begin{array}{ll} e \in D^{\mathfrak{A}_{c_0}} & \text{for all } e \text{ such that } \mathsf{dist}_{\mathfrak{A}_{c_0}}(c_0, e) \leqslant N_d; \\ e \notin D^{\mathfrak{A}_{c_0}} & \text{for all } e \text{ such that } \mathsf{dist}_{\mathfrak{A}_{c_0}}(c_0, e) > N_d + 2(L_{\mathcal{K}} + 1). \end{array}$$

Moreover, $\mathfrak{A}_{c_0}$ contains a path $e_0, \ldots, e_{n_d} \ldots, e_{n_d+2N_d} = c_0$ such that t_0 is realized in e_0, and

$$\begin{array}{ll} (e_i, e_{i+1}) \in R_i^{\mathfrak{A}_{c_0}} & \text{for all } i < n_d; \\ (e_{n_d+2k+1}, e_{n_d+2k}), (e_{n_d+2k+1}, e_{n_d+2k+2}) \in S^{\mathfrak{A}_{c_0}} & \text{for all } 0 \leqslant k < N_d; \\ e_{n_d+2k} \in C_1^{\mathfrak{A}_{c_0}} & \text{for all } k \leqslant N_d; \\ e_{n_d+2k+1} \in C_0^{\mathfrak{A}_{c_0}} & \text{for all } k < N_d. \end{array}$$

Then $\mathfrak{F}_d$ is obtained from $\mathfrak{A}_{c_0}$ by renaming e_0 to d. Finally $\mathfrak{C}$ is obtained from $\mathfrak{A}^{\leqslant \ell}_{\mathcal{D},b}$ by hooking $\mathfrak{F}_d$ at d to $\mathfrak{A}^{\leqslant \ell}_{\mathcal{D},b}$ for all $d \in L$ (see figure), via identifying e_0 with d. $\mathfrak{C}$ is a model of $\mathcal{K}$ since t_0 is realized in e_0. Moreover, clearly $\mathfrak{C}$ satisfies Condition (1). For Condition (2) assume $d \in L$ is as above. Let $C_d = \forall \Sigma^{N_d}.D$, where $\forall \Sigma^k.D$ stands for $\neg \exists \Sigma^k.\neg D$. As depicted below, condition (2) now follows from the fact that there exists a path from d to a node satisfying C_d that is shorter than any such path in $\mathfrak{C}$ from any other node in $\mathfrak{A}^{\leqslant \ell}_{\mathcal{D},b}$ to a node satisfying C_d. $\dashv$



An illustration of $\mathfrak{C}$, with example points d, d', d'' in $\mathsf{dom}(\mathfrak{A}^{\leq \ell}_{\mathcal{D},b})$ such that $N_{d''} < N_d < N_{d'}$.

2.77. *Remark.* Theorem 2.74 still holds without any mention of "connectedness". That mention simply highlights that if $\mathsf{tp}^{cl}_\mathcal{K}(\mathfrak{A}, b^\mathfrak{A})$ is complete and not connected, then Point 1 already suffices. Intuitively, if $\mathsf{tp}^{cl}_\mathcal{K}(\mathfrak{A}, b^\mathfrak{A})$ is not connected, then separability is determined only at the concept name level. Example 1.55 shows that when relations are involved (and more particularly cycles), the weak expressiveness of $\mathcal{ALCI}$ cannot match the discriminating power of homomorphisms. At the concept name level, homomorphisms lose that edge. Still by Example 1.55, we know that Point 1 is not enough if $\mathsf{tp}^{cl}_\mathcal{K}(\mathfrak{A}, b^\mathfrak{A})$ is complete: there is no homomorphism $(\mathcal{D}_a, a) \to (\mathfrak{A}, b^{\mathfrak{A}})$ but the labeled knowledge base is still not non-projectively separable. But Point 2 is also not enough if $\mathsf{tp}^{cl}_{\mathcal{K}}(\mathfrak{A}, b^{\mathfrak{A}})$ is not complete. Intuitively, completeness makes it so that the $\mathsf{cl}(\mathcal{K})$-type determines the bisimulation type, therefore the full $\mathcal{L}$-type. Then, whether it is realizable or not tells the full story. If $\mathsf{tp}^{cl}_{\mathcal{K}}(\mathfrak{A}, b^{\mathfrak{A}})$ is not complete however, there can be a separating concept, *i.e.* a difference in full logical types, that is not captured by $\mathsf{cl}(\mathcal{K})$-types. There are many more concepts using $\mathsf{sig}(\mathcal{K})$ than there are in $\mathsf{cl}(\mathcal{K})$, so it is easy to find such a counter-example. For instance, if $\mathcal{O} = \emptyset$ and $\mathcal{D} = \{R(a,b), R(b,c)\}$, then $\exists R.\exists R.\top$ is a separating concept but for any model $\mathfrak{A}$ of $\mathcal{K}$, $\mathsf{tp}^{cl}_{\mathcal{K}}(\mathfrak{A}, b^{\mathfrak{A}})$, which cannot contain any information about paths of length > 1, is still realizable in $(\mathcal{K}, a)$.

2.4.2. *Complexity*

Theorem 2.74 implies the following.

2.78. Corollary. *Weak projective $(\mathcal{ALCI}, \mathcal{ALCI})$-separability is polynomial-time reducible to weak non-projective $(\mathcal{ALCI}, \mathcal{ALCI})$-separability.*

Proof. Let $(\mathcal{K}, E^+, \{b\})$, $\mathcal{K} = (\mathcal{O}, \mathcal{D})$, be a labeled $\mathcal{ALCI}$-knowledge base. Then $\mathcal{K}$ is projectively $\mathcal{ALCI}$-separable iff $(\mathcal{K}', E^+, \{b\})$ is non-projectively $\mathcal{ALCI}$-separable where $\mathcal{K}' = (\mathcal{O}', \mathcal{D})$ and $\mathcal{O}' = \mathcal{O} \cup \{A \sqsubseteq A\}$, A a fresh concept name: $(\mathcal{K}, E^+, \{b\})$ is clearly projectively $\mathcal{ALCI}$-separable iff $(\mathcal{K}', E^+, \{b\})$ is, which is itself projectively $\mathcal{ALCI}$-separable iff it is non-projectively $\mathcal{ALCI}$-separable because no connected $\mathsf{cl}(\mathcal{K}')$-type is $\mathcal{ALCI}$-complete, thus Point 2 of Theorem 2.74 is vacuously true. ⊣

2.79. The argument above also implies that whenever a labeled $\mathcal{ALCI}$-knowledge base is projectively separable, then a single fresh concept name suffices for separation.

2.80. Theorem. *Weak non-projective $(\mathcal{ALCI}, \mathcal{ALCI})$-separability is* NExp*-complete in combined complexity and in data complexity.*

Proof. The lower bound is immediate from Corollary 2.78. For the upper bound, let $(\mathcal{K}, E^+, \{b\})$ be a labeled $\mathcal{ALCI}$-knowledge base. For any $\mathsf{cl}(\mathcal{K})$-type t, let $\mathcal{K}_t = (\mathcal{O}_t, \mathcal{D}_t)$ where $\mathcal{O}_t = \mathcal{O} \cup \{A \sqsubseteq \bigsqcap_{C \in t} C\}$ and $\mathcal{D}_t = \mathcal{D} \cup \{A(b)\}$ for a fresh concept name A. By Theorem 2.74, $(\mathcal{K}, E^+, \{b\})$ is $\mathcal{ALCI}$-separable iff there exists a $\mathsf{cl}(\mathcal{K})$-type t that is realizable in $(\mathcal{K}, b)$ such that (i) $\mathcal{K}_t \not\models \bigvee_{a \in E^+} q(\mathcal{D}_a, a)(b)$ and (ii) if t is connected and $\mathcal{ALCI}$-complete, then t is not realizable in $(\mathcal{K}, a)$ for any $a \in E^+$: if (i) is witnessed by $\mathfrak{A} \models \mathcal{K}_t$ then $\mathfrak{A} \models \mathcal{K}$ and that directly implies Condition (1) of Theorem 2.74. Condition (2) of Theorem 2.74 follows immediately from (ii) and the fact that $t \subseteq \mathsf{tp}^{cl}_{\mathcal{K}}(\mathfrak{A}, b^{\mathfrak{A}})$ implies $t = \mathsf{tp}^{cl}_{\mathcal{K}}(\mathfrak{A}, b^{\mathfrak{A}})$ by definition of $\mathsf{cl}(\mathcal{K})$-types. The NExp upper bound now follows from the following facts.

1. Rooted UCQ evaluation on $\mathcal{ALCI}$-knowledge bases is in coNExp (complement of (i)).

2. $\mathcal{ALCI}$-completeness of t can be checked in exponential time.

   Proof. There exists a model $\mathfrak{A}$ whose domain consists of all $\mathsf{cl}(\mathcal{K})$-types t and such that $t \in A^{\mathfrak{A}}$ if $A \in t$ and $(t_1, t_2) \in R^{\mathfrak{A}}$ if $t_1 \rightsquigarrow_R t_2$ (see 2.72). Then t is not $\mathcal{ALCI}$-complete iff there exists a path starting at t in $\mathfrak{A}$ that ends with $R^{\mathfrak{A}}_{n-1} t_n R^{\mathfrak{A}}_n t_{n+1}$ such that the second condition for sequences witnessing $\mathcal{ALCI}$-incompleteness holds. The existence of such a path can be decided in polynomial time in the size of $\mathfrak{A}$, thus in exponential time.

3. Realizability of t can be checked in exponential time: t is realizable w.r.t. $(\mathcal{K}, a)$ if $\bigsqcap_{C \in t} C(a)$ is satisfiable w.r.t. $\mathcal{K}$, which is decidable in exponential time in $\mathcal{ALCI}$.

⊣

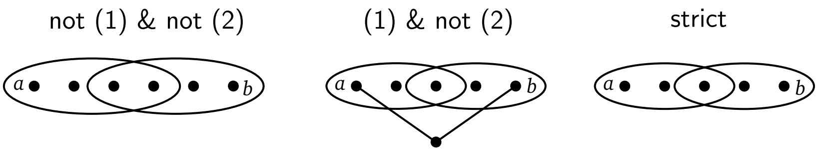
§ 2.5. GF-ONTOLOGIES

Projective and non-projective (**GF**, **GF**)-separability turn out to behave similarly to their $\mathcal{ALCI}$ counterparts in many ways. First, the projective and non-projective cases also do not coincide, as shown in the preliminaries by Example 1.55, because **GF** contains the first-order translation of $\mathcal{ALCI}$. Then, **GF** also admits a notion of bisimulation, which we use to characterise separability in a similar way to $\mathcal{ALCI}$: in the projective case we also characterise separability following a “bisimulation-simulation-homomorphism” pattern, while in the non-projective case we also rely on a notion of “type incompleteness”. An analogous connection with **UCQ**-evaluation then yields 2Exp-completeness of projective and non-projective full weak (**GF**, **GF**)-separability. Overall, the results are significantly more difficult to establish than in the $\mathcal{ALCI}$ case: **GF** deals with n-ary relations, which poses additional challenges. For instance, trees and unfoldings may not be used as easily. We define and work with the more elaborate *guarded tree decompositions* instead. *Guarded bisimulations*, defined as sets of partial isomorphisms, are also more involved than simple bisimulations.

2.5.1. *Preliminaries*

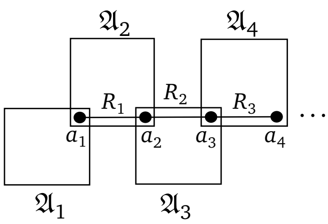
We introduce necessary definitions and results. General **GF** definitions are available in Section 1.5. Recall that for any tuple $\mathbf{x} = (x_1, \ldots, x_n)$ we write $[\mathbf{x}]$ for $\{x_1, \ldots, x_n\}$. We define paths and strict paths in the context of **GF**. It is not as straightforward as in $\mathcal{ALCI}$ to obtain a **GF** counterpart to Lemma 2.72. Using strict paths makes that construction possible.

2.81. *Definition.* A *path* of length n from a to b in a model $\mathfrak{A}$ is (in this section) a path of length n from a to b in the Gaifman graph associated with $\mathfrak{A}$ (Def. 1.11), *i.e.* a sequence $R_1(\mathbf{b}_1)\ldots R_n(\mathbf{b}_n)$ where $R_1, \ldots, R_n \in \mathsf{rel} \setminus \mathsf{rel}_1$, $a \in [\mathbf{b_1}]$, $b \in [\mathbf{b}_n]$, and $[\mathbf{b}_i] \cap [\mathbf{b}_{i+1}] \neq \emptyset$ for all i. We call a path *strict* if **(1)** all $[\mathbf{b}_i] \cap [\mathbf{b}_{i+1}]$ are singletons consisting of distinct points c_i and **(2)** there are sets $A_1, \ldots, A_n \subseteq \mathsf{dom}(\mathfrak{A})$ covering $\mathsf{dom}(\mathfrak{A})$ such that $[\mathbf{b}_i] \subseteq A_i$, $A_i \cap A_{i+1} = \{c_i\}$ and such that if $i < j$, then any path in the Gaifman graph of $\mathfrak{A}$ from an element of A_i to an element of A_j contains c_k for all $k \in \{i, \ldots, j-1\}$.



The following construction allows us to transform paths into strict paths, while preserving the bisimulation type.

2.82. *Definition.* The *partial unfolding* $\mathfrak{A}_\mathbf{a}$ of a model $\mathfrak{A}$ along a tuple $\mathbf{a} = (a_1, \ldots, a_n)$ in $\mathsf{dom}(\mathfrak{A})$ such that $\mathsf{dist}_\mathfrak{A}(a_i, a_{i+1}) = 1$ for all $i < n$ is defined as the following union of $n+1$ copies of $\mathfrak{A}$. Denote the copies by $\mathfrak{A}_1, \mathfrak{A}_2, \ldots, \mathfrak{A}_{n+1}$. The copies are mutually disjoint except that $\mathfrak{A}_i$ and $\mathfrak{A}_{i+1}$ share a copy of a_i. The domain of $\mathfrak{A}_i$ is $\mathfrak{A} \times \{i\}$ with (a_{i-1}, i) and $(a_{i-1}, i-1)$ identified, for all $i > 1$. The constants are interpreted in $\mathfrak{A}_1$ and we often denote the elements $(a, 1)$ of $\mathfrak{A}_1$ simply by a. The following figure illustrates the construction.



2.83. It is straightforward to check that if $i < j$, then any path in $\mathfrak{A}_\mathbf{a}$ from an element of $\mathsf{dom}(\mathfrak{A}_i)$ to an element of $\mathsf{dom}(\mathfrak{A}_j)$ contains (a_k, k) for all $k \in \{i, \ldots, j-1\}$, which makes $R_1, \ldots, R_n$ a strict path.

We then show that partial unfoldings preserve the bisimulation type.

2.84. Lemma. *Let I contain for all i with $1 \leqslant i \leqslant n+1$ and all guarded $(b_1, \dots, b_k)$ in $\mathfrak{A}$ the mappings $p : (b_1, \dots, b_k) \mapsto (c_1, \dots, c_k)$, where $c_j = (b_j, i)$ if $b_j \neq a_{i-1}$ and $c_j = (b_j, i-1)$ if $b_j = a_{i-1}$. Then I is a guarded bisimulation between $\mathfrak{A}$ and $\mathfrak{A}_{\mathbf{a}}$.*

Proof. For any i, every such mapping $(b_1, \dots, b_k) \mapsto (c_1, \dots, c_k)$ from $\mathfrak{A}$ to $\mathfrak{A}_{\mathbf{a}}$ (in particular to $\mathfrak{A}_i$) is a partial isomorphism by definition of $\mathfrak{A}_{\mathbf{a}}$. We then need to check the (Forth) and (Back) conditions. Let $(b_1, \dots, b_k) \mapsto (c_1, \dots, c_k) \in I$. Write $\mathbf{b} = (b_1, \dots, b_k)$.

(Forth) Let $\mathbf{b}' = (b'_1, \dots, b'_{k'})$ be a guarded tuple in $\mathfrak{A}$. Then $(b'_1, \dots, b'_{k'}) \mapsto (c'_1, \dots, c'_{k'})$, where c'_j is defined w.r.t. b'_j as c_j is defined w.r.t. b_j, obviously coincides with $(b_1, \dots, b_k) \mapsto ((b_1, i), \dots, (b_n, i))$ on $[\mathbf{b}] \cap [\mathbf{b}']$.

(Back) Let $(c'_1, \dots, c'_{k'})$ be a guarded tuple in $\mathfrak{A}_{\mathbf{a}}$ that intersects $(c_1, \dots, c_k)$. By definition, $c_1, \dots, c_k \in \mathsf{dom}(\mathfrak{A}_i)$ for some i. Thus, by definition of $\mathfrak{A}_{\mathbf{a}}$ and the fact that $(c'_1, \dots, c'_{k'})$ is guarded, we know that $c'_1, \dots, c'_{k'} \in \mathfrak{A}_\ell$ for some $\ell \in \{i-1, i, i+1\}$, assuming without loss of generality that $n > i > 1$ (cases n and 1 are similar). For any of those three values of ℓ, the mapping sending c'_j to its first coordinate coincides with $(c_1, \dots, c_k) \mapsto (b_1, \dots, b_k)$ on $[\mathbf{c}] \cap [\mathbf{c}']$. ⊣

2.85. As an immediate consequence of the previous lemma, If $\mathfrak{A} \models \mathcal{K}$, then $\mathfrak{A}_{\mathbf{a}} \models \mathcal{K}$. Also, the mapping h from $\mathfrak{A}_{\mathbf{a}}$ to $\mathfrak{A}$ defined by setting $h(b, i) = b$ is a homomorphism from $\mathfrak{A}_{\mathbf{a}}$ to $\mathfrak{A}$: by definition, $\mathfrak{A}_{\mathbf{a}} \models R((b_1, i_1), \dots, (b_m, i_m))$ implies $\mathfrak{A} \models R(b_1, \dots, b_m)$, for any tuple $((b_1, i_1), \dots, (b_m, i_m))$ in $\mathfrak{A}_{\mathbf{a}}$.

2.86. Assume that $R_0(\mathbf{a}_0), \dots, R_n(\mathbf{a}_n)$ is a path in $\mathfrak{A}$ with $a_{i+1} \in [\mathbf{a}_i] \cap [\mathbf{a}_{i+1}]$ for $i \leqslant n$. Let $\mathbf{a}_i = (a_i^1, \dots, a_i^{n_i})$ and assume $a_i^1 = a_{i+1}$. Then $R_0(\mathbf{a}_0, 1), \dots, R_n(\mathbf{a}_n, n+1)$ is a strict path in $\mathfrak{A}_{\mathbf{a}}$ realizing the same $\mathcal{K}$-types as the original path, where

$$\begin{aligned} (\mathbf{a}_0, 1) &:= ((a_0^1, 1), \dots, (a_0^{n_0}, 1)) \\ (\mathbf{a}_i, i+1) &:= ((a_i^1, i), (a_i^2, i+1) \dots, (a_i^{n_i}, i+1)) \end{aligned}$$

We introduce guarded embeddings as a tool to prove Theorem 2.90. They act as the **GF** counterpart to DL simulations (Def. 2.17).

2.87. *Definition.* Let $(\mathcal{D}, \mathbf{a})$ be a pointed database, $(\mathfrak{A}, \mathbf{b}^{\mathfrak{A}})$ a pointed model, $\ell \geqslant 0$, and $\Sigma \supseteq \mathsf{sig}(\mathcal{K})$ a signature. A *partial embedding* is an injective partial homomorphism. A pair (e, H) is a *guarded Σ ℓ-embedding between $(\mathcal{D}, \mathbf{a})$ and $(\mathfrak{A}, \mathbf{b})$* if e is a homomorphism from $\mathcal{D}$ onto a database $\mathcal{D}'$ and H is a set of partial embeddings from $\mathcal{D}'$ to $\mathfrak{A}$ containing $h_0 : e(\mathbf{a}) \mapsto \mathbf{b}$ and a partial embedding

h from any guarded set in $\mathcal{D}'$ to $\mathfrak{A}$ such that the following condition holds. If $h_i : \mathbf{a}_i \mapsto \mathbf{b}_i \in H$ for $i = 1, 2$, then there exists a partial isomorphism $p : h_1([\mathbf{a}_1] \cap [\mathbf{a}_2]) \mapsto h_2([\mathbf{a}_1] \cap [\mathbf{a}_2])$ such that $p \circ h_1$ and h_2 coincide on $[\mathbf{a}_1] \cap [\mathbf{a}_2]$ and for any $\mathbf{c}$ with $[\mathbf{c}] = h_1([\mathbf{a}_1] \cap [\mathbf{a}_2])$, $(\mathfrak{A}, \mathbf{c}) \sim^{\ell}_{\mathbf{oGF},\Sigma} (\mathfrak{A}, p(\mathbf{c}))$. We write $(\mathcal{D}, \mathbf{a}) \preceq^{\ell}_{\mathbf{oGF},\Sigma} (\mathfrak{A}, \mathbf{b}^{\mathfrak{A}})$ if there exists a guarded Σ ℓ-embedding H between $(\mathcal{D}, \mathbf{a})$ and $(\mathfrak{A}, \mathbf{b})$.

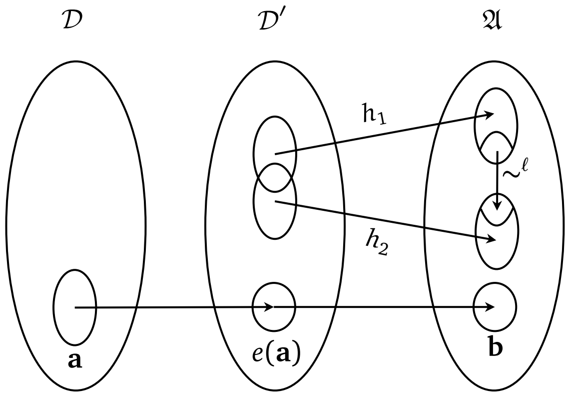


Illustration for Definition 2.87.

The following is a useful property of guarded embeddings that will be involved in later proofs.

2.88. Lemma. *Let $(\mathcal{D}, \mathbf{a})$ be a pointed database and $(\mathfrak{A}, \mathbf{b}^{\mathfrak{A}})$ be a pointed model such that $(\mathcal{D}, \mathbf{a}) \preceq^{\ell}_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} (\mathfrak{A}, \mathbf{b}^{\mathfrak{A}})$. Then there there exist a surjective homomorphism $e : \mathcal{D} \to \mathcal{D}'$ for some database $\mathcal{D}'$ and sets $H_\ell, \dots, H_0$ of partial embeddings $\mathcal{D}' \to \mathfrak{A}$ such that*

1. *for all $k \leqslant \ell$, all $h \in H_k$ and all guarded sets $\mathbf{c}$ in $\mathcal{D}'$ such that $[\mathbf{c}] \cap \mathsf{dom}(h) \neq \emptyset$, there exists $h' \in H_{k-1}$ with domain $[\mathbf{c}]$ such that h' coincides with h on $[\mathbf{c}] \cap \mathsf{dom}(h)$,*
2. *for all $k_1, k_2 \leqslant \ell$, all $h_1 \in H_{k_1}, h_2 \in H_{k_2}$, and all tuples $\mathbf{c}_1, \mathbf{c}_2$ in $\mathcal{D}'$ such that $[\mathbf{c}_i] = \mathsf{dom}(h_i)$, for all $\mathbf{c}$ such that $[\mathbf{c}] = [\mathbf{c}_1] \cap [\mathbf{c}_2]$, we have*

$$(\mathfrak{A}, h_1(\mathbf{c})) \sim^{\min(k_1,k_2)}_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} (\mathfrak{A}, h_2(\mathbf{c})).$$

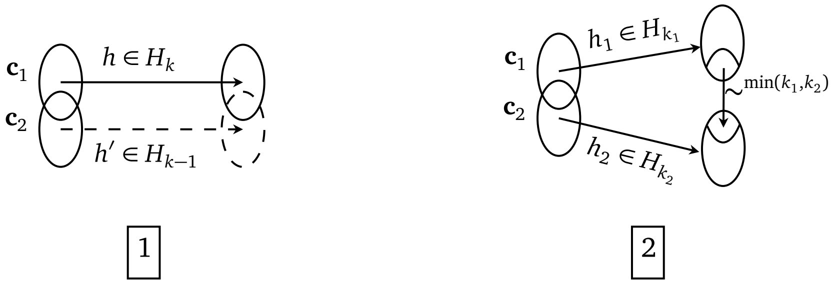
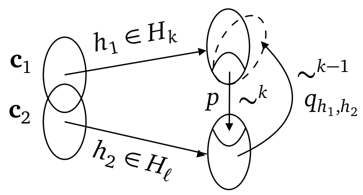


Illustration for conditions (1) and (2) above.

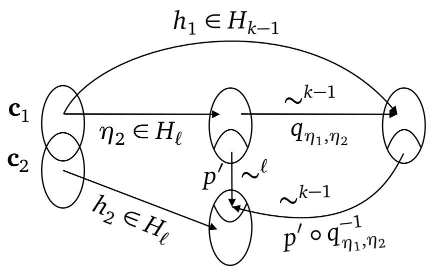
Proof. Let H be the set of partial embeddings witnessing $(\mathcal{D}, \mathbf{a}) \preceq^{\ell}_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} (\mathfrak{A}, \mathbf{b}^{\mathfrak{A}})$. Define $H_\ell := H$. We define H_k for $k < \ell$ by induction. Suppose H_k has been defined. We define H_{k-1}. We assume that for all $h_1 \in H_k, h_2 \in H_\ell$ having intersecting domains $[\mathbf{c}_1], [\mathbf{c}_2]$, with $\mathbf{c}_2$ being guarded the following condition holds:

(*) for any tuple $\mathbf{c}$ in $\mathcal{D}'$ such that $[\mathbf{c}] = [\mathbf{c}_1] \cap [\mathbf{c}_2]$, there is a partial isomorphism $p : h_1(\mathbf{c}) \mapsto h_2(\mathbf{c})$ witnessing $h_1(\mathbf{c}) \sim^{k}_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} h_2(\mathbf{c})$.

Now assume that h_1, h_2 satisfying the conditions above are given. As $[h_2(\mathbf{c}_2)]$ is guarded (by $\mathbf{c}_2$ being guarded and h a partial homomorphism) and intersects $h_2[[\mathbf{c}_1] \cap [\mathbf{c}_2]]$, and as p witnesses an $\mathbf{oGF}(\mathsf{sig}(\mathcal{K}))$ k-bisimulation, there exists a partial isomorphism q_{h_1,h_2} with domain $[h_2(\mathbf{c}_2)]$ witnessing $h_2(\mathbf{c}_2) \sim^{k-1}_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} q_{h_1,h_2}(h_2(\mathbf{c}_2))$ and that coincides with p^{-1} on $h_2[[\mathbf{c}_1] \cap [\mathbf{c}_2]]$. We then include $q_{h_1,h_2} \circ h_2$ in H_{k-1}.



This is well-defined, as the assumption (*) holds for all $k \leqslant \ell$. Indeed, iff $k = \ell$, then (*) is stated in the definition of $\mathsf{sig}(\mathcal{K})$ ℓ-guarded embeddings. If $0 < k < \ell$ and (*) holds for k, let $h_1 \in H_{k-1}, h_2 \in H_\ell$ with intersecting domains $[\mathbf{c}_1], [\mathbf{c}_2]$ and $\mathbf{c}_2$ guarded be given. Then $h_1 = q_{\eta_1,\eta_2} \circ \eta_2$ for some $\eta_1 \in H_k, \eta_2 \in H_\ell$, by definition of H_{k-1}. By definition of $\mathsf{sig}(\mathcal{K})$ ℓ-guarded embeddings, as η_2 and h_2 are both in H_ℓ and have intersecting domains $[\mathbf{c}_1]$ and $[\mathbf{c}_2]$, there exists a partial isomorphism p' witnessing $\eta_2(\mathbf{c}) \sim^{\ell}_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} h_2(\mathbf{c})$ for any $\mathbf{c}$ such that $[\mathbf{c}] = [\mathbf{c}_1] \cap [\mathbf{c}_2]$. Then, by composition of bisimulations, $p := p'_{|\eta_2[\mathbf{c}]} \circ (q^{-1}_{\eta_1,\eta_2})_{|h_1[\mathbf{c}]}$ is a partial isomorphism witnessing that (*) holds for $k-1$.



Elements of H_{k-1} are partial embeddings, as compositions of partial isomorphisms with partial embeddings. We thus have a homomorphism $e : \mathcal{D} \to \mathcal{D}'$ and sets $H_\ell, \dots, H_0$ of partial embeddings $\mathcal{D}' \to \mathfrak{A}$. We now prove that Conditions 1 and 2 hold.

1. Let $0 \leqslant k \leqslant \ell$ and $h_1 \in H_k$ with domain $[\mathbf{c}_1]$. Let $\mathbf{c}_2$ be guarded in $\mathcal{D}'$ such that $[\mathbf{c}_1] \cap [\mathbf{c}_2] \neq \emptyset$. By definition of ℓ-guarded embeddings, every guarded tuple is the domain of some embedding in $H = H_\ell$. In particular there exists $h_2 \in H_\ell$ with domain $[\mathbf{c}_2]$. Then Condition (*) holds, with matching notation. Consider q_{h_1,h_2} and p as defined above. A witnessing partial homomorphism h' can be defined as $h' := q_{h_1,h_2} \circ h_2 \in H_{k-1}$. Since $p^{-1} \circ h_2$ coincides with h_1 on $[\mathbf{c}_1] \cap [\mathbf{c}_2]$, and q_{h_1,h_2} coincides with p^{-1} on $h_2[[\mathbf{c}_1] \cap [\mathbf{c}_2]]$, it follows that h' coincides with h_1 on $[\mathbf{c}_1] \cap [\mathbf{c}_2]$.

2. Let $h_1 \in H_{k_1}, h_2 \in H_{k_2}$ with intersecting domains $[\mathbf{c}_1], [\mathbf{c}_2]$. By definition of ℓ-guarded embeddings, there exists h_2' in $H = H_\ell$ with domain $[\mathbf{c}_2]$. By (*), for every $\mathbf{c}$ in $\mathcal{D}'$ such that $[\mathbf{c}] = [\mathbf{c}_1] \cap [\mathbf{c}_2]$ we have $h_1(\mathbf{c}) \sim^{k_1}_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} h_2'(\mathbf{c})$ and $h_2(\mathbf{c}) \sim^{k_2}_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} h_2'(\mathbf{c})$, thus $h_1(\mathbf{c}) \sim^{\min(k_1,k_2)}_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} h_2(\mathbf{c})$ by composition of bisimulations. ⊣

2.89. If $H_\ell, \dots, H_0$ satisfying the conditions of Lemma 2.88 exist, then $H_\ell \subseteq \dots \subseteq H_0$: let $k \leqslant \ell$ and $\mathbf{c} \mapsto \mathbf{d} \in H_k$. By condition (1), since $[\mathbf{c}] \cap [\mathbf{c}] \neq \emptyset$ there exists $\mathbf{c} \mapsto \mathbf{d}' \in H_{k-1}$ that coincides with $\mathbf{c} \mapsto \mathbf{d}$ on $[\mathbf{c}]$, *i.e.* $\mathbf{c} \mapsto \mathbf{d} \in H_{k-1}$.

2.5.2. *Intermediate characterisation*

We can now state the model-theoretic characterisation of separability in terms of bisimulation and embeddings, as we did for $\mathcal{ALCI}$ in Lemma 2.69. The process to go from simulation/embedding to bisimulation is essentially the same as for $\mathcal{ALCI}$. As guarded bisimulations are defined with respect to n-ary relations, the proof is heavier in its execution but not conceptually.

2.90. Theorem. *Let $(\mathcal{K}, E^+, \{\mathbf{b}\})$ be a labeled* **GF***-knowledge base. The following conditions are equivalent.*

1. *$(\mathcal{K}, E^+, \{\mathbf{b}\})$ is non-projectively* **oGF***-separable.*
2. *$(\mathcal{K}, E^+, \{\mathbf{b}\})$ is non-projectively* **GF***-separable.*
3. *There exist $\mathfrak{A} \models \mathcal{K}$ (finite), $\ell \geqslant 0$ such that $(\mathfrak{B}, \mathbf{a}^{\mathfrak{B}}) \not\sim^{\ell}_{\mathbf{GF},\mathsf{sig}(\mathcal{K})} (\mathfrak{A}, \mathbf{b}^{\mathfrak{A}})$ for all $\mathfrak{B} \models \mathcal{K}, \mathbf{a} \in E^+$.*
4. *There exist $\mathfrak{A} \models \mathcal{K}$ (finite), $\ell \geqslant 0$ such that $(\mathcal{D}_{\mathbf{a}}, \mathbf{a}) \not\lesssim^{\ell}_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} (\mathfrak{A}, \mathbf{b}^{\mathfrak{A}})$ for all $\mathbf{a} \in E^+$.*
5. *There exist $\mathfrak{A} \models \mathcal{K}$ (finite), $\ell \geqslant 0$ such that $(\mathfrak{B}, \mathbf{a}^{\mathfrak{B}}) \not\sim^{\ell}_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} (\mathfrak{A}, \mathbf{b}^{\mathfrak{A}})$ for all $\mathfrak{B} \models \mathcal{K}, \mathbf{a} \in E^+$.*

Proof. The implications (1) ⇒ (2), (2) ⇒ (3) and (5) ⇒ (1) are straightforward. For an indirect proof of (4) ⇒ (5), suppose $I_{2\ell},\dots,I_0$ is a guarded $\mathsf{sig}(\mathcal{K})$ 2ℓ-bisimulation between $(\mathfrak{B},\mathbf{a}^{\mathfrak{B}})$ and $(\mathfrak{A},\mathbf{b}^{\mathfrak{A}})$ for a model $\mathfrak{B}$ of $\mathcal{K}$, where $\ell \geqslant |\mathcal{D}|$. We may assume that $I_{i+1} \subseteq I_i$ for $i < 2\ell$. Let $\mathcal{D}'$ be the restriction of $\mathfrak{B}$ to $\{c^{\mathfrak{B}} \mid c \in \mathsf{cons}(\mathcal{D}_{\mathbf{a}})\}$, seen as a database. Define $e : \mathcal{D}_{\mathbf{a}} \to \mathcal{D}'$ by setting $e(c) = c^{\mathfrak{B}}$. Let H contain $h_0 : \mathbf{a}^{\mathfrak{B}} \mapsto \mathbf{b}^{\mathfrak{A}}$ and, for every guarded tuple $\mathbf{d}$ in $\mathcal{D}'$ any $h : \mathbf{d} \mapsto \mathbf{c} \in I_\ell$. Then (e,H) is a guarded $\mathsf{sig}(\mathcal{K})$ ℓ-embedding: assume that $h_i : \mathbf{c}_i \mapsto \mathbf{d}_i \in H$ for $i = 1,2$. Let X_1,X_2 be the images of $[\mathbf{c}_1] \cap [\mathbf{c}_2]$ under h_i and $\mathbf{d}$ such that $[\mathbf{d}] = X_1$. Then we have $h_i : \mathbf{c}_i \mapsto \mathbf{d}_i \in I_\ell$, for $i = 1,2$. Let p be the restriction of $h_2 \circ h_1^{-1}$ to X_1. By definition p is a partial isomorphism from X_1 to X_2. It is as required, as $(\mathfrak{A},\mathbf{d}) \sim^{\ell}_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} (\mathfrak{B},h_1^{-1}(\mathbf{d})) \sim^{\ell}_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} (\mathfrak{A},h_2(h^{-1}(\mathbf{d})))$.

(3) ⇒ (4). Take a model $\mathfrak{A}$ of $\mathcal{K}$ and $\ell \geqslant 0$ witnessing Condition 3. We may assume that ℓ exceeds the maximum guarded quantifier rank of formulas in $\mathcal{K}$. We show that Condition 4 holds for $\mathfrak{A}$ and ℓ. Assume for a proof by contradiction that there exists $\mathbf{a}_0 \in E^+$ such that there exists a guarded $\mathsf{sig}(\mathcal{K})$ ℓ-embedding (e,H) from $(\mathcal{D}_{\mathbf{a}_0},\mathbf{a}_0)$ to $(\mathfrak{A},\mathbf{b}^{\mathfrak{A}})$. Assume $e : \mathcal{D}_{\mathbf{a}_0} \mapsto \mathcal{D}'$ and $e(\mathbf{a}_0) = \mathbf{a}_0'$. We construct a model $\mathfrak{B}$ as follows. First take a copy $\mathfrak{B}'$ of $\mathfrak{A}$. We define $\mathfrak{B}$ as the disjoint union of $\mathfrak{B}'$ and $\mathfrak{B}''$, where $\mathfrak{B}''$ is defined next. We denote by H' the set obtained from H with $\mathbf{a}_0' \mapsto \mathbf{b}^{\mathfrak{A}}$ removed if $\mathbf{a}_0$ is not guarded. Then $\mathfrak{B}''$ is defined as follows.

1. $\mathsf{dom}(\mathfrak{B}'') = (H' \times \mathsf{dom}(\mathfrak{A}))/_\sim$, where $\sim$ identifies all $(h,d),(h',d')$ such that $(h,d) = (h',d')$ or there exists $c \in \mathsf{dom}(h) \cap \mathsf{dom}(h')$ such that $h(c) = d$ and $h'(c) = d'$. Denote the equivalence class of (h,d) w.r.t. $\sim$ by $[h,d]$.
2. For any $R \in \mathsf{rel}$, let $R^{\mathfrak{B}''}$ be defined by setting, for $e_1,\dots,e_n \in \mathsf{dom}(\mathfrak{B}'')$, $\mathfrak{B}'' \models R(e_1,\dots,e_n)$ if there exists $h \in H'$ and $c_1,\dots,c_n \in \mathsf{dom}(\mathfrak{A})$ such that $e_i = [h,c_i]$ and $\mathfrak{A} \models R(c_1,\dots c_n)$.

For any constant c in $\mathcal{D} \setminus \mathcal{D}_{\mathbf{a}_0}$, we define $c^{\mathfrak{B}}$ as the copy of $c^{\mathfrak{A}}$ in $\mathfrak{B}'$. For any constant c in $\mathcal{D}_{\mathbf{a}_0}$, we set $c^{\mathfrak{B}} = c^{\mathfrak{B}''} = [h,h(e(c))]$, where $h \in H'$ is such that $e(c) \in \mathsf{dom}(h)$. This is well-defined as $(h',h'(e(c))) \sim (h,h(e(c)))$ for any $h' \in H'$ with $e(c) \in \mathsf{dom}(h')$. Then, an embedding from $\mathfrak{A}$ to $\mathfrak{B}''$ is given by the map $f_h : \mathsf{dom}(\mathfrak{A}) \to (H' \times \mathsf{dom}(\mathfrak{A}))/_\sim$ defined by $f_h(c) = [h,c]$. We show that $(\mathfrak{B},\mathbf{a}_0^{\mathfrak{B}}) \sim^{\ell}_{\mathbf{GF},\mathsf{sig}(\mathcal{K})} (\mathfrak{A},\mathbf{b}^{\mathfrak{A}})$. By construction and the assumption that ℓ exceeds the guarded quantifier rank of $\mathcal{K}$, it also follows that $\mathfrak{B}$ is a model of $\mathcal{K}$. It thus follows that we have derived a contradiction to the assumption that $\mathfrak{A}$ and ℓ witness Condition 3.

To define a guarded $\mathsf{sig}(\mathcal{K})$ ℓ-bisimulation $\hat{H}_\ell,\dots,\hat{H}_0$, let S_i be the set of $p : \mathbf{c} \mapsto \mathbf{d}$ witnessing that $(\mathfrak{A},\mathbf{c}) \sim^{i}_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} (\mathfrak{A},\mathbf{d})$, where $\mathbf{c}$ is guarded. Then include in $\hat{H}_i$

1. all $\mathbf{c}' \mapsto \mathbf{c}$, where $\mathbf{c}'$ is the copy in $\mathfrak{B}'$ of the guarded tuple $\mathbf{c}$ in $\mathfrak{A}$;

2. all compositions $p \circ (f_h^{-1})_{|[\mathbf{d}]}$ for any guarded tuple $\mathbf{d}$ in the range of f_h and $p \in S_i$;

In addition, include $\mathbf{a}_0^{\mathfrak{B}} \mapsto \mathbf{b}^{\mathfrak{A}}$ in all $\hat{H}_i$, $0 \leqslant i \leqslant \ell$. We prove that $\hat{H}_\ell, \ldots, \hat{H}_0$ is a guarded sig($\mathcal{K}$) ℓ-bisimulation witnessing $(\mathfrak{B}, \mathbf{a}_0^{\mathfrak{B}}) \sim^\ell_{\mathbf{GF},\mathsf{sig}(\mathcal{K})} (\mathfrak{A}, \mathbf{b}^{\mathfrak{A}})$. For any $k \in \{0, \ldots, \ell\}$, any element g in $\hat{H}_k$ is clearly a partial sig($\mathcal{K}$)-isomorphism, either trivially if $\mathsf{dom}(g) \subseteq \mathfrak{B}'$ or by composition of partial sig($\mathcal{K}$)-isomorphisms if $\mathsf{dom}(g) \subseteq \mathfrak{B}''$. By definition, $\hat{H}_\ell$ contains $\mathbf{a}_0^{\mathfrak{B}} \mapsto \mathbf{b}^{\mathfrak{A}}$, thus $\hat{H}_k$ does for any $k \in \{0, \ldots, \ell\}$. We then only need to check satisfaction of the Forth and Back conditions. Let $g \in \hat{H}_k$ for some $k \in \{1, \ldots, \ell\}$. By definition of $\hat{H}_k$, we have either $\mathsf{dom}(g) \subseteq \mathfrak{B}'$ or $\mathsf{dom}(g) \subseteq \mathfrak{B}''$. In each case, we show that for any $\mathbf{c}$ guarded in $\mathfrak{B}$ and $\mathbf{d}$ guarded in $\mathfrak{A}$, there exists $g_0' \in \hat{H}_{k-1}$ of domain $[\mathbf{c}]$ that coincides with g on $[\mathbf{c}] \cap \mathsf{dom}(g)$ (Forth) and there exists $g_1' \in \hat{H}_{k-1}$ such that $\mathsf{dom}((g_1')^{-1}) = [\mathbf{d}]$ and $(g_1')^{-1}$ coincides with g^{-1} on $[\mathbf{d}] \cap \mathsf{im}(g)$ (Back).

The subcase $[\mathbf{c}] \cap \mathsf{dom}(g) = \emptyset$ is straightforward: because it is guarded in $\mathfrak{B}$, $\mathbf{c}$ is either included in $\mathfrak{B}'$ or included in $\mathfrak{B}''$. If $\mathbf{c}$ is in $\mathfrak{B}'$, then the partial isomorphism mapping $\mathbf{c}$ to its copy in $\mathfrak{A}$ is in $\hat{H}_{k-1}$ and as required. If $\mathbf{c}$ is in $\mathfrak{B}''$, then $\mathbf{c}$ can be written $([h, c_i])_{1 \leqslant i \leqslant n}$ for some $h \in H'$ and $c_1, \ldots, c_n \in \mathsf{dom}(\mathfrak{A})$ as it is guarded. But then $(f_h)^{-1}_{|[\mathbf{c}]} \in \hat{H}_{k-1}$ is as required. Similarly, the case when $[\mathbf{d}] \cap \mathsf{im}(g) = \emptyset$ is straightforward: if we write $\mathbf{d} := (d_1, \ldots, d_n)$ and $[h, \mathbf{d}] := ([h, d_i])_{1 \leqslant i \leqslant n} \in \mathfrak{B}''$ for any $h \in H'$, then $(f_h)^{-1}_{|[h,\mathbf{d}]} \in \hat{H}_{k-1}$ is as required, for any $h \in H'$. We now focus on proving (Forth) and (Back) assuming intersections are not empty.

1. Suppose $\mathsf{dom}(g) \subseteq \mathfrak{B}'$.

(1. Forth) Let $\mathbf{c}$ be guarded in $\mathfrak{B}$ such that $[\mathbf{c}] \cap \mathsf{dom}(g) \neq \emptyset$. We show there exists $g' \in \hat{H}_{k-1}$ that coincides with g on $\mathsf{dom}(g) \cap \mathsf{dom}(g')$, such that $[\mathbf{c}] = \mathsf{dom}(g')$. By construction of $\mathfrak{B}$, $[\mathbf{c}] \cap \mathsf{dom}(g) \neq \emptyset$ and $\mathsf{dom}(g) \subseteq \mathfrak{B}'$ imply $[\mathbf{c}] \subseteq \mathfrak{B}'$. By definition of $\hat{H}_k$, $\mathsf{dom}(g)$ is the copy of $\mathsf{im}(g)$ in $\mathfrak{B}'$. Therefore simply take g' to be the partial isomorphism $\mathbf{c} \mapsto \mathbf{d}$ such that $\mathbf{c}$ is the copy in $\mathfrak{B}'$ of $\mathbf{d}$; it clearly coincides with g on the intersection of their domains, and is in $\hat{H}_{k-1}$ which contains every "copy" mapping, by definition.

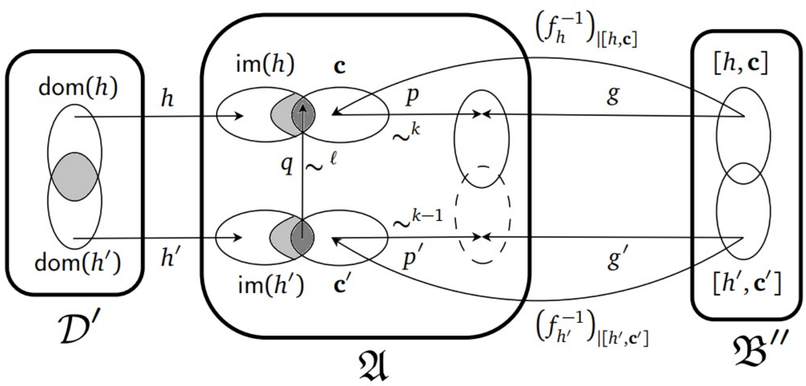
(1. Back) Let $\mathbf{d}$ be guarded in $\mathfrak{A}$ such that $[\mathbf{d}] \cap \mathsf{im}(g) \neq \emptyset$. Take $\mathbf{c}$ to be the copy in $\mathfrak{B}'$ of $\mathbf{d}$. Then, the partial isomorphism $g' := \mathbf{c} \mapsto \mathbf{d}$ is in $\hat{H}_{k-1}$ by definition, and is such that $(g')^{-1}$ coincides with g^{-1} on $\mathsf{im}(g) \cap \mathsf{im}(g')$.

2. Suppose $\mathsf{dom}(g) \subseteq \mathfrak{B}''$.

(2. Forth) Write $\mathsf{dom}(g)$ as $([h_i, c_i])_{1 \leqslant i \leqslant n}$ with $h_1, \ldots, h_n \in H'$ and $(c_1, \ldots, c_n) =:$ $\mathbf{c}$ a tuple in $\mathfrak{A}$. We want to prove that for any $([h_i', c_i'])_{1 \leqslant i \leqslant m}$ guarded in $\mathfrak{B}$ that intersects $\mathsf{dom}(g)$ there exists $g' \in \hat{H}_{k-1}$ of domain $([h', c_i'])_{1 \leqslant i \leqslant m}$ that coincides with g on $\mathsf{dom}(g) \cap \mathsf{dom}(g')$. Because $([h_i', c_i'])_{1 \leqslant i \leqslant m}$ is guarded in $\mathfrak{B}$ and intersects $\mathsf{dom}(g)$ which is in $\mathfrak{B}''$, it also has to be contained in $\mathfrak{B}''$ by

construction of $\mathfrak{B}$. The fact it is guarded implies we can write it as $([h', c'_i])_{1 \leqslant i \leqslant m}$ for some $h' \in H'$, with $(c'_1, \ldots, c'_m)$ being guarded in $\mathfrak{A}$, still by construction of $\mathfrak{B}$. As for $([h_i, c_i])_{1 \leqslant i \leqslant n}$, we can write it as $([h, c_i])_{1 \leqslant i \leqslant n}$ for some $h \in H'$, either because it is guarded or because it is equal to $\mathbf{a}^{\mathfrak{B}}$, *i.e.* $([h, h(a_i)])_{1 \leqslant i \leqslant n}$ for some $h \in H'$. By definition of $\hat{H}_k$, we can write g as $p \circ (f_h^{-1})_{|\mathsf{dom}(g)}$ for some $p \in S_k$, and we know g' has to be of the form $p' \circ (f_{h'}^{-1})_{|\mathsf{dom}(g')}$ for some $p' \in S_{k-1}$. For notation purposes, we write $[h, \mathbf{c}] = ([h, c_i])_{1 \leqslant i \leqslant n}$ and $[h', \mathbf{c}'] = ([h', c'_i])_{1 \leqslant i \leqslant m}$.

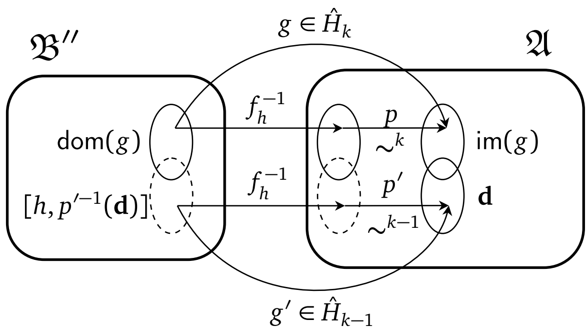
Suppose $h \neq h'$. For all $[h, c_i]$ in $[[h, \mathbf{c}]] \cap [[h', \mathbf{c}']]$ we have $c_i = h(d_i)$ and $c'_i = h'(d_i)$ for some $d_i \in \mathsf{dom}(h) \cap \mathsf{dom}(h')$. For any tuple $\mathbf{d}$ in $\mathcal{D}'$ such that $[\mathbf{d}] = \mathsf{dom}(h) \cap \mathsf{dom}(h')$, a partial isomorphism $q : [h'(\mathbf{d})] \to [h(\mathbf{d})]$ witnesses $(\mathfrak{A}, h'(\mathbf{d})) \sim^{\ell}_{\mathbf{GF},\mathsf{sig}(\mathcal{K})} (\mathfrak{A}, h(\mathbf{d}))$. Via p, for any $\mathbf{d}'$ such that $[\mathbf{d}'] = [h(\mathbf{d})] \cap [\mathbf{c}]$, we have $(\mathfrak{A}, \mathbf{d}') \sim^{k}_{\mathbf{GF},\mathsf{sig}(\mathcal{K})} (\mathfrak{A}, p(\mathbf{d}'))$. By composition, for any $\mathbf{d}''$ such that $[\mathbf{d}''] = [h'(\mathbf{d})] \cap [\mathbf{c}']$ we have $(\mathfrak{A}, \mathbf{d}'') \sim^{k}_{\mathbf{GF},\mathsf{sig}(\mathcal{K})} (\mathfrak{A}, p(q(\mathbf{d}'')))$. Because $p \circ q$ is in S_k (by definition of S_k) and because $[\mathbf{c}']$ trivially intersects $[h'(\mathbf{d})] \cap [\mathbf{c}']$, there exists, by definition of guarded k-bisimulations, a partial isomorphism $p' \in S_{k-1}$ of domain $[\mathbf{c}']$ that coincides with $p \circ q$ on $[h'(\mathbf{d})] \cap [\mathbf{c}']$. Then, $g' := p' \circ (f_{h'}^{-1})_{|[h', \mathbf{c}']}$ is the desired partial isomorphism in $\hat{H}_{k-1}$.



Suppose $h = h'$. Then $[h, \mathbf{c}] \cap [h, \mathbf{c}'] \neq \emptyset$ implies $[\mathbf{c}] \cap [\mathbf{c}'] \neq \emptyset$. Then, since $p \in S_k$ and $\mathsf{dom}(p) = [\mathbf{c}]$, by definition of guarded k-bisimulations there exists $p' \in S_{k-1}$ of domain $[\mathbf{c}']$ that coincides with p on $[\mathbf{c}] \cap [\mathbf{c}']$. Then $g' := p' \circ (f_h^{-1})_{|[h, \mathbf{c}']}$ in $\hat{H}_{k-1}$ is as required.

(2. Back) Let $\mathbf{d}$ be guarded in $\mathfrak{A}$ such that $[\mathbf{d}] \cap \mathsf{im}(g) \neq \emptyset$. We show there exists $g' \in \hat{H}_{k-1}$ of image $[\mathbf{d}]$ such that $(g')^{-1}$ coincides with g^{-1} on $[\mathbf{d}] \cap \mathsf{im}(g)$. By definition of $\hat{H}_k$ we can write $g = p \circ (f_h^{-1})_{|\mathsf{dom}(g)}$ for some $h \in H'$ and $p \in S_k$, and we know g' has to be of the form $p' \circ (f_{h'}^{-1})_{|[\mathbf{d}]}$ for some $h' \in H'$. By definition of guarded k-bisimulations, there exists $p' \in S_{k-1}$ such that $\mathsf{im}(p') = [\mathbf{d}]$ and p'^{-1} coincides with p^{-1} on $\mathsf{im}(p) \cap \mathsf{im}(p')$. Given that $\mathsf{im}(g) = \mathsf{im}(p)$, if we write $\mathbf{d} :=$

$(d_1, \ldots, d_n)$ and $[h, p'^{-1}(\mathbf{d})]$ for $([h, p'^{-1}(d_i)])_{1 \leqslant i \leqslant n}$, then $g' = p' \circ (f_h)^{-1}_{|[h,p'^{-1}(\mathbf{d})]}$ in $\hat{H}_{k-1}$ is as required.



⊣

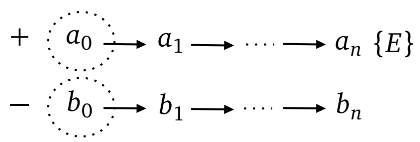
We get a similar result for projective separability when replacing sig($\mathcal{K}$) by the full signature. It is proved in the same way.

2.91. Theorem. *Let $(\mathcal{K}, E^+, \{\mathbf{b}\})$ be a labeled* **GF***-knowledge base. The following conditions are equivalent.*

1. $(\mathcal{K}, E^+, \{\mathbf{b}\})$ *is projectively* **oGF***-separable.*
2. $(\mathcal{K}, E^+, \{\mathbf{b}\})$ *is projectively* **GF***-separable.*
3. *There exist $\mathfrak{A} \models \mathcal{K}$ (finite), $\ell \geqslant 0$ such that $(\mathfrak{B}, \mathbf{a}^{\mathfrak{B}}) \not\sim^{\ell}_{\mathbf{GF}} (\mathfrak{A}, \mathbf{b}^{\mathfrak{A}})$ for all $\mathfrak{B} \models \mathcal{K}, \mathbf{a} \in E^+$.*
4. *There exist $\mathfrak{A} \models \mathcal{K}$ (finite), $\ell \geqslant 0$ such that $(\mathcal{D}_{\mathbf{a}}, \mathbf{a}) \not\preceq^{\ell}_{\mathbf{oGF}} (\mathfrak{A}, \mathbf{b}^{\mathfrak{A}})$ for all $\mathbf{a} \in E^+$.*
5. *There exist $\mathfrak{A} \models \mathcal{K}$ (finite), $\ell \geqslant 0$ such that $(\mathfrak{B}, \mathbf{a}^{\mathfrak{B}}) \not\sim^{\ell}_{\mathbf{oGF}} (\mathfrak{A}, \mathbf{b}^{\mathfrak{A}})$ for all $\mathfrak{B} \models \mathcal{K}, \mathbf{a} \in E^+$.*

While (**GF**, **GF**)-separability coincides with (**GF**, **oGF**)-separability in the projective and the non-projective case, using **oGF** for separation instead of **GF** can come at the expense of much larger formulas.

2.92. oGF *can force large formulas.* Let $\mathcal{D}$ (depicted below) contain two R-paths of length n, $a_0 R a_1 R \ldots R a_n$ and $b_0 R b_1 R \ldots R b_n$ with a_n labeled with E.



Let $\mathcal{O} = \{A \sqsubseteq \forall R.A,\ \forall xy(R(x,y) \rightarrow \neg R(y,x))\}$. Consider the labeled **GF**-knowledge base $(\mathcal{K}, \{a_0\}, \{b_0\})$ with $\mathcal{K} = (\mathcal{O}, \mathcal{D})$. Then the **GF**-formula $A(x) \rightarrow$

$\exists y(A(y) \wedge E(y))$ separates $(\mathcal{K}, \{a_0\}, \{b_0\})$, but the shortest separating **oGF**-formula has guarded quantifier rank n. To prove that no **oGF**-formula of depth $m < n$ separates $(\mathcal{K}, \{a_0\}, \{b_0\})$, it is sufficient to prove that for all models $\mathfrak{A}$ of $\mathcal{K}$ there exists a model $\mathfrak{B}$ of $\mathcal{K}$ such that $(\mathfrak{A}, b_0^{\mathfrak{A}}) \sim^m_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} (\mathfrak{B}, a_0^{\mathfrak{B}})$. Let $\mathfrak{A}$ be a model of $\mathcal{K}$. Define $\mathfrak{B}$ as the disjoint union of the standard unfolding $\mathfrak{A}^*_{a_0}$ of $\mathfrak{A}$ at $b_0^{\mathfrak{A}}$ into a guarded tree-decomposable model [HLPW20] and $\mathfrak{A}$, modified by interpreting $a_i^{\mathfrak{B}}$, $0 \leqslant i \leqslant n$, by the strict $R^{\mathfrak{A}^*_a}$-chain starting at $b_0^{\mathfrak{A}}$ and corresponding to the path $b_0^{\mathfrak{A}} R^{\mathfrak{A}} \cdots R^{\mathfrak{A}} b_n^{\mathfrak{A}}$, adding $a_n^{\mathfrak{B}}$ to $E^{\mathfrak{B}}$, and setting $b_i^{\mathfrak{B}} := b_i^{\mathfrak{A}}$, for $0 \leqslant i \leqslant n$. It is straightforward to check that $\mathfrak{B} \models \mathcal{K}$ and $(\mathfrak{A}, b_0^{\mathfrak{A}}) \sim^m_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} (\mathfrak{B}, a_0^{\mathfrak{B}})$ for all $m < n$.

2.5.3. *Final characterisation*

We formulate a characterisation of projective and non-projective (**GF**, **GF**) separability using, once again, a similar approach to the $\mathcal{ALCI}$ case's, based on homomorphisms from the database and, in the non-projective case, on a **GF** version of type incompleteness. For **GF**, we need to work with a slightly more involved notion of "cl($\mathcal{K}$)-type" than in $\mathcal{ALCI}$.

2.93. *Definition.* For a **GF**-knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{D})$, let $\mathsf{cl}(\mathcal{K})$ denote the smallest superset of $\mathcal{K}$ that is closed under subformulas, single negation and contains the following, assuming for all $n \geq 1$ a tuple $\mathbf{x}_n$ of length n, made of distinct variables, to be fixed.

1. $x = y$ for all distinct variables x, y,
2. For all $R \in \mathsf{sig}(\mathcal{K})$ of arity $n \geq 2$ and all distinct $x, y \in [\mathbf{x}_n]$, the formulas $R(\mathbf{x}_n)$, $\exists \mathbf{y}_1\, (R(\mathbf{x}_n) \wedge x \neq y)$ where $\mathbf{y}_1$ is $\mathbf{x}_n$ without x, and $\exists \mathbf{y}_2\, R(\mathbf{x}_n)$ for all $\mathbf{y}_2$ with $[\mathbf{y}_2] \subseteq [\mathbf{x}_n] \setminus \{x, y\}$ (for $n \geq 3$).

Let $\mathfrak{A}$ be a model and $\mathbf{a}$ a tuple in $\mathfrak{A}$. In this section we call the cl($\mathcal{K}$)-type of $\mathbf{a}$ in $\mathfrak{A}$ the set $\mathsf{tp}^{cl}_{\mathcal{K}}(\mathfrak{A}, \mathbf{a}) := \{\varphi \in \mathsf{cl}(\mathcal{K})[\mathbf{x}] \mid \mathfrak{A} \models \varphi(\mathbf{a})\}$, where $\mathsf{cl}(\mathcal{K})[\mathbf{x}]$ is obtained from $\mathsf{cl}(\mathcal{K})$ by substituting in any formula $\varphi \in \mathsf{cl}(\mathcal{K})$ the free variables of φ by variables in $\mathbf{x}$ in all possible ways, $\mathbf{x}$ being a tuple of same length as $\mathbf{a}$. We then call cl($\mathcal{K}$)-type any set t such that $t = \mathsf{tp}^{cl}_{\mathcal{K}}(\mathfrak{A}, \mathbf{a})$ for some $(\mathfrak{A}, \mathbf{a})$. A cl($\mathcal{K}$)-type is said to be *connected* if it contains a formula of the form $\exists \mathbf{y}_1\, (R(\mathbf{x}_n) \wedge x \neq y)$ as described above. A *guarded* cl($\mathcal{K}$)*-type* $\Phi(\mathbf{x})$ is a cl($\mathcal{K}$)-type that contains an atom $R(\mathbf{x})$. Call cl($\mathcal{K}$)-types $\Phi_1(\mathbf{x}_1)$ and $\Phi_2(\mathbf{x}_2)$ *coherent* if there exists a model $\mathfrak{A}$ of $\mathcal{K}$ satisfying $\Phi_1 \cup \Phi_2$ under an assignment μ for the variables in $[\mathbf{x}_1] \cup [\mathbf{x}_2]$. For a cl($\mathcal{K}$)-type $\Phi(\mathbf{x})$ and a subsequence $\mathbf{x}_I$ of $\mathbf{x}$ we denote by $\Phi_{|\mathbf{x}_I}$ (*restriction of* Φ *to* $\mathbf{x}_I$) the subset of Φ containing all formulas in Φ with free variables from $\mathbf{x}_I$.

Observe that cl($\mathcal{K}$)-types $\Phi_1(\mathbf{x}_1)$ and $\Phi_2(\mathbf{x}_2)$ are coherent iff their restrictions to

$[\mathbf{x}_1] \cap [\mathbf{x}_2]$ are logically equivalent.

2.94. *Definition.* Assume a $\mathsf{cl}(\mathcal{K})$-type $\Phi(x)$ is given. We say $\Phi(x)$ is *complete* if for all pointed models $(\mathfrak{A}, a), (\mathfrak{B}, b)$ it holds that $\mathrm{tp}^{cl}_{\mathcal{K}}(\mathfrak{A}, a) = \mathrm{tp}^{cl}_{\mathcal{K}}(\mathfrak{B}, b) = \Phi(x)$ implies $(\mathfrak{A}, a) \sim_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} (\mathfrak{B}, b)$. We say a sequence $\sigma = (\Phi_i(\mathbf{x}_i))_{i=0}^{n+1}$ *witnesses* **oGF**-*incompleteness of* Φ if Φ is the restriction of Φ_0 to x, $n \geqslant 0$, and all Φ_i, $0 \leqslant i \leqslant n+1$, are guarded $\mathsf{cl}(\mathcal{K})$-types each containing $\neg(x = y)$ for some variables x, y (we say that the Φ_i are *non-unary*) such that

1. $[\mathbf{x}_i] \cap [\mathbf{x}_{i+1}] \neq \emptyset$;
2. all Φ_i, Φ_{i+1} are coherent;
3. there exists a model $\mathfrak{A}$ of $\mathcal{K}$ and a tuple $\mathbf{a}_n$ in $\mathfrak{A}$ such that $\mathfrak{A} \models (\Phi_n \wedge \neg\exists \mathbf{x}'_{n+1} \Phi_{n+1})(\mathbf{a}_n)$, where $\mathbf{x}'_{n+1}$ is the sequence $\mathbf{x}_{n+1}$ without $[\mathbf{x}_n] \cap [\mathbf{x}_{n+1}]$.

The following is a **GF** counterpart of Lemma 2.73. We also use it to make up for the lack of helper symbols.

2.95. Lemma. *The following conditions are equivalent for any* $\mathsf{cl}(\mathcal{K})$*-type* $\Phi(x)$.

1. $\Phi(x)$ *is not* **oGF**-*complete.*
2. *There exists a sequence witnessing* **oGF**-*incompleteness of* $\Phi(x)$.
3. *There exists a sequence of length* $\leqslant 2^{2^{||\mathcal{K}||}} + 2$ *witnessing* **oGF**-*incompleteness of* $\Phi(x)$.

Proof. One can, similarly to what is done in the proof of Lemma 2.71, construct a guarded tree decomposable model $\mathfrak{A}$ of $\mathcal{O}$ with tree decomposition (T, E, bag) and root r such that $\Phi(x)$ is realized in $\mathrm{bag}(r)$ by a and for every $\mathsf{cl}(\mathcal{K})$-type $\Psi_1(\mathbf{x})$ realized in some $\mathrm{bag}(t)$ by $\mathbf{a}$ and every $\mathsf{cl}(\mathcal{K})$-type $\Psi_2(\mathbf{y})$ coherent with $\Psi_1(\mathbf{x})$ there exists a successor t' of t in T such that $\Psi_1(\mathbf{x}) \cup \Phi_2(\mathbf{y})$ is realized in $\mathrm{bag}(t) \cup \mathrm{bag}(t')$ in $\mathfrak{A}$ under an assignment μ of the variables $[\mathbf{x}] \cup [\mathbf{y}]$ such that $\mu(\mathbf{x}) = \mathbf{a}$. Thus, $\mathfrak{A}$ satisfies $\forall \mathbf{x}(\Psi_1 \rightarrow \exists \mathbf{y}' \Psi_2)$ for any coherent pair $\Psi_1(\mathbf{x}), \Psi_2(\mathbf{y})$, where $\mathbf{y}'$ is $\mathbf{y}$ without $[\mathbf{x}] \cap [\mathbf{y}]$.

(1) $\Rightarrow$ (2). If $\Phi(x)$ is not **oGF**-complete, then there exists a guarded tree decomposable model $\mathfrak{A}'$ of $\mathcal{K}$ with root r which realizes $\Phi(x)$ in $\mathrm{bag}(r)$ at a' such that $(\mathfrak{A}, a) \not\sim_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} (\mathfrak{A}', a')$. It is immediate that $(\mathfrak{A}, a)$ realizes a sequence σ that witnesses **oGF**-incompleteness of $\Phi(x)$, unless there exists a guarded non-unary $\mathsf{cl}(\mathcal{K})$-type $\Phi_0(\mathbf{x}_0)$ which is realized in some $\mathbf{a}_0$ in $\mathfrak{A}$ with $a \in [\mathbf{a}_0]$ but no $\mathbf{a}'_0$ in $\mathfrak{A}'$ containing a' and realizing $\Phi_0(\mathbf{x}_0)$. Let $R_0(\mathbf{x}_0) \in \Phi_0$. Then, because of condition (2) in Definition 2.93 and because it is non-unary, $\Phi_0(\mathbf{x}_0)$ contains $\exists \mathbf{y}(R_0(\mathbf{x}_0) \wedge x \neq y)$ for any $y \in [\mathbf{y}]$ and $[\mathbf{y}] = \mathbf{x}_0 \setminus \{x\}$. Then, they are also contained in $\Phi(x)$ (as restriction of Φ_0), and thus satisfied by $(\mathfrak{A}', a')$. That implies there exists a non-unary guarded $\mathsf{cl}(\mathcal{K})$-type $\Phi'(\mathbf{x}'_0)$ containing $R_0(\mathbf{x}'_0)$ such that there exists a tuple $\mathbf{a}'_0$ in $\mathfrak{A}'$ containing a' realizing Φ'. We obtain a

sequence σ of any length by first taking $\Phi'(\mathbf{x}_0)$ an arbitrary number of times and then appending Φ_0.

(2) ⇒ (3). This can be proved by a straightforward pumping argument, in particular if one works with a sequence σ realized by a strict path. Consider a sequence $\sigma = \Phi_0(\mathbf{x}_0), \ldots, \Phi_n(\mathbf{x}_n), \Phi_{n+1}(\mathbf{x}_{n+1})$ that witnesses **oGF**-incompleteness of $\Phi(x)$. We may assume (by possibly repeating Φ_n once in the sequence) that there is a model $\mathfrak{A}$ of $\mathcal{K}$ with a path $R_0(\mathbf{a}_0), \ldots, R_n(\mathbf{a}_n)$ such that $\mathbf{a}_i$ realizes Φ_i and $\mathfrak{A} \models (\Phi_n \wedge \neg\exists \mathbf{x}'_{n+1}\Phi_{n+1})(\mathbf{a}_n)$. We now modify $\mathfrak{A}$ in such a way that we obtain a sequence witnessing **oGF**-incompleteness of $\Phi(x)$ which is realized by a strict path. Choose a sequence $\mathbf{a} = (a_1, \ldots, a_m)$ such that $a_1 = a$ for the node a in $\mathbf{a}_0$ realizing $\Phi(x)$, $a_i \neq a_{i+1}$ and $a_i, a_{i+1} \in [\mathbf{a}_j]$ for some $j \leqslant n$, for all $i < m$, and $a_m \in \mathbf{a}_n$. Clearly one can find such a sequence for some $m \leqslant 2n$. Then take the partial unfolding $\mathfrak{A}_{\mathbf{a}}$ of $\mathfrak{A}$ along $\mathbf{a}$. In $\mathfrak{A}_{\mathbf{a}}$ we find the required strict path (Lem. 2.86). Pumping on this path is straightforward.

(3) ⇒ (1). Straightforward. ⊣

2.96. A 2EXP upper bound for deciding whether a guarded $\mathsf{cl}(\mathcal{K})$-type is **oGF**-complete can now be proved similarly to the EXP upper bound for deciding whether a type defined by an $\mathcal{ALCI}$-knowledge base is $\mathcal{ALCI}$-complete (2.80).

2.97. Lemma. *A guarded* $\mathsf{cl}(\mathcal{K})$*-type* $\Phi(\mathbf{x})$ *is* **oGF***-complete iff all restrictions* $\Phi(x)$ *of* Φ *to some variable* x *in* $\mathbf{x}$ *are* **oGF***-complete.*

Proof. The left to right direction is straightforward. Conversely, assume that $\Phi(\mathbf{x})$ is not **oGF**-complete. One can show similarly to the proof of Lemma 2.95 that (i) or (ii) holds.

(i) There exists a guarded $\mathsf{cl}(\mathcal{K})$-type $\Phi_0(\mathbf{x}_0)$ sharing with $\mathbf{x}$ the variables $\mathbf{x}_I$ for some nonempty $I \subseteq \{1, \ldots, n\}$ such that for $\mathbf{x}'_0$ the variables in $\mathbf{x}_0$ without $\mathbf{x}_I$ the following holds:

  (1) there exists $\mathfrak{A} \models \mathcal{K}$ realizing Φ in a tuple $\mathbf{a}$ such that $\mathfrak{A} \models (\exists \mathbf{x}'_0 \Phi_0)(\mathbf{a}_I)$;
  (2) there exists $\mathfrak{A}' \models \mathcal{K}$ realizing Φ in a tuple $\mathbf{a}$ such that $\mathfrak{A}' \not\models (\exists \mathbf{x}'_0 \Phi_0)(\mathbf{a}_I)$.

(ii) There exists a guarded $\mathsf{cl}(\mathcal{K})$-tuple $\Phi_0(\mathbf{x}_0)$ sharing with $\mathbf{x}$ the variables $\mathbf{x}_I$ for some nonempty $I \subseteq \{1, \ldots, n\}$ and a sequence $\Phi_1(\mathbf{x}_1), \ldots, \Phi_n(\mathbf{x}_n), \Phi_{n+1}(x_{n+1})$ of guarded $\mathsf{cl}(\mathcal{K})$-tuples with $n \geqslant 1$ such that $\Phi(\mathbf{x}) \cup \Phi(\mathbf{x}_0)$ is satisfiable in a model of $\mathcal{K}$ and $\Phi_0(\mathbf{x}_0), \ldots, \Phi_{n+1}(x_{n+1})$ satisfy the conditions of a sequence witnessing non **oGF**-completeness, except that no type $\Phi(x)$ of which it witnesses non-**oGF**-completeness is given.

If (ii), then we are done by taking any variable x in $\mathbf{x}_I$ and the restriction $\Phi_{|x}$ of Φ to x. Then $\Phi_{|x}$ is not **oGF**-complete. Now assume that (i) holds. We are again done if I contains at most one element (we can simply take the type $\mathsf{tp}^{cl}_{\mathcal{K}}(\mathfrak{A}, a_I)$

then). Consider a relation R_0 with $R_0(\mathbf{x}_0) \in \Phi_0$. By the closure condition on $\mathsf{cl}(\mathcal{K})$-types, we have $\mathfrak{A}' \models \exists \mathbf{x}_0' R_0(\mathbf{x}_0)(\mathbf{a}_I)$. Take an extension $\mathbf{a}_1$ of $\mathbf{a}_I$ such that $\mathfrak{A}' \models R_0(\mathbf{a}_1)$. Take any $a \in \mathbf{a}_I$, the unary $\mathsf{cl}(\mathcal{K})$-type $\Phi(x) = \mathsf{tp}^{cl}_{\mathcal{K}}(\mathfrak{A}', a)$, and the $\mathsf{cl}(\mathcal{K})$-type $\Phi_1(\mathbf{x}_1) := \mathsf{tp}^{cl}_{\mathcal{K}}(\mathfrak{A}', \mathbf{a}_1)$. Then the sequence Φ_1, Φ_0 shows that $\Phi(x)$ is not **oGF**-complete. ⊣

The following lemma helps bridge the gap between embedding and homomorphism in the next Theorem, whose proof revolves around some subset of elements being pairwise non-bisimilar. See Lemma 2.75 in the $\mathcal{ALCI}$ case.

2.98. Lemma. *Let $(\mathcal{D}, \mathbf{a})$ be a pointed database, $(\mathfrak{A}, \mathbf{b})$ be a pointed model, and $\ell \geqslant |\mathcal{D}|$. Suppose that $(\mathcal{D}_{\mathbf{a}}, \mathbf{a}) \preceq^{\ell}_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} (\mathfrak{A}, \mathbf{b})$ and $(\mathcal{D}_{\mathbf{a}}, \mathbf{a}) \not\to (\mathfrak{A}, \mathbf{b})$. Then there exist d, d' with $d \neq d'$ in $\mathfrak{A}^{\leqslant |\mathcal{D}|}_{\mathbf{b}}$ such that $(\mathfrak{A}, d) \sim^{\ell - |\mathcal{D}|}_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} (\mathfrak{A}, d')$.*

Proof. Let $e, H_\ell, \dots, H_0$ witness $(\mathcal{D}_{\mathbf{a}}, \mathbf{a}) \preceq^{\ell}_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} (\mathfrak{A}, \mathbf{b})$ as in Lemma 2.88. We define a sequence of mappings $S_0, \dots, S_\ell$ with $S_k \subseteq H_{\ell-k}$ for $k \leqslant \ell$ as follows. Define $S_0 := \{e(\mathbf{a}) \mapsto \mathbf{b}\}$ and assume that S_k has been defined for some $k < \ell$. To define S_{k+1}, choose for every $h \in S_k$ and all guarded $\mathbf{c}$ intersecting $\mathsf{dom}(h)$ an $h' \in H_{\ell-k-1}$ with domain $[\mathbf{c}]$ that coincides with h on $[\mathbf{c}] \cap \mathsf{dom}(h)$ (this is possible by Condition 1 of Lemma 2.88) and add it to S_{k+1}. Define $\hat{h} := \bigcup(\bigcup_{k \leq |\mathcal{D}|} S_k)$. We can see $\hat{h}$ as a set of pairs from $\mathsf{cons}(\mathcal{D}'_{e(\mathbf{a})}) \times \mathsf{dom}(\mathfrak{A})$, which may or may not be functional. We know h is not a homomorphism from $(\mathcal{D}'_{e(\mathbf{a})}, e(\mathbf{a}))$ to $(\mathfrak{A}, \mathbf{b})$ because $h \circ e$ would otherwise witness $(\mathcal{D}_{\mathbf{a}}, \mathbf{a}) \to (\mathfrak{A}, \mathbf{b})$. However,

1. $\mathcal{D}' \models R(c_1, \dots, c_n)$ implies $\mathfrak{A} \models R(d_1, \dots, d_n)$ for every $(c_1, d_1), \dots, (c_n, d_n) \in \hat{h}$ and every n-ary $R \in \Sigma$, since $\hat{h}$ is a union of partial homomorphisms;
2. for every $c \in \mathcal{D}'_{e(\mathbf{a})}$ there exists $h \in \bigcup_{k \leqslant \mathsf{dist}_{\mathcal{D}'}(c, e(\mathbf{a}))} S_k$ such that $c \in \mathsf{dom}(h)$, so $\hat{h}$ is defined on the entire underlying set of $\mathcal{D}'_{e(\mathbf{a})}$, as $\mathsf{dist}_{\mathcal{D}'}(c, e(\mathbf{a}))$ is bounded by $|\mathcal{D}'|$ and $|\mathcal{D}'| \leqslant |\mathcal{D}|$ follows from surjectivity of e;
3. $e(\mathbf{a}) \mapsto \mathbf{b}$ is included in $\hat{h}$.

Therefore the only possible cause of $\hat{h}$ not being a homomorphism witnessing $(\mathcal{D}_{\mathbf{a}}, \mathbf{a}) \to (\mathfrak{A}, \mathbf{b})$ is that $\hat{h}$ is not functional, *i.e.* there exist $c \in \mathsf{cons}(\mathcal{D}'_{e(\mathbf{a})})$ and $d, d' \in \mathsf{dom}(\mathfrak{A})$ such that $d \neq d'$ and $(c, d), (c, d') \in \hat{h}$. As every h included in $\hat{h}$ is functional, that implies there exist $h, h' \in \bigcup_{k \leqslant |\mathsf{cons}(\mathcal{D})|} S_k$ such that $h(c) = d$ and $h'(c) = d'$. There exist $k_1, k_2 \geqslant \ell - |\mathsf{cons}(\mathcal{D})|$ such that $h \in H_{k_1}$ and $h' \in H_{k_2}$. By condition (2) of Lemma 2.88 we get

$$(\mathfrak{A}, d) \sim^{\min(k_1,k_2)}_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} (\mathfrak{A}, d'),$$

$$\text{hence } (\mathfrak{A}, d) \sim^{\ell - |\mathsf{cons}(\mathcal{D})|}_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} (\mathfrak{A}, d').$$

Finally, $d, d' \in \mathfrak{A}_{\mathbf{b}}^{\leq |\mathsf{cons}(\mathcal{D})|}$ follows from the fact that $\mathsf{dist}_{\mathfrak{A}}(h(c), \mathbf{b}) \leqslant k$ for any $c \in \mathsf{dom}(h)$ such that $h \in S_k$. This can be proved by induction on k. Case $k = 0$ is straightforward. Now suppose $h \in S_{k+1}$ and $c \in \mathsf{dom}(h)$. By definition of S_{k+1}, h coincides with some $h' \in S_k$ on $\mathsf{dom}(h) \cap \mathsf{dom}(h')$ which is not empty. By induction hypothesis $\mathsf{dist}_{\mathfrak{A}}(h'(c'), \mathbf{b}) \leqslant k$ for any $c' \in \mathsf{dom}(h')$ and there is a path of guarded sets $(\mathsf{dom}(h), \mathsf{dom}(h'))$ of length 1 between c and c', hence a path $(\mathsf{im}(h), \mathsf{im}(h'))$ of length 1 between $h(c)$ and $h(c')$, from the fact that h, h' are partial homomorphisms (guaranteeing that $\mathsf{im}(h), \mathsf{im}(h')$ are guarded) and the fact that they coincide on $\mathsf{dom}(h) \cap \mathsf{dom}(h')$. Then, we get $\mathsf{dist}_{\mathfrak{A}}(h(c), \mathbf{b}) \leqslant k + 1$. ⊣

2.99. Theorem. *A labeled* **GF***-knowledge base* $(\mathcal{K}, E^+, \{\mathbf{b}\})$ *with* $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ *and* $\mathbf{b} = (b_1, \dots, b_n)$ *is weakly non-projectively* **GF***-separable iff there exists* $\mathfrak{A} \models \mathcal{K}$ *such that for all* $\mathbf{a} \in E^+$*, the following conditions are met.*

1. $(\mathcal{D}_{\mathbf{a}}, \mathbf{a}) \not\to (\mathfrak{A}, \mathbf{b}^{\mathfrak{A}})$
2. *if* $I = \{i : \mathsf{tp}_{\mathcal{K}}^{cl}(\mathfrak{A}, b_i^{\mathfrak{A}})$ *connected and* **oGF***-complete*$\} \neq \emptyset$*, then*
   either $J = \{1, \dots, n\} \setminus I \neq \emptyset$ *and* $(\mathcal{D}_{\mathbf{a}_J}, \mathbf{a}_J) \not\to (\mathfrak{A}, \mathbf{b}_J^{\mathfrak{A}})$
   or $\mathsf{tp}_{\mathcal{K}}^{cl}(\mathfrak{A}, \mathbf{b}^{\mathfrak{A}})$ *is not realizable in* $(\mathcal{K}, \mathbf{a})$*.*

For projective separability, Point 2 must be dropped.

Proof.
$(\Rightarrow)$. Assume $(\mathcal{K}, E^+, \{\mathbf{b}\})$ is non-projectively **GF**-separable. By Theorem 2.90, there exists a finite model $\mathfrak{A}$ of $\mathcal{K}$ and $\ell_0 \geqslant 0$ such that $(\mathcal{D}_{\mathbf{a}}, \mathbf{a}) \not\preceq^{\ell_0}_{\mathbf{oGF}, \mathsf{sig}(\mathcal{K})} (\mathfrak{A}, \mathbf{b}^{\mathfrak{A}})$ for all $\mathbf{a} \in E^+$. Assume $\mathbf{a} \in E^+$ is given. As $(\mathcal{D}_{\mathbf{a}}, \mathbf{a}) \to (\mathfrak{A}, \mathbf{b}^{\mathfrak{A}})$ implies $(\mathcal{D}_{\mathbf{a}}, \mathbf{a}) \preceq^{\ell}_{\mathbf{oGF}, \mathsf{sig}(\mathcal{K})} (\mathfrak{A}, \mathbf{b}^{\mathfrak{A}})$ for all $\ell \geqslant 0$, Condition 1 holds. To show that Condition 2 holds for $\mathfrak{A}$ and $\mathbf{a}$, assume that I as defined in the theorem is not empty and that $\mathsf{tp}_{\mathcal{K}}^{cl}(\mathfrak{A}, \mathbf{b}^{\mathfrak{A}})$ is realizable in $(\mathcal{K}, \mathbf{a})$. Take a model $\mathfrak{B}$ witnessing this. Consider the maximal sets $I_1, \dots, I_k \subseteq \{1, \dots, n\}$ such that $\mathbf{b}_{I_j}^{\mathfrak{B}}$ is in a connected component $\mathfrak{B}_j$ of $\mathfrak{B}$. Then there exists at least one j such that $\mathsf{tp}_{\mathcal{K}}^{cl}(\mathfrak{A}, \mathbf{b}_{I_j}^{\mathfrak{A}})$ is not **oGF**-complete or not connected: otherwise the following sequence of implications holds and leads to a contradiction.

$$
\begin{array}{lll}
 & \mathsf{tp}_{\mathcal{K}}^{cl}(\mathfrak{B}, \mathbf{a}_{I_j}^{\mathfrak{B}}) = \mathsf{tp}_{\mathcal{K}}^{cl}(\mathfrak{A}, \mathbf{b}_{I_j}^{\mathfrak{A}}) & \text{for all } j \in \{1, \dots, k\} \\
\Rightarrow & (\mathfrak{B}, \mathbf{a}_{I_j}^{\mathfrak{B}}) \sim_{\mathbf{oGF}, \mathsf{sig}(\mathcal{K})} (\mathfrak{A}, \mathbf{b}_{I_j}^{\mathfrak{A}}) & \text{for all } j \in \{1, \dots, k\} \\
\Rightarrow & (\mathcal{D}_{\mathbf{a}_{I_j}}, \mathbf{a}_{I_j}) \preceq^{\ell}_{\mathbf{oGF}, \mathsf{sig}(\mathcal{K})} (\mathfrak{A}, \mathbf{b}_{I_j}^{\mathfrak{A}}) & \text{for all } \ell \geqslant 0, j \in \{1, \dots, k\} \\
\Rightarrow & (\mathcal{D}_{\mathbf{a}}, \mathbf{a}) \preceq^{\ell}_{\mathbf{oGF}, \mathsf{sig}(\mathcal{K})} (\mathfrak{A}, \mathbf{b}^{\mathfrak{A}}) & \text{for all } \ell \geqslant 0, j \in \{1, \dots, k\}
\end{array}
$$

For any j such that $\mathsf{tp}_{\mathcal{K}}^{cl}(\mathfrak{A}, \mathbf{b}_{I_j}^{\mathfrak{A}})$ is not **oGF**-complete we have by Lemma 2.97 that $\mathsf{tp}_{\mathcal{K}}^{cl}(\mathfrak{A}, \mathbf{b}_i^{\mathfrak{A}})$ is not **oGF**-complete for any $i \in I_j$. Therefore $J \neq \emptyset$. Assume now for a proof by contradiction that $(\mathcal{D}_{\mathbf{a}_J}, \mathbf{a}_J) \to (\mathfrak{A}, \mathbf{b}_J^{\mathfrak{A}})$. Then, $(\mathcal{D}_{\mathbf{a}_J}, \mathbf{a}_J) \preceq^{\ell}_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} (\mathfrak{A}, \mathbf{b}_J^{\mathfrak{A}})$ for any $\ell \geqslant 0$. By Lemma 2.97, $\mathsf{tp}^{cl}_{\mathcal{K}}(\mathfrak{A}, \mathbf{b}_I^{\mathfrak{A}})$ is **oGF**-complete. Then, $(\mathfrak{B}, \mathbf{a}_I^{\mathfrak{B}}) \sim_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} (\mathfrak{A}, \mathbf{b}_I^{\mathfrak{A}})$ thus $(\mathcal{D}_{\mathbf{a}_I}, \mathbf{a}_I) \preceq^{\ell}_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} (\mathfrak{A}, \mathbf{b}_I^{\mathfrak{A}})$ for all $\ell \geqslant 0$. Then, since $(\mathcal{D}_{\mathbf{a}_J}, \mathbf{a}_J) \preceq^{\ell}_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} (\mathfrak{A}, \mathbf{b}_J^{\mathfrak{A}})$ for all $\ell \geqslant 0$ and $\mathcal{D}_{\mathbf{a}_I}, \mathcal{D}_{\mathbf{a}_J}$ are disjoint, we get $(\mathcal{D}_{\mathbf{a}}, \mathbf{a}) \preceq^{\ell}_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} (\mathfrak{A}, \mathbf{b}^{\mathfrak{A}})$ for all $\ell \geqslant 0$.

($\Leftarrow$). For the projective case, assume Condition 1 holds for some $\mathfrak{A} \models \mathcal{K}$, that we can assume without loss of generality such that for each $x \in \mathsf{dom}(\mathfrak{A})$ there exists a fresh unary predicate A_x such that $A_x^{\mathfrak{A}} = \{x\}$. Then $(\mathfrak{A}, x) \not\sim_{\mathbf{GF}} (\mathfrak{A}, y)$ for any $x \neq y$. Then Condition 1 implies Condition 3 of Theorem 2.90, which concludes the proof.

For the non-projective case, assume Conditions 1 and 2 hold for some model $\mathfrak{A}$ of $\mathcal{K}$ and all $\mathbf{a} \in E^+$. As **GF** is finitely controllable, there exists a finite such model $\mathfrak{A}$. Assume that the set I defined in the theorem is empty: the case in which it is not empty is very similar to this case and omitted. Let X be the set of i such that $\Phi_i(x) = \mathsf{tp}^{cl}_{\mathcal{K}}(\mathfrak{A}, b_i^{\mathfrak{A}})$ is not connected. If $X = \{1, \ldots, n\}$, then $\neg \bigwedge_{i \in X} \Phi_i(x_i)$ separates $(\mathcal{K}, E^+, \{\mathbf{b}\})$ (as condition 1 holds), which concludes the proof. Otherwise, let $\mathfrak{A}_i$, $i \in X$, be the maximal connected components of $\mathfrak{A}$ containing the singleton $b_i^{\mathfrak{A}}$. Our aim is to show that there exists a variant $\mathfrak{C}$ of $\mathfrak{A}$ and a sufficiently large ℓ such that $(\mathcal{D}_{\mathbf{a}}, \mathbf{a}) \not\preceq^{\ell}_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} (\mathfrak{C}, \mathbf{b}^{\mathfrak{A}})$. We partition the remaining part of $\mathfrak{A}$ without $(\mathfrak{A}_i)_{i \in X}$ into components as follows. Define an equivalence relation E on the class of $\mathsf{cl}(\mathcal{K})$-types $\Phi(x)$ with one free variable x such that $(\Phi(x), \Psi(x)) \in E$ iff there exists $\mathfrak{A} \models \mathcal{K}$ and nodes a, b in $\mathsf{dom}(\mathfrak{A})$ such that a, b are in the same connected component in $\mathfrak{A}$ and a and b realize Φ and Ψ, respectively. Let $\mathfrak{A}'$ and $\{\mathfrak{C} \mid \mathfrak{C} \in K\}$ be the maximal components of $\mathfrak{A}$ without $\{b_i^{\mathfrak{A}} \mid i \in X\}$ such that

1. all nodes in any $\mathfrak{C}$ are connected to a node in $\{c^{\mathfrak{A}} \mid c \in \mathsf{dom}(\mathcal{D})\}$;
2. all $\mathsf{cl}(\mathcal{K})$-types $\Phi(x)$ realized in a same $\mathfrak{C}$ are E-equivalent;
3. no node in $\mathfrak{A}'$ is connected to a node in $\{c^{\mathfrak{A}} \mid c \in \mathsf{dom}(\mathcal{D})\}$.

Observe that $\mathfrak{A}$ is the disjoint union of $\mathfrak{A}_i$, $i \in X$, $\mathfrak{A}'$, and the models in K. Let $\mathfrak{C} \in K$ and let $\mathcal{D}'$ be the restriction of $\mathcal{D}$ to the constants $c \in \mathsf{cons}(\mathcal{D})$ such that $c^{\mathfrak{A}} \in \mathsf{dom}(\mathfrak{C})$. Let I_0 be the set of i with $b_i^{\mathfrak{A}} \in \mathsf{dom}(\mathfrak{C})$. We aim to construct a model $\mathfrak{C}$ of $(\mathcal{O}, \mathcal{D}')$ such that

$$(*) \qquad (\mathcal{D}_{\mathbf{a}_{I_0}}, \mathbf{a}_{I_0}) \not\to (\mathfrak{A}, \mathbf{b}_{I_0}^{\mathfrak{A}}) \text{ implies } \exists \ell \geqslant 0 : (\mathcal{D}_{\mathbf{a}_{I_0}}, \mathbf{a}_{I_0}) \not\preceq^{\ell}_{\mathbf{oGF},\Sigma} (\mathfrak{C}, \mathbf{b}_{I_0}^{\mathfrak{A}}).$$

For any model $\mathfrak{C}$ of $\mathcal{D}'$ and $d \in \mathsf{dom}(\mathfrak{C})$ we let the distance $\mathsf{dist}_{\mathfrak{C}}(\mathcal{D}', d) = \ell$ iff ℓ is minimal such $\mathsf{dist}(c^{\mathfrak{C}}, d) \leqslant \ell$ for at least one $c \in \mathsf{cons}(\mathcal{D}')$. We denote by $\mathfrak{C}^{\leqslant \ell}_{\mathcal{D}'}$ the submodel of $\mathfrak{C}$ induced by the set of nodes d in $\mathfrak{C}$ with $\mathsf{dist}_{\mathfrak{C}}(\mathcal{D}', d) \leqslant \ell$. We construct for any $\ell \geqslant 0$ a model $\mathfrak{C}$ of $\mathcal{O}$ that coincides with $\mathfrak{C}$ on $\{c^{\mathfrak{C}} \mid c \in$

$\mathsf{dom}(\mathcal{D}')\}$ such that $\mathfrak{C}^{\leqslant \ell}_{\mathcal{D}'}$ is finite and there exists $\ell' \geqslant \ell$ with

(a) $(\mathfrak{C}^{\leqslant \ell}_{\mathcal{D}'}, \mathbf{b}^{\mathfrak{A}}_{I_0}) \to (\mathfrak{A}, \mathbf{b}^{\mathfrak{A}}_{I_0})$;
(b) for any two distinct $d_1, d_2 \in \mathsf{dom}(\mathfrak{C}^{\leqslant \ell}_{\mathcal{D}'})$, $(\mathfrak{C}, d_1) \not\sim^{\ell'}_{\mathbf{oGF},\Sigma} (\mathfrak{C}, d_2)$.

We first show that $(*)$ follows. Assume ℓ' is such that (b) holds. Let $\ell'' = \ell' + |\mathcal{D}|$ and $\ell \geqslant |\mathcal{D}|$. Assume that $(\mathcal{D}_{\mathbf{a}_{I_0}}, \mathbf{a}_{I_0}) \preceq^{\ell''}_{\mathbf{oGF},\Sigma} (\mathfrak{C}, \mathbf{b}^{\mathfrak{A}}_{I_0})$ and $(\mathcal{D}_{\mathbf{a}_{I_0}}, \mathbf{a}_{I_0}) \not\to (\mathfrak{A}, \mathbf{b}^{\mathfrak{A}}_{I_0})$. By Condition (a), $(\mathcal{D}_{\mathbf{a}_{I_0}}, \mathbf{a}_{I_0}) \not\to (\mathfrak{C}^{\leqslant \ell''}_{\mathcal{D}'}, \mathbf{b}^{\mathfrak{C}}_{I_0})$. By Lemma 2.98, there exist d, d' with $d \neq d'$ in $\mathfrak{C}^{\leqslant |\mathcal{D}|}_{\mathcal{D}'}$ such that $(\mathfrak{C}, d) \sim^{\ell'}_{\mathbf{oGF},\Sigma} (\mathfrak{C}, d')$ and Condition (b) is contradicted.

Assume $\ell \geqslant 0$ is given. To construct $\mathfrak{C}$, tet $T_{\mathfrak{E}}$ be the set of $\mathsf{cl}(\mathcal{K})$-types $\Phi(x)$ that are E-equivalent to some $\mathsf{cl}(\mathcal{K})$-type realized in $\mathfrak{E}$. Observe that $T_{\mathfrak{E}}$ is an equivalence class for the relation E, by construction of $\mathfrak{E}$. By definition, no $\mathsf{cl}(\mathcal{K})$-type in $T_{\mathfrak{E}}$ is **oGF**-complete. As they are all E-equivalent, there exists a sequence $\sigma = \Phi^{\sigma}_0, \Phi^{\sigma}_1, \Phi^{\sigma}_2$ such that for any $\mathsf{cl}(\mathcal{K})$-type $\Phi(x) \in T_{\mathfrak{E}}$ we find a sequence witnessing **oGF**-incompleteness of $\Phi(x)$ that ends with σ: choose one such sequence for one of the types, then for any other type, simply walk to the former type and then follow that sequence. The concatenation creates a witnessing sequence. As a first step of the construction of $\mathfrak{C}$, we define a model $\mathfrak{B}$ of $\mathcal{K}$ by repeatedly forming the partial unfolding of $\mathfrak{E}$ so that

(path) from any $f_0 \in \mathfrak{B}^{\leqslant \ell}_{\mathcal{D}'}$ there exists a strict path $R^{f_0}_1(\mathbf{d}_1), \dots, R^{f_0}_k(\mathbf{d}_k)$ from f_0 to some f_1 such that $\mathsf{dist}_{\mathfrak{B}}(\mathcal{D}', f_1) = \ell$.

For the construction of $\mathfrak{B}$, let $\mathfrak{B}_0 = \mathfrak{A}$ and include all $d \in \mathsf{dom}(\mathfrak{A}^{\leqslant \ell}_{\mathcal{D}'})$ into the frontier F_0. Assume $\mathfrak{B}_i$ and frontier F_i have been constructed. If F_i is empty, we are done and set $\mathfrak{B} = \mathfrak{B}_i$. Otherwise take $d \in F_i$ and let $d' \neq d$ be any element contained in a joint guarded set with d in $\mathfrak{B}_i$. Assume $k = \mathsf{dist}_{\mathfrak{B}_i}(\mathcal{D}', d)$. Then let $\mathfrak{B}_{i+1}$ be the partial unfolding $(\mathfrak{B}_i)_{\mathbf{d}}$ of $\mathfrak{B}_i$ for the tuple $\mathbf{d} = (d, d', d, d', \dots)$ of length $\ell - k$, and obtain F_{i+1} by removing d from F_i and adding all new nodes in $\mathsf{dom}((\mathfrak{B}_{i+1})^{\leqslant \ell}_{\mathcal{D}'})$. Clearly this construction terminates after finitely many steps and (path) holds, see 2.83.

Let L denote the set of all d in $\mathfrak{B}$ with $\mathsf{dist}_{\mathfrak{B}}(\mathcal{D}', d) = \ell$ and let L' denote the set of all $\mathbf{d}$ of arity $\geqslant 2$ in $\mathfrak{B}$ such that there exist R with $\mathfrak{B} \models R(\mathbf{d})$ and $d \in [\mathbf{d}]$ with $\mathsf{dist}_{\mathfrak{B}}(\mathcal{D}', d) = \ell$. We obtain $\mathfrak{C}$ by keeping $\mathfrak{B}^{\leqslant \ell}_{\mathcal{D}'}$ and the guarded sets that intersect with it and attaching to every $d \in L$ and $\mathbf{d} \in L'$ guarded tree decomposable $\mathfrak{F}_d$ and $\mathfrak{F}'_{\mathbf{d}}$ such that in the resulting model no d in L is guarded Σ ℓ'-bisimilar to any other d' in $\mathfrak{B}^{\leqslant \ell}_{\mathcal{D}'}$ for a sufficiently large ℓ'. It then directly follows that $\mathfrak{C}$ satisfies Conditions (a) and (b).

The construction of $\mathfrak{F}'_{\mathbf{d}}$ is straightforward. Fix $\mathbf{d} \in L'$. Let $\Phi'_0 := \mathsf{tp}^{cl}_{\mathcal{K}}(\mathfrak{B}, \mathbf{d})$. Then $\mathfrak{F}'_{\mathbf{d}}$ is defined as the tree decomposible model $\mathfrak{A}'_r$ of $\mathcal{O}$ with tree decompositon (T', E', bag') and root r such that $\mathfrak{A}'_r \models \Phi'_0(\mathbf{d})$ and $\mathsf{bag}(r) = [\mathbf{d}]$ and for every cl($\mathcal{K}$)-type $\Psi_1(\mathbf{x}_1)$ realized by some $\mathbf{c}$ with $[\mathbf{c}] = \text{bag}(t)$ and $\mathcal{K}$ type $\Psi_2(\mathbf{x}_2)$ coherent with $\Psi_1(\mathbf{x}_1)$ there exists a successor t' of t in T such that $\Psi_1(\mathbf{x}_1) \cup \Psi_2(\mathbf{x}_2)$ is realized in $\text{bag}(t) \cup \text{bag}(t')$ under an assignment μ of the variables $[\mathbf{x}_1] \cup [\mathbf{x}_2]$ such that $\mu(\mathbf{x}_1) = \mathbf{c}_1$. The only property of $\mathfrak{F}'_\mathbf{d}$ we need is, $\mathbf{x}_1$ being the variables in Φ_1^σ and $\mathbf{x}'$ the variables in Φ_2 that are not in Φ_1, that $\mathfrak{F}'_\mathbf{d} \models \forall \mathbf{x}_1(\Phi_1^\sigma \rightarrow \exists \mathbf{x}' \Phi_2^\sigma)$.

The construction of $\mathfrak{F}_d$ is more involved. Let $L_\mathcal{K} = 2^{2^{||\mathcal{K}||}} + 1$ and take for any $d \in L$ a number N_d such that $N_d > |\mathfrak{B}_{\mathcal{D}'}^{\leqslant \ell+1}| + 2(L_\mathcal{K} + 1)$ and $|N_d - N_{d'}| > 2(L_\mathcal{K} + 1)$ for $d \neq d'$. Fix $d \in L$ and let $\Phi_0(x) = \text{tp}_\mathcal{K}^{cl}(\mathfrak{A}, d)$. Then $\Phi_0(x) \in T_\mathfrak{C}$ and there exists a sequence $\Phi_0(\mathbf{x}_0), \dots, \Phi_{n_d}(\mathbf{x}_{n_d}), \Phi_{n_d+1}(\mathbf{x}_{n_d+1})$ that witnesses **oGF**-incompleteness of $\Phi_0(x_0)$ and ends with $\Phi_0^\sigma \Phi_1^\sigma \Phi_2^\sigma$. By Lemma 2.73 we may assume that $1 \leqslant n_d \leqslant L_\mathcal{K} + 1$. Let $\Psi(x) := \exists \Sigma^{L_\mathcal{K}+1}.(\Phi_1^\sigma \wedge \neg \exists \mathbf{x}' \Phi_2^\sigma)$, where $\exists \Sigma^k.\chi$ stands for the disjunction of all **oGF**-formulas stating that there exists a path from x along relations in Σ of length at most k to a tuple where χ holds. To construct $\mathfrak{F}_d$ consider the tree decomposable model $\mathfrak{A}_r$ of $\mathcal{O}$ with tree decomposition (T, E, bag) and root r such that $\mathfrak{A}_r \models \Phi_0(c_0)$ for some constant c_0 with $\text{bag}(r) = \{c_0\}$ and for every cl($\mathcal{K}$)-type $\Psi_1(\mathbf{x}_1)$ realized by some $\mathbf{c}$ with $[\mathbf{c}] = \text{bag}(t)$ and $\mathcal{K}$ type $\Psi_2(\mathbf{x}_2)$ coherent with $\Psi_1(\mathbf{x}_1)$ there exists a successor t' of t in T such that $\Psi_1(\mathbf{x}_1) \cup \Psi_2(\mathbf{x}_2)$ is realized in $\text{bag}(t) \cup \text{bag}(t')$ under an assignment μ of the variables $[\mathbf{x}_1] \cup [\mathbf{x}_2]$ such that $\mu(\mathbf{x}_1) = \mathbf{c}_1$, except if $\Psi_1 \wedge \neg \exists \mathbf{x}' \Psi_2$ (with $\mathbf{x}'$ the sequence of variables in $\mathbf{x}_2$ which are not in $\mathbf{x}_1$) is equivalent to $\Phi_1^\sigma \wedge \neg \exists \mathbf{x}' \Phi_2^\sigma$ and $\text{dist}_{\mathfrak{A}_r}(\text{bag}(t), \text{bag}(r)) \leqslant N_d + L_\mathcal{K} + 1$. Observe that

$$\begin{aligned} &\mathfrak{A}_r \models \Psi(e) && \text{for all } e \text{ with } \text{dist}_{\mathfrak{A}_r}(c_0, e) \leqslant N_d \\ &\mathfrak{A}_r \models \neg \Psi(e) && \text{for all } e \text{ with } \text{dist}_{\mathfrak{A}_r}(c_0, e) > N_d + 2(L_\mathcal{K} + 1). \end{aligned}$$

Moreover, $\mathfrak{A}_r$ contains a strict path $R_1(\mathbf{e}_1), \dots, R_{n_d}(\mathbf{e}_{n_d}), \dots, R_{n_d}(\mathbf{e}_{n_d+2N_d})$ from $e_0 \in [\mathbf{e}_1]$ to $c_0 \in [\mathbf{e}_{n_d+2N_d}]$ such that $\Phi_0(x)$ is realized in e_0. Then $\mathfrak{F}_d$ is obtained from $\mathfrak{A}_r$ by renaming e_0 to d.

Finally, $\mathfrak{C}$ is obtained by hooking $\mathfrak{F}_d$ at d to $\mathfrak{B}_{\mathcal{D}'}^{\leqslant \ell}$ for all $d \in L$.

$\mathfrak{C}$ is a model of $\mathcal{K}$ since $\Phi_0(x)$ is realized in e_0 and d. Moreover, it clearly satisfies Condition (a). For Condition (b) assume $d \in L$ is as above. Let $\varphi_d(x) = \forall \Sigma^{N_d}.\Psi$ where $\forall \Sigma^k.\chi$ stands for $\neg \exists \Sigma^k.\neg \chi$. Then $\mathfrak{C} \models \varphi_d(c_0)$ and by construction no node that is not in $\text{dom}(\mathfrak{F}_d)$ satisfies φ_d. Condition (b) now follows from the fact that there exists a path from d to a node satisfying φ_d that is shorter than any such path in $\mathfrak{C}$ from any other node in $\mathfrak{B}_{\mathcal{D}'}^{\leqslant \ell}$ to a node satisfying φ_d. Then, $(*)$ is proved.

Finally, it only remains to prove that if $(*)$ holds, then for appropriately defined $\mathfrak{C}$, all $\mathbf{a} \in E^+$, and sufficiently large ℓ, $(\mathcal{D}_\mathbf{a}, \mathbf{a}) \not\preceq_{\mathbf{oGF},\Sigma}^\ell (\mathfrak{C}, \mathbf{b}^\mathfrak{A})$, which would conclude the proof. Let $\mathbf{a} \in E^+$ be fixed. If $(\mathcal{D}_{\mathbf{a}_{I_0}}, \mathbf{a}_{I_0}) \not\to (\mathfrak{A}, \mathbf{b}^{\mathfrak{A}}_{I_0})$ for some I_0 associated to some $\mathfrak{C} \in K$, then, $(\mathcal{D}_{\mathbf{a}_{I_0}}, \mathbf{a}_{I_0}) \not\preceq^{\ell}_{\mathbf{oGF},\Sigma} (\mathfrak{C}, \mathbf{b}^{\mathfrak{C}}_{I_0})$ for some ℓ by $(*)$, so $(\mathcal{D}_{\mathbf{a}}, \mathbf{a}) \not\preceq^{\ell}_{\mathbf{oGF},\Sigma} (\mathfrak{C}, \mathbf{b}^{\mathfrak{A}})$ for some ℓ. Now assume the contrary, *i.e.* that $(\mathcal{D}_{\mathbf{a}_{I_0}}, \mathbf{a}_{I_0}) \to (\mathfrak{A}, \mathbf{b}^{\mathfrak{A}}_{I_0})$ for all I_0 associated with any $\mathfrak{C} \in K$. We know that $(\mathcal{D}_{\mathbf{a}}, \mathbf{a}) \not\to (\mathfrak{A}, \mathbf{b}^{\mathfrak{A}})$. One of those conditions hold.

1. $(\mathcal{D}_{a_i}, a_i) \not\preceq^{\ell}_{\mathbf{oGF},\Sigma} (\mathfrak{C}, b^{\mathfrak{C}}_i)$ for some $i \in X$.
2. Some a_i, a_j with $i \neq j$ and $i, j \in X$ are connected in $\mathcal{D}$.
3. Some a_i, $i \in X$ and $a \in [\mathbf{a}_{I_0}]$ with I_0 linked to some $\mathfrak{C} \in K$ are connected in $\mathcal{D}$.
4. Some $a \in [\mathbf{a}_{I_0}]$ and $a' \in [\mathbf{a}_{I'_0}]$ with I_0, I'_0 linked to distinct $\mathfrak{C} \in K$ are connected in $\mathcal{D}$.

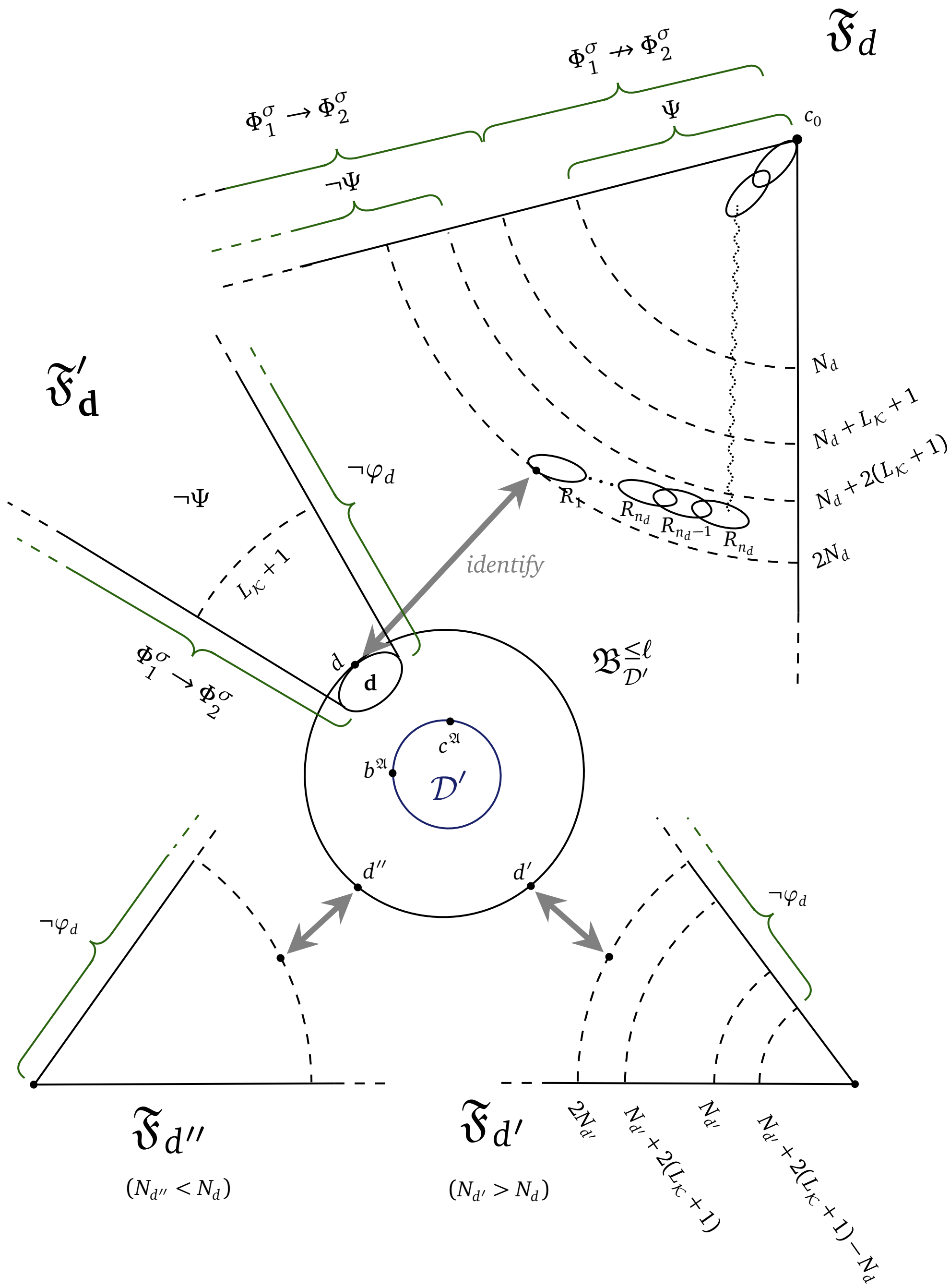
In all these cases it follows that $(\mathcal{D}_{\mathbf{a}}, \mathbf{a}) \not\preceq^{\ell}_{\mathbf{oGF},\Sigma} (\mathfrak{C}, \mathbf{b}^{\mathfrak{A}})$, for sufficiently large ℓ. ⊣

2.100. As in the $\mathcal{ALCI}$ case, we immediately get the following corollary, as well as a polynomial reduction from projective to non-projective separability, giving us complexity bounds.

2.101. Corollary. $\mathsf{sep}^p_w(\mathbf{GF}, \mathbf{GF}) = \mathsf{sep}^p_w(\mathbf{GF}, \mathcal{L})$ *for any* **FO**-*fragment* $\mathcal{L} \supseteq$ **UCQ**.

2.102. Corollary. *Weak projective and non-projective* (**GF**, **GF**)*-separability are* 2EXP*-complete in combined complexity.*

Proof. We then only need to show this for projective separability. The lower bound is immediate from satisfiability of **GF**-knowledge bases [G99]. The upper bound is immediate from the complexity of UCQ-evaluation on **GF**-knowledge bases [BGO14]. ⊣



Construction of $\mathfrak{C}$ in the proof of Theorem 2.99.

§ 2.6. Undecidability for $(\mathcal{ALC}, \mathcal{EL}(\mathcal{I}))$

[BKLRWZ19] showed undecidability of the following problem for $\mathcal{L} \in \{\mathcal{EL}, \mathcal{ELI}\}$, in which we replaced tree-shaped CQs by $\mathcal{EL}(\mathcal{I})$-concepts.

- Given $\mathcal{ALC}$-knowledge bases $\mathcal{K}_1 = (\mathcal{O}_1, \mathcal{D}), \mathcal{K}_2 = (\mathcal{O}_2, \mathcal{D})$ such that $\mathsf{cons}(\mathcal{D}) = \{a\}$, does $\mathcal{K}_1 \models C(a)$ imply $\mathcal{K}_2 \models C(a)$ for any $\mathcal{L}$-concept C?

We can reduce that problem to weak full $(\mathcal{ALC}, \mathcal{EL}(\mathcal{I}))$-separability to obtain the following.

2.103. Theorem. *Let $\mathcal{L}_O \supseteq \mathcal{ALC}$ and $\mathcal{L}_S \in \{\mathcal{EL}, \mathcal{ELI}\}$. Then weak full $(\mathcal{L}_O, \mathcal{L}_S)$-separability is undecidable.*

For the reduction, let $\mathcal{K}_1 = (\mathcal{O}_1, \mathcal{D})$ and $\mathcal{K}_2 = (\mathcal{O}_2, \mathcal{D})$ be $\mathcal{ALC}$-knowledge bases such that $\mathsf{cons}(\mathcal{D}) = \{a\}$. Let A_1, A_2, B be fresh concept names and a_1, a_2, b be fresh constants. Let $\mathcal{D}_{a_1}, \mathcal{D}_{a_2}, \mathcal{D}_b$ be obtained by replacing a in $\mathcal{D}$ by a_1, a_2, b respectively. Let $\mathcal{K}_0 = (\mathcal{O}_0, \mathcal{D}_0)$ where $\mathcal{O}_0 = (\mathcal{O}_1)_{|A_1} \cup (\mathcal{O}_1)_{|A_2} \cup (\mathcal{O}_2)_{|B}$ and $\mathcal{D}_0 = \mathcal{D}_{a_1} \cup \mathcal{D}_{a_2} \cup \mathcal{D}_b \cup \{A_1(a_1), A_2(a_2), B(b)\}$. The following lemmas then hold.

2.104. Lemma. *The following hold for any $\mathcal{ELI}$ concept C.*

If $\mathcal{K}_0 \models C(a_1)$, then C does not contain A_2 or B.
If $\mathcal{K}_0 \models C(a_2)$, then C does not contain A_1 or B.
If $\mathcal{K}_0 \models C(b)$, then C does not contain A_1 or A_2.

Proof. All 3 proofs are similar so we only prove the first statement. Let $\mathfrak{A}_{A,1} \models ((\mathcal{O}_1)_{|A_1}, \mathcal{D}_{a_1})$, $\mathfrak{A}_{A,2} \models ((\mathcal{O}_1)_{|A_2}, \mathcal{D}_{a_2})$ and $\mathfrak{A}_B \models ((\mathcal{O}_2)_{|B}, \mathcal{D}_b)$. One can assume w.l.o.g. that $a_2, b \notin \mathsf{dom}(\mathfrak{A}_{A,1}), a_1, b \notin \mathsf{dom}(\mathfrak{A}_{A,2})$ and $a_1, a_2 \notin \mathsf{dom}(\mathfrak{A}_B)$. One can also assume $A_2^{\mathfrak{A}_{A,1}} = B^{\mathfrak{A}_{A,1}} = A_1^{\mathfrak{A}_{A,2}} = B^{\mathfrak{A}_{A,2}} = A_1^{\mathfrak{A}_B} = A_2^{\mathfrak{A}_B} = \emptyset$. Let $\mathfrak{A}_0 = \mathfrak{A}_{A,1} \uplus \mathfrak{A}_{A,2} \uplus \mathfrak{A}_B$. Then $\mathfrak{A}_0 \models \mathcal{K}_0$ and $\mathfrak{A}_0 \not\models C(a_1)$ if A_2 or B occurs in C, because no node in $A_2^{\mathfrak{A}_0}$ or $B^{\mathfrak{A}_0}$ is reachable from a_1 in $\mathfrak{A}_0$ and C is an $\mathcal{ELI}$-concept. ⊣

The following lemma is straightforwardly checked.

2.105. Lemma. *The following hold for any $\mathcal{ELI}$ concept C.*

$$\mathcal{K}_0 \models C(a_1) \iff ((\mathcal{O}_1)_{|A_1}, \mathcal{D}_{a_1}) \models C(a_1)$$
$$\mathcal{K}_0 \models C(a_2) \iff ((\mathcal{O}_1)_{|A_2}, \mathcal{D}_{a_2}) \models C(a_2)$$
$$\mathcal{K}_0 \models C(b) \iff ((\mathcal{O}_2)_{|B}, \mathcal{D}_b) \models C(b)$$

$$A_1 \notin \mathsf{sig}(C) \;\Rightarrow\; \big(\, \mathcal{K}_1 \models C(a) \;\Leftrightarrow\; ((\mathcal{O}_1)_{|A_1}, \{A(a_1)\}) \models C(a_1) \,\big)$$
$$A_2 \notin \mathsf{sig}(C) \;\Rightarrow\; \big(\, \mathcal{K}_1 \models C(a) \;\Leftrightarrow\; ((\mathcal{O}_1)_{|A_2}, \{A(a_2)\}) \models C(a_2) \,\big)$$
$$B \notin \mathsf{sig}(C) \;\Rightarrow\; \big(\, \mathcal{K}_2 \models C(a) \;\Leftrightarrow\; ((\mathcal{O}_2)_{|B}, \{A(b)\}) \models C(b) \,\big)$$

Proof of Theorem 2.103. We now show that $(\mathcal{K}_0, \{a_1, a_2\}, \{b\})$ is weakly $\mathcal{L}$-separable iff there exists an $\mathcal{L}$-concept C such that $\mathcal{K}_1 \models C(a)$ and $\mathcal{K}_2 \not\models C(a)$. Suppose $(\mathcal{K}_0, \{a_1, a_2\}, \{b\})$ is weakly separated by some $\mathcal{L}$-concept C. By 2.104, C does not contain A_1, A_2, B. By 2.105, $\mathcal{K}_1 \models C(a)$ and $\mathcal{K}_2 \not\models C(a)$. Conversely, suppose $\mathcal{K}_1 \models C(a)$ and $\mathcal{K}_2 \not\models C(a)$. Since A_1, A_2, B do not occur in $\mathcal{K}_1$, they do not occur in C. Then 2.105 yields $\mathcal{K}_0 \models C(a_1)$, $\mathcal{K}_0 \models C(a_2)$ and $\mathcal{K}_0 \not\models C(b)$ from $\mathcal{K}_2 \not\models C(a)$. ⊣

If the ontology language is reduced, we can reach decidability. [FJLPW19] shows that full weak $(\mathcal{EL}(\mathcal{I}), \mathcal{EL})$-separability is Exp-complete in both combined and data complexity, and full weak $(\mathcal{ELI}, \mathcal{ELI})$-separability is undecidable. The former is proved using a model-theoretic characterisation based on universal models and simulations. The latter is proved by a reduction from a tiling problem, inspired by the proof in [BKLRWZ19] that CQ-entailment between $\mathcal{ALC}$-knowledge bases is undecidable.

Chapter **3**

Full strong separability

We now switch from weak to strong separability, while still assuming full signature. Recall there is no distinction between projective and non-projective separability in the strong case (Prop. 1.60). As in the weak case, it appears that **UCQ** matches **FO**'s separating power on any labeled **FO**-knowledge base and the same UCQ witnesses it, although the proof's essence is different. We establish that equivalence by model-theoretically characterising the problem. That characterisation induces a crucial reduction from full strong separability to satisfiability, in contrast to the one from full weak separability to query evaluation. It is in fact even more fruitful than its weak counterpart: it implies that $(\mathcal{L}, \mathcal{L})$-separability coincides with $(\mathcal{L}, \mathbf{FO})$-separability for $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}, \mathbf{GF}, \mathbf{FO}^2\}$, whereas this was not true for $\mathcal{ALC}$ and $\mathbf{FO}^2$ in the weak case. Explicit separating formulas also follow in each case.

Using the connection with satisfiability, we obtain the following completeness results for full signature strong $(\mathcal{L}_O, \mathcal{L}_S)$-separability where $\mathcal{L}_O = \mathcal{L}_S$ is displayed in the left column.

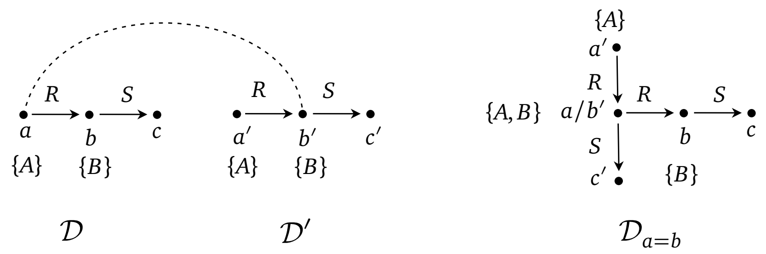
	Combined	Data
$\emptyset$	Never separable	
$\mathcal{EL}$	Never separable	
$\mathcal{ELI}$	Never separable	
$\mathcal{ALC}$	Exp	coNP
$\mathcal{ALCI}$	Exp	coNP
GF	2Exp	coNP
$\mathbf{FO}^2$	NExp	coNP
FO	Undecidable	

§ 3.1. Reduction to satisfiability with FO

We characterize strong $(\mathbf{FO}, \mathbf{FO})$-separability in terms of knowledge base unsatisfiability and show that strong $(\mathbf{FO}, \mathbf{FO})$-separability coincides with strong $(\mathbf{FO}, \mathbf{UCQ})$-separability. Let $\mathcal{D}$ be a database and let $\mathbf{a} = (a_1, \ldots, a_n)$ and $\mathbf{b} = (b_1, \ldots, b_n)$ be tuples of constants in $\mathcal{D}$. We write $\mathcal{D}_{\mathbf{a}=\mathbf{b}}$ to denote the database obtained by taking $\mathcal{D} \cup \mathcal{D}'$, $\mathcal{D}'$ a disjoint copy of $\mathcal{D}$, and then identifying a_i with the copy b_i' of b_i for $1 \leqslant i \leqslant n$. For example,

$$\mathcal{D} = \{R(a,b), S(b,c), A(a), B(b)\}$$
$$\Rightarrow \quad \mathcal{D}_{a=b} = \{R(a,b), S(b,c), A(a), B(b), R(a',a), S(a,c'), A(a'), B(a)\}$$

where a', c' are 'copies' of a and c respectively and we identify the copy b' of b with a, as depicted below.



3.1. Theorem. *Let $(\mathcal{K}, E^+, E^-)$ be a labeled* **FO***-knowledge base, $\mathcal{K} = (\mathcal{O}, \mathcal{D})$. The following conditions are equivalent.*

1. *$(\mathcal{K}, E^+, E^-)$ is strongly* **UCQ***-separable;*
2. *$(\mathcal{K}, E^+, E^-)$ is strongly* **FO***-separable;*
3. *For all $\mathbf{a} \in E^+$ and $\mathbf{b} \in E^-$, the knowledge base $(\mathcal{O}, \mathcal{D}_{\mathbf{a}=\mathbf{b}})$ is unsatisfiable.*
4. *The UCQ $\bigvee_{\mathbf{a} \in E^+} q(\mathcal{D}_{\mathbf{a}}, \mathbf{a})$ strongly separates $(\mathcal{K}, E^+, E^-)$.*

Proof. (1) ⇒ (2), (2) ⇒ (3), and (4) ⇒ (1) are straightforward. It remains to prove (3) ⇒ (4). Thus assume that $\bigvee_{\mathbf{a} \in E^+} q(\mathcal{D}_{\mathbf{a}}, \mathbf{a})$ does not strongly separate $(\mathcal{K}, E^+, E^-)$. Then there exist a model $\mathfrak{A}$ of $\mathcal{K}$, $\mathbf{a} \in E^+$, and $\mathbf{b} \in E^-$ such that $\mathfrak{A} \models q(\mathcal{D}_{\mathbf{a}}, \mathbf{a})(\mathbf{b}^{\mathfrak{A}})$, witnessed by some homomorphism h. One can easily interpret the constants of $\mathcal{D}_{\mathbf{a}=\mathbf{b}}$ in such a way that $\mathfrak{A}$ becomes a model of $\mathcal{D}_{\mathbf{a}=\mathbf{b}}$: let $\mathfrak{A}'$ be a model of identical domain and interpretation of relations as $\mathfrak{A}$, and such that $(c')^{\mathfrak{A}'} = c^{\mathfrak{A}}$ and $c^{\mathfrak{A}'} = h(c)$ for all $c \in \mathsf{cons}(\mathcal{D})$. Then $\mathfrak{A}' \models \mathcal{D}_{\mathbf{a}=\mathbf{b}}$ and therefore $\mathfrak{A}' \models (\mathcal{O}, \mathcal{D}_{\mathbf{a}=\mathbf{b}})$. ⊣

Theorem 3.1 immediately implies Corollary 3.2, in the same way that Theorem 2.1 implies Corollary 2.2.

3.2. Corollary. $\mathsf{sep}_s(\mathbf{FO}, \mathcal{L}_S) = \mathsf{sep}_s(\mathbf{FO}, \mathbf{FO})$ *for any* **FO**-*fragment* $\mathcal{L}_S$ *containing* **UCQ**.

The link between separability and satisfiability established in Theorem 3.1 also provides complexity bounds for separability with any ontology language that contains **UCQ**, in particular **GNF**. Recall that **GNF** has a 2EXP-complete satisfiability problem [BCS15] in combined complexity and NP-complete in data complexity.

3.3. Corollary. *For any* **FO**-*fragment* $\mathcal{L}_S$ *that contains* **UCQ**, *strong* $(\mathbf{GNF}, \mathcal{L}_S)$-*separability is* 2EXP-*complete in combined complexity and* coNP-*complete in data complexity.*

We next study strong $(\mathcal{L}, \mathcal{L})$-separability for $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}, \mathbf{GF}, \mathbf{FO}^2\}$. In every case, we show that strong $(\mathcal{L}, \mathcal{L})$-separability coincides with strong $(\mathcal{L}, \mathbf{FO})$-separability. We can thus use the upper bound from KB unsatisfiability provided by Theorem 3.1 and the lower bound from Remark 1.63 to obtain complexity bounds.

§ 3.2. $\mathcal{ALC}(\mathcal{I})$-ONTOLOGIES

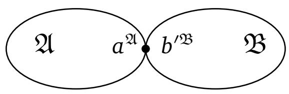
By characterizing strong $(\mathcal{ALC}, \mathcal{ALC})$-separability and $(\mathcal{ALCI}, \mathcal{ALCI})$-separability, we show they respectively coincide with strong $(\mathcal{ALC}, \mathbf{FO})$-separability and $(\mathcal{ALCI}, \mathbf{FO})$-separability (as in the weak case). Then, the reduction to unsatisfiability obtained in the previous section can also be applied here.

Let $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$.

3.4. Theorem. *For every labeled* $\mathcal{L}$-*knowledge base* $(\mathcal{K}, E^+, E^-)$, *the following conditions are equivalent.*

1. $(\mathcal{K}, E^+, E^-)$ *is strongly* $\mathcal{L}$-*separable.*
2. $(\mathcal{K}, E^+, E^-)$ *is strongly* **FO**-*separable.*
3. $\mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, a^{\mathfrak{A}}) \neq \mathsf{tp}_{\mathcal{K}}(\mathfrak{B}, b^{\mathfrak{B}})$ *for all* $a \in E^+, b \in E^-$ *and* $\mathfrak{A}, \mathfrak{B} \models \mathcal{K}$.
4. *The* $\mathcal{L}$-*concept* $(\bigsqcap t_1) \sqcup \cdots \sqcup (\bigsqcap t_n)$ *strongly separates* $(\mathcal{K}, E^+, E^-)$, $t_1, \ldots, t_n$ *being the* $\mathcal{K}$-*types realizable in* $(\mathcal{K}, a)$ *for some* $a \in E^+$.
5. $\mathsf{tp}_{\mathcal{O}}(\mathfrak{A}, a^{\mathfrak{A}}) \neq \mathsf{tp}_{\mathcal{O}}(\mathfrak{B}, b^{\mathfrak{B}})$ *for all* $a \in E^+, b \in E^-$ *and* $\mathfrak{A}, \mathfrak{B} \models \mathcal{K}$.
6. *The* $\mathcal{L}$-*concept* $(\bigsqcap t_1) \sqcup \cdots \sqcup (\bigsqcap t_m)$ *strongly separates* $(\mathcal{K}, E^+, E^-)$, $t_1, \ldots, t_m$ *being the* $\mathcal{O}$-*types realizable in* $(\mathcal{K}, a)$ *for some* $a \in E^+$.

Proof. (1) ⇒ (2), (3) ⇒ (4) (4) ⇒ (1), (5) ⇒ (3), (5) ⇒ (6) and (6) ⇒ (1) are straightforward. We prove (2) ⇒ (3) and (3) ⇒ (5). To show (2) ⇒ (3), let $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ and assume that (3) does not hold, *i.e.* there exist models $\mathfrak{A}$ and $\mathfrak{B}$ of $\mathcal{K}$ and $a \in E^+$, $b \in E^-$ such that $\mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, a^{\mathfrak{A}}) = \mathsf{tp}_{\mathcal{K}}(\mathfrak{B}, b^{\mathfrak{B}})$. We prove that $(\mathcal{O}, \mathcal{D}_{a=b})$ is satisfiable. This implies that $(\mathcal{K}, E^+, E^-)$ is not strongly **FO**-separable, by Theorem 3.1. With $(c')^{\mathfrak{B}} = c^{\mathfrak{B}}$ for all $c \in \mathsf{cons}(\mathcal{D})$, $\mathfrak{B}$ is a model of the database $\mathcal{D}'$ from the definition of $\mathcal{D}_{a=b}$. Define the model $\mathfrak{C}$ as $\mathfrak{A} \uplus \mathfrak{B}$ in which $a^{\mathfrak{A}}$ and $b'^{\mathfrak{B}}$ are identified, as depicted below. Using the fact that $\mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, a^{\mathfrak{A}}) = \mathsf{tp}_{\mathcal{K}}(\mathfrak{B}, b'^{\mathfrak{B}})$ and a simple induction on the structure of concepts C, it easily follows that for all $C \in \mathsf{sub}(\mathcal{O})$ and $d \in \mathsf{dom}(\mathfrak{C})$, $d \in C^{\mathfrak{C}} \iff d \in C^{\mathfrak{A}}$ if $d \in \mathsf{dom}(\mathfrak{A})$ and $d \in C^{\mathfrak{C}} \iff d \in C^{\mathfrak{B}}$ if $d \in \mathsf{dom}(\mathfrak{B})$. Then, $\mathfrak{C} \models \mathcal{O}$ follows from $\mathfrak{A}, \mathfrak{B} \models \mathcal{O}$. It is also clear from its definition that $\mathfrak{C} \models \mathcal{D}_{a=b}$. To show (3) ⇒ (5), suppose that ¬(5) is witnessed by some $\mathfrak{A}, \mathfrak{B}, a, b$ but (3) holds. Then $\mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, a^{\mathfrak{A}})$ and $\mathsf{tp}_{\mathcal{K}}(\mathfrak{B}, b^{\mathfrak{B}})$ can only differ with respect to some concept names $A_1, \ldots, A_n \in \mathsf{rel}_1$ that do not occur in $\mathcal{O}$. Assume w.l.o.g. that $a \in A_1^{\mathfrak{A}} \cap \cdots \cap A_n^{\mathfrak{A}}$ and $b \notin A_1^{\mathfrak{B}} \cup \cdots \cup A_n^{\mathfrak{B}}$. Then, the model $\mathfrak{B}'$ obtained from $\mathfrak{B}$ by adding b to $A_i^{\mathfrak{B}}$ for each $i \in \{1, \ldots, n\}$ is, just like $\mathfrak{B}$, a model of $\mathcal{K}$, since $A_1, \ldots, A_n$ do not occur in $\mathcal{O}$. Then, $\mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, a^{\mathfrak{A}}) = \mathsf{tp}_{\mathcal{K}}(\mathfrak{B}', b^{\mathfrak{B}'})$ *i.e.* (3) is negated. ⊣



Note that (6) of Theorem 3.4 provides concrete separating concepts. They are trivial and do not provide a good generalization of the examples. However, their size is at most $2^{p(||\mathcal{O}||)}$, p a polynomial, and does not depend on $\mathcal{D}$.

From Theorem 3.4 and Theorem 3.1 we immediately obtain an analogous result to Corollary 2.29.

3.5. Corollary. *For any* **FO**-*fragment* $\mathcal{L}_S \supseteq$ **UCQ**,

$$\mathsf{sep}_s(\mathcal{ALC}, \mathcal{ALC}) = \mathsf{sep}_s(\mathcal{ALC}, \mathcal{L}_S),$$
$$\mathsf{sep}_s(\mathcal{ALCI}, \mathcal{ALCI}) = \mathsf{sep}_s(\mathcal{ALCI}, \mathcal{L}_S).$$

As announced in the introduction, from the combination of Theorem 3.4 and equivalence (2) ⟺ (3) of Theorem 3.1 we also freely obtain complexity bounds from satisfiability. Recall that satisfiability of $\mathcal{L}$-knowledge bases is EXP-complete in combined complexity and NP-complete in data complexity for $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$.

3.6. Corollary. *For any* **FO**-*fragment* $\mathcal{L}_S \supseteq$ **UCQ** *and* $\mathcal{L}_O \in \{\mathcal{ALC}, \mathcal{ALCI}\}$, *full signature strong* $(\mathcal{L}_O, \mathcal{L}_S)$*-separability is* EXP*-complete in combined complexity and* CONP*-complete in data complexity.*

§ 3.3. GF-ONTOLOGIES

With a similar characterization to the $\mathcal{ALC}(\mathcal{I})$ one, we find that strong $(\mathbf{GF},\mathbf{GF})$-separability coincides with strong $(\mathbf{GF},\mathbf{FO})$-separability. Rigorously, the equivalence between (3) and (4) require two intermediate equivalences as in the $\mathcal{ALC}(\mathcal{I})$ case. We omit them, as their proof is the same.

Let $\mathcal{K}$-types be defined in this context as $\mathsf{cl}(\mathcal{K})$-types in the context of full signature $(\mathbf{GF},\mathbf{GF})$ weak separability, but simply from the set $\mathsf{sub}(\mathcal{K})$ (closure under subformulas and single negation of $\mathcal{K}$) instead of $\mathsf{cl}(\mathcal{K})$, *i.e.* without conditions (1) and (2) from Definition 2.93.

3.7. Theorem. *For every labeled* **GF***-knowledge base* $(\mathcal{K},E^+,E^-)$*, the following conditions are equivalent.*

1. $(\mathcal{K},E^+,E^-)$ *is strongly* **GF***-separable;*
2. $(\mathcal{K},E^+,E^-)$ *is strongly* **FO***-separable;*
3. $\mathsf{tp}_{\mathcal{K}}(\mathfrak{A},\mathbf{a}^{\mathfrak{A}}) \neq \mathsf{tp}_{\mathcal{K}}(\mathfrak{B},\mathbf{b}^{\mathfrak{B}})$ *for all* $\mathfrak{A},\mathfrak{B} \models \mathcal{K}$ *and* $\mathbf{a}\in E^+,\mathbf{b}\in E^-$*;*
4. *The* **GF***-formula* $(\bigwedge \Phi_1(\mathbf{x})) \vee \cdots \vee (\bigwedge \Phi_n(\mathbf{x}))$ *strongly separates* $(\mathcal{K},E^+,E^-)$*, with* $\Phi_1(\mathbf{x}),\ldots,\Phi_n(\mathbf{x})$ *being the* $\mathcal{O}$*-types realizable in* $(\mathcal{K},\mathbf{a})$ *for some* $\mathbf{a}\in E^+$.

Proof. Only (2) $\Rightarrow$ (3) is not trivial. It is proved similarly. We assume $\neg$(3), assume without loss of generality that witnessing models $\mathfrak{A},\mathfrak{B}$ are disjoint and $\mathfrak{B}\models\mathcal{D}'$, then construct a model $\mathfrak{C}$ by identifying $a_i^{\mathfrak{A}}$ with $b_i'^{\mathfrak{B}}$ for all i. It is then clear that $\mathfrak{C}\models\mathcal{D}_{\mathbf{a}=\mathbf{b}}$. It only remains to show $\mathfrak{C}\models\mathcal{O}$. One easily sees that, to show $\mathfrak{C}\models\mathcal{O}$, it suffices to show that for all $\mathbf{c}$ guarded in $\mathfrak{C}$ (note that, by definition, $[\mathbf{c}]\subseteq\mathsf{dom}(\mathfrak{A})$ or $[\mathbf{c}]\subseteq\mathsf{dom}(\mathfrak{B})$), $\mathsf{tp}_{\mathcal{K}}(\mathfrak{C},\mathbf{c}) = \mathsf{tp}_{\mathcal{K}}(\mathfrak{A},\mathbf{c})$ if $[\mathbf{c}]\subseteq\mathsf{dom}(\mathfrak{A})$ and $\mathsf{tp}_{\mathcal{K}}(\mathfrak{C},\mathbf{c}) = \mathsf{tp}_{\mathcal{K}}(\mathfrak{B},\mathbf{c})$ if $[\mathbf{c}]\subseteq\mathsf{dom}(\mathfrak{B})$. We prove it inductively. The atomic case is straightforward by definition of relations in $\mathfrak{C}$. The conjunction and negation cases are trivial. For the existential case, suppose without loss of generality $[\mathbf{c}]\subseteq\mathsf{dom}(\mathfrak{A})$ as the proof is identical in the case $[\mathbf{c}]\subseteq\mathsf{dom}(\mathfrak{B})$. Suppose $\mathfrak{A}\models\exists\mathbf{y}(R(\mathbf{c},\mathbf{y})\wedge\varphi(\mathbf{c},\mathbf{y}))$ for some $R\in\mathsf{rel}$ and $\varphi\in\mathsf{sub}(\mathcal{O})$. It immediately follows, by definition and induction hypothesis, that $\mathfrak{C}\models\exists\mathbf{y}(R(\mathbf{c},\mathbf{y})\wedge\varphi(\mathbf{c},\mathbf{y}))$. Conversely, suppose $\mathfrak{C}\models\exists\mathbf{y}(R(\mathbf{c},\mathbf{y})\wedge\varphi(\mathbf{c},\mathbf{y}))$ and that the existential quantifier is witnessed by some $\mathbf{c}'$ in $\mathfrak{C}$. By definition of $\mathfrak{C}$, $\mathfrak{C}\models R(\mathbf{c},\mathbf{c}')$ implies $[\mathbf{c}']\subseteq\mathsf{dom}(\mathfrak{A})$ or $[\mathbf{c}']\subseteq\mathsf{dom}(\mathfrak{B})$. If $[\mathbf{c}']\subseteq\mathsf{dom}(\mathfrak{A})$, then, by definition and induction hypothesis, $\mathfrak{A}\models R(\mathbf{c},\mathbf{c}')\wedge\varphi(\mathbf{c},\mathbf{c}')$. If $[\mathbf{c}']\subseteq\mathsf{dom}(\mathfrak{B})$, then $[\mathbf{c}]\subseteq[\mathbf{a}]=[\mathbf{b}]$ so $[\mathbf{c}]\subseteq\mathsf{dom}(\mathfrak{B})$. Then $\mathfrak{B}\models\exists\mathbf{y}(R(\mathbf{c},\mathbf{y})\wedge\varphi(\mathbf{c},\mathbf{y}))$. As we assumed $\mathsf{tp}_{\mathcal{K}}(\mathfrak{A},\mathbf{a}) = \mathsf{tp}_{\mathcal{K}}(\mathfrak{B},\mathbf{b})$ we have $\mathsf{tp}_{\mathcal{K}}(\mathfrak{A},\mathbf{c}) = \mathsf{tp}_{\mathcal{K}}(\mathfrak{B},\mathbf{c})$ thus $\mathfrak{A}\models\exists\mathbf{y}R(\mathbf{c},\mathbf{y})\wedge\varphi(\mathbf{c},\mathbf{y}))$. ⊣

In the same way we obtained Corollaries 3.5 and 3.6 for $\mathcal{ALC}(\mathcal{I})$, we obtain the following. Recall that deciding satisfiability of **GF**-knowledge bases is 2EXP-complete in combined complexity and NP-complete in data complexity.

3.8. Corollary. *Full signature strong* (**GF**, **GF**)*-separability coincides with strong* (**GF**, $\mathcal{L}$)*-separability for all* **FO***-fragments* $\mathcal{L} \supseteq$ **UCQ***. It is* 2EXP*-complete in combined complexity and* CONP*-complete in data complexity.*

We can formulate another characterization that has a counterpart in the full signature weak case (Thm. 2.90). It shows that, in the strong case too, separability with **GF** coincides with separability with the fragment **oGF** of **GF** (Def. 1.46). However, the arguments are different.

3.9. Theorem. *Let* $(\mathcal{K}, E^+, E^-)$ *be a labeled* **GF***-knowledge base. Then the following conditions are equivalent.*

1. $(\mathcal{K}, E^+, E^-)$ *is strongly* **oGF***-separable;*
2. $(\mathcal{K}, E^+, E^-)$ *is strongly* **GF***-separable;*
3. $(\mathfrak{A}, \mathbf{a}^{\mathfrak{A}}) \not\sim_{\mathbf{GF},\mathsf{sig}(\mathcal{K})} (\mathfrak{B}, \mathbf{b}^{\mathfrak{B}})$ *for all* $\mathfrak{A}, \mathfrak{B} \models \mathcal{K}, \mathbf{a} \in E^+, \mathbf{b} \in E^-$*;*
4. $(\mathfrak{A}, \mathbf{a}^{\mathfrak{A}}) \not\sim_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} (\mathfrak{B}, \mathbf{b}^{\mathfrak{B}})$ *for all* $\mathfrak{A}, \mathfrak{B} \models \mathcal{K}, \mathbf{a} \in E^+, \mathbf{b} \in E^-$.

Proof. The implications (1) ⇒ (2) and (4) ⇒ (3) are trivial. Moreover, (2) ⇒ (3) and (1) ⇒ (4) are immediate from Lemma 1.49. We prove below (3) ⇒ (2), (4) ⇒ (1), and (3) ⇒ (4). We start with the first implication; the proof of the second is analogous. Suppose $(\mathcal{K}, E^+, E^-)$ is not strongly **GF**-separable. Set the following.

$$\Gamma_+ := \{\varphi(\mathbf{x}) \in \mathbf{GF}(\mathsf{sig}(\mathcal{K})) \mid \forall \mathbf{a} \in E^+ : \mathcal{K} \models \varphi(\mathbf{a})\}$$
$$\Gamma_- := \{\varphi(\mathbf{x}) \in \mathbf{GF}(\mathsf{sig}(\mathcal{K})) \mid \forall \mathbf{a} \in E^- : \mathcal{K} \models \varphi(\mathbf{a})\}.$$

In what follows we use the fact that Γ_+ and Γ_- are closed under conjunction. We say that a set Γ of **GF** formulas is satisfiable in **a** w.r.t. a knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ if the extended (possibly infinite) knowledge base $\mathcal{K}' = (\mathcal{O}, \mathcal{D} \cup \{\varphi(\mathbf{a}) \mid \varphi(\mathbf{x}) \in \Gamma\})$ is satisfiable.

Claim 1. (1) There exists $\mathbf{a} \in E^+$ such that $\Gamma_+ \cup \Gamma_-$ is satisfiable in **a** w.r.t. $\mathcal{K}$.
(2) There exists $\mathbf{a} \in E^-$ such that $\Gamma_+ \cup \Gamma_-$ is satisfiable in **a** w.r.t. $\mathcal{K}$.

We prove (1), the proof of (2) is dual. Assume $\Gamma_+ \cup \Gamma_-$ is not satisfiable in any $\mathbf{a} \in E^+$ w.r.t. $\mathcal{K}$. Then Γ_- is not satisfiable in any $\mathbf{a} \in E^+$ w.r.t. $\mathcal{K}$. By compactness, there exist $\varphi_{\mathbf{a}}(\mathbf{x}) \in \Gamma_-$ such that $\mathcal{K} \models \neg\varphi_{\mathbf{a}}(\mathbf{a})$, for all $\mathbf{a} \in E^+$. We then get the desired contradiction, as $\mathcal{K} \models \neg(\bigwedge_{\mathbf{b} \in E^+} \varphi_{\mathbf{b}})(\mathbf{a})$ for all $\mathbf{a} \in E^+$

and $\mathcal{K} \models (\bigwedge_{\mathbf{b} \in E^+} \varphi_{\mathbf{b}})(\mathbf{a})$ for all $\mathbf{a} \in E^-$. Now, let $\Gamma_0 = \Gamma_+ \cup \Gamma_-$ and consider an enumeration $\varphi_1, \varphi_2, \ldots$ of all **GF**(sig($\mathcal{K}$)) formulas that do not belong to Γ_0. Then we inductively set $\Gamma_{i+1} = \Gamma_i \cup \{\varphi_{i+1}\}$ if there exist $\mathbf{a} \in E^+$ and $\mathbf{b} \in E^-$ such that $\Gamma_i \cup \{\varphi_{i+1}\}$ is satisfiable in both $\mathbf{a}$ and $\mathbf{b}$ w.r.t. $\mathcal{K}$. Set $\Gamma_{i+1} = \Gamma_i \cup \{\neg\varphi_{i+1}\}$, otherwise.

Claim 2. For all $i > 0$: there are $\mathbf{a} \in E^+$ and $\mathbf{b} \in E^-$ such that $\Gamma_i \cup \{\varphi_{i+1}\}$ is satisfiable in both $\mathbf{a}$ and $\mathbf{b}$ w.r.t. $\mathcal{K}$ or there are $\mathbf{a} \in E^+$ and $\mathbf{b} \in E^-$ such that $\Gamma_i \cup \{\neg\varphi_{i+1}\}$ is satisfiable in both $\mathbf{a}$ and $\mathbf{b}$ w.r.t. $\mathcal{K}$.

Assume Claim 2 has been proved for $i-1$. Let, without loss of generality, $\Gamma_i = \Gamma_+ \cup \Gamma_- \cup \{\varphi_1, \ldots, \varphi_i\}$. Assume Claim 2 does not hold for i. Then, again without loss of generality, there is no $\mathbf{a} \in E^+$ such that $\Gamma_i \cup \{\varphi_{i+1}\}$ is satisfiable in $\mathbf{a}$ w.r.t. $\mathcal{K}$ and there is no $\mathbf{b} \in E^-$ such that $\Gamma_i \cup \{\neg\varphi_{i+1}\}$ is satisfiable in $\mathbf{b}$ w.r.t. $\mathcal{K}$. By compactness, there exists $\varphi \in \Gamma_-$ such that $\mathcal{K} \models \varphi'(\mathbf{a})$ for all $\mathbf{a} \in E^+$, where $\varphi' = ((\varphi \sqcap \varphi_1 \sqcap \cdots \sqcap \varphi_i) \rightarrow \neg\varphi_{i+1})$. Then, by definition, $\varphi' \in \Gamma_+$. Then $\varphi' \in \Gamma_i$ so there is no $\mathbf{b} \in E^+$ such that Γ_i is satisfiable in $\mathbf{b}$ w.r.t. $\mathcal{K}$ - contradiction.

Let $\Gamma = \bigcup_{i \geqslant 0} \Gamma_i$. Then there exist models $\mathfrak{A}$ and $\mathfrak{B}$ of $\mathcal{K}$ and $\mathbf{a} \in E^+$ and $\mathbf{b} \in E^+$ such that $\mathfrak{A} \models \varphi(\mathbf{a})$ for all $\varphi \in \Gamma$ and $\mathfrak{B} \models \varphi(\mathbf{b})$ for all $\varphi \in \Gamma$. Thus, $(\mathfrak{A}, \mathbf{a}) \equiv_{\mathbf{GF}(\mathsf{sig}(\mathcal{K}))} (\mathfrak{B}, \mathbf{b})$. By Theorem 1.7 we may assume without loss of generality that $\mathfrak{A}$ and $\mathfrak{B}$ are ω-saturated. By Lemma 1.49, we obtain $(\mathfrak{A}, \mathbf{a}) \sim_{\mathbf{GF}, \mathsf{sig}(\mathcal{K})} (\mathfrak{B}, \mathbf{b})$, as required.

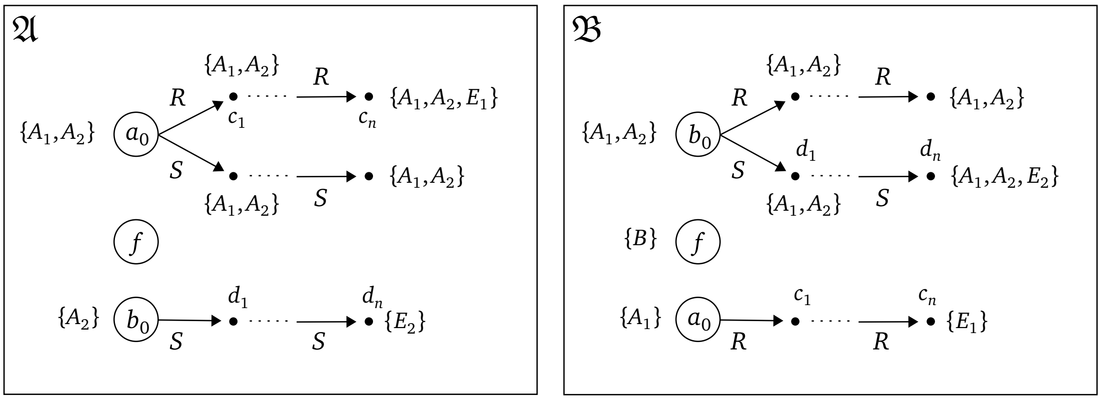
(3) $\Rightarrow$ (4). Suppose there are models $\mathfrak{A}$ and $\mathfrak{B}$ of $\mathcal{K}$ and $\mathbf{a} \in E^+, \mathbf{b} \in E^-$ such that $(\mathfrak{A}, \mathbf{a}^{\mathfrak{A}}) \sim_{\mathbf{oGF}, \mathsf{sig}(\mathcal{K})} (\mathfrak{B}, \mathbf{b}^{\mathfrak{B}})$. Obtain $\mathfrak{B}'$ by adding a disjoint copy of $\mathfrak{A}$ to the connected component of $\mathbf{b}^{\mathfrak{B}}$ in $\mathfrak{B}$. Clearly, $(\mathfrak{A}, \mathbf{a}^{\mathfrak{A}}) \sim_{\mathbf{oGF}, \mathsf{sig}(\mathcal{K})} (\mathfrak{B}', \mathbf{b}^{\mathfrak{B}'})$, as the witnessing connected guarded bisimulation can be extended to a guarded bisimulation by adding all partial isomorphisms between $\mathfrak{A}$ and its copy in $\mathfrak{B}'$. Then, $\mathfrak{B}' \models \mathcal{K}$ since $\mathfrak{B}' \models \mathcal{D}$ (clearly) and one can verify that $\mathfrak{A}$ and $\mathfrak{B}'$ satisfy the same **GF**(sig($\mathcal{K}$))-sentences to establish $\mathfrak{B}' \models \mathcal{O}$. It suffices to consider sentences of the form $\psi = \exists \mathbf{y}(\alpha(\mathbf{y}) \wedge \varphi(\mathbf{y}))$. We inductively assume that $(*)$ all subsentences of ψ are satisfied in $\mathfrak{A}$ iff they are satisfied in $\mathfrak{B}'$. Suppose first that ψ is satisfied in $\mathfrak{A}$. Since $\mathfrak{A}$ is a submodel of $\mathfrak{B}'$ and by $(*)$, ψ is also satisfied in $\mathfrak{B}'$. Conversely, assume that ψ is satisfied in $\mathfrak{B}'$ and let $\mathbf{c}$ be such that $\mathfrak{B}' \models \alpha(\mathbf{c}) \wedge \varphi(\mathbf{c})$. If $\mathbf{c}$ is in the copy of $\mathfrak{A}$ in $\mathfrak{B}'$, then ψ is also satisfied in $\mathfrak{A}$, due to $(*)$. If $\mathbf{c}$ is connected to $\mathbf{b}^{\mathfrak{B}}$, then $\mathfrak{A} \models \psi$ by bisimulation. $\dashv$

It follows from Theorem 3.7 that the minimal size of strongly separating **GF**-formulas is at most $2^{2^{p(||\mathcal{O}||)}}$, p a polynomial, and thus does not depend on the database. A variation shows that is not the case for separating **oGF**-formulas: as in the weak case (2.92), using **oGF** for separation instead of **GF** can come at the expense of much larger formulas.

3.10. *Example.* Let $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ be the **GF**-knowledge base (expressed in $\mathcal{ALC}$ with the universal role[1])

$$\mathcal{O} = \left\{ \begin{array}{l} A_1 \sqsubseteq \forall S.A_1 \\ A_2 \sqsubseteq \forall R.A_2 \\ E_2 \sqcap A_1 \sqsubseteq \exists u.B \\ E_1 \sqcap A_2 \sqsubseteq \neg\exists u.B \end{array} \right\} \qquad \mathcal{D} = \left\{ \begin{array}{l} A_1(a_0), R(a_0, c_1), \ldots, R(c_{n-1}, c_n), \\ E_1(c_n), A_2(b_0), E_2(c'_n), \\ S(b_0, c'_1), \ldots, S(c'_{n-1}, c'_n) \end{array} \right\}$$

In **GF** (in fact in $\mathcal{ALC}$ with the universal role) $(\mathcal{K}, \{a_0\}, \{b_0\})$ is strongly separated by the formula $(A_1 \sqcap A_2 \sqcap \neg\exists u.B) \sqcup (A_1 \sqcap \neg A_2)$. In contrast, any strongly separating formula in **oGF** has guarded quantifier rank at least n: there exist $\mathfrak{A}$, $\mathfrak{B} \models \mathcal{K}$ (depicted below) such that $(\mathfrak{A}, a_0^{\mathfrak{A}}) \sim^{n-1}_{\mathbf{oGF},\mathsf{sig}(\mathcal{K})} (\mathfrak{B}, b_0^{\mathfrak{B}})$.



§ 3.4. FO²-ONTOLOGIES

We show that unlike its weak counterpart, full signature strong $(\mathbf{FO}^2, \mathbf{FO}^2)$-separability is decidable and coincides with strong $(\mathbf{FO}^2, \mathbf{FO})$-separability. The proof strategy is the same as for $\mathcal{ALCI}$ and **GF** thus we first need a suitable notion of $\mathcal{K}$-type for $\mathbf{FO}^2$-KBs. Existing such notions, such as the ones defined in [GKV97], are not strong enough for our purposes, so we define and work with a more powerful notion. We can then once more establish a theorem that parallels Theorem 3.4 and show that strong separability has the same complexity as non-satisfiability of KBs, both in combined complexity (coNExp-complete) and in data complexity (coNP-complete).

We start by introducing appropriate types for $\mathbf{FO}^2$-KBs. Recall that we assume

[1] A role u such that $u^{\mathfrak{A}} = \mathrm{dom}(\mathfrak{A})^2$ for any model $\mathfrak{A}$, see [BHLS17] for example.

that $\mathbf{FO}^2$ uses unary and binary relation symbols only and that positive and negative examples are tuples of length $\leqslant 2$.

3.11. *Definition.* Assume that $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ is a $\mathbf{FO}^2$-knowledge base. Let $\mathsf{cl}_s(\mathcal{K})$ denote the union of the closure under single negation and swapping the variables x, y of the set of subformulas of $\mathcal{K}$ and $\{R(x,x),\ R(x,y),\ A(x) \mid R \in \mathsf{sig}(\mathcal{K}) \cap \mathsf{rel}_2,\ A \in \mathsf{sig}(\mathcal{K}) \cap \mathsf{rel}_1\}$. The $\mathcal{K}^1$-*type* of a pointed model $(\mathfrak{A}, a)$, denoted $\mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, a)$, is the set of all formulas $\psi(x) \in \mathsf{cl}_s(\mathcal{K})$ such that $\mathfrak{A} \models \psi(a)$. We denote by $T_x(\mathcal{K})$ the set of all $\mathcal{K}^1$-types. We say that $t(x) \in T_x(\mathcal{K})$ is *realized* in $(\mathfrak{A}, a)$ if $t(x) = \mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, a)$. Denote by $t(x)[y/x]$ the set of formulas obtained from $t(x)$ by swapping y and x. The $\mathcal{K}^2$-*type* of a pointed model $(\mathfrak{A}, a, b)$, denoted $\mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, a, b)$, is the set of all

$$
\begin{array}{ll}
R(x,y) & \text{if } \mathfrak{A} \models R(a,b) \\
R(y,x) & \text{if } \mathfrak{A} \models R(b,a) \\
\neg R(x,y) & \text{if } \mathfrak{A} \not\models R(a,b) \\
\neg R(y,x) & \text{if } \mathfrak{A} \not\models R(b,a) \\
x = y, y = x & \text{if } a = b \\
\neg(x = y), \neg(y = x) & \text{if } a \neq b
\end{array}
$$

where $R \in \mathsf{sig}(\mathcal{K})$. We denote by $T_{x,y}(\mathcal{K})$ the set of all $\mathcal{K}^2$-types. We say that $t(x,y) \in T_{x,y}(\mathcal{K})$ is *realized* in $(\mathfrak{A}, a, b)$ if $t(x,y) = \mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, a, b)$. For $t(x) \in T_x(\mathcal{K})$, we set $t(x)^{=1} = \forall y (\bigwedge t(y) \rightarrow (x = y))$. The *extended* $\mathcal{K}^2$-*type* of a pointed model $(\mathfrak{A}, a, b)$, denoted $\mathsf{tp}^*_{\mathcal{K}}(\mathfrak{A}, a, b)$, is the conjunction of

(1) $\mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, a, b) \wedge \mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, a) \wedge \mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, b)[y/x]$;

(2) $\exists y (\mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, a, c) \wedge \mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, c)[y/x])$ for any $c \in \mathsf{dom}(\mathfrak{A}) \setminus \{a, b\}$ such that $\mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, c)$ is realized only once in $\mathfrak{A}$;

(3) $\exists x (\mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, c, b) \wedge \mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, c)[x/y])$ for any $c \in \mathsf{dom}(\mathfrak{A}) \setminus \{a, b\}$ such that $\mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, c)$ is realized only once in $\mathfrak{A}$;

(4) $\neg \exists x \bigwedge t(x)$, for any $t(x) \in T_x(\mathcal{K})$ not realized in $\mathfrak{A}$;

(5) $\exists x (\bigwedge t(x) \wedge t(x)^{=1})$ if $t(x) \in T_x(\mathcal{K})$ is realized exactly once in $\mathfrak{A}$;

(6) $\exists x (\bigwedge t(x) \wedge \neg t^{=1}(x))$ if $t(x) \in T_x(\mathcal{K})$ is realized at least twice in $\mathfrak{A}$;

(7) $\exists x y \bigwedge \mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, c, d) \wedge \mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, c) \wedge \mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, d)[y/x]$ for any $c \neq d$ such that $\mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, c)$ and $\mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, d)$ are realized only once in $\mathfrak{A}$.

We denote by $T^*_{x,y}(\mathcal{K})$ the set of all extended $\mathcal{K}^2$-types. We say that $t(x,y) \in T^*_{x,y}(\mathcal{K})$ is *realized* in $(\mathfrak{A}, a, b)$ if $t(x,y) = \mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, a, b)$. The *extended* $\mathcal{K}^1$-*type* of a pointed model $(\mathfrak{A}, a)$ is defined in the same way with $\mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, a, b)$ and $\mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, b)[y/x]$ removed in Point 1 and with Point 3 completely removed. We also define the realization of such types by pointed models as expected.

3.12. Theorem. *For every labeled* $\mathbf{FO}^2$*-knowledge base such that the tuples in* $E^+ \cup E^-$ *have length* $i \in \{1,2\}$*, the following are equivalent.*

1. $(\mathcal{K}, E^+, E^-)$ *is strongly* $\mathbf{FO}^2$*-separable.*
2. $(\mathcal{K}, E^+, E^-)$ *is strongly* $\mathbf{FO}$*-separable.*
3. *For all* $\mathfrak{A}, \mathfrak{B} \models \mathcal{K}$*,* $\mathbf{a} \in E^+, \mathbf{b} \in E^-$*,* $\mathbf{a}^{\mathfrak{A}}$ *and* $\mathbf{b}^{\mathfrak{B}}$ *do not realize the same extended* $\mathcal{K}^i$*-type.*
4. *The* $\mathbf{FO}^2$*-formula* $(\bigwedge t_1) \vee \cdots \vee (\bigwedge t_n)$ *strongly separates* $(\mathcal{K}, E^+, E^-)$*,* $t_1, \ldots, t_n$ *being the extended* $\mathcal{K}^i$*-types realizable in* $(\mathcal{K}, \mathbf{a})$ *for some* $\mathbf{a} \in E^+$*.*

Proof. Assume that the tuples in E^+ and E^- have length two (the other case is proved similarly). Implications (1) $\Rightarrow$ (2), (3) $\Rightarrow$ (4) and (4) $\Rightarrow$ (1) are straightforward. For (2) $\Rightarrow$ (3) assume that Condition 3 does not hold. Thus, there are $\mathbf{a} = (a_1, a_2) \in E^+$ and $\mathbf{b} = (b_1, b_2) \in E^-$ and models $\mathfrak{A}$ and $\mathfrak{B}$ of $\mathcal{K}$ such that the extended 2-types of $(\mathfrak{A}, \mathbf{a})$ and $(\mathfrak{B}, \mathbf{b})$ coincide. We assume without loss of generality that $\mathfrak{A}, \mathfrak{B}$ are disjoint and that $\mathfrak{B} \models \mathcal{D}'$. We show that there exists a model $\mathfrak{C}$ of $(\mathcal{O}, \mathcal{D}_{\mathbf{a}=\mathbf{b}})$. Then, by Theorem 3.1, $(\mathcal{K}, E^+, E^-)$ is not $\mathbf{FO}$-separable. Before defining $\mathfrak{C}$, assume that $a_1^{\mathfrak{A}} \neq a_2^{\mathfrak{A}}$. The case $a_1^{\mathfrak{A}} = a_2^{\mathfrak{A}}$ is similar and omitted. Then, by the first conjunct of extended types and since $\mathcal{K}^2$-types contain equality assertions, $b_1^{\mathfrak{B}} \neq b_2^{\mathfrak{B}}$. By Points 5 and 6, $\mathfrak{A}$ and $\mathfrak{B}$ realize exactly the same $\mathcal{K}^1$-types once. Let K denote the set of such types.

We define $\mathfrak{C}$ from $\mathfrak{A}$ and $\mathfrak{B}$ by first identifying $a_i^{\mathfrak{A}}$ with $b_i^{\mathfrak{B}}$ for all $i \in \{1,2\}$ and all $c \in \mathsf{dom}(\mathfrak{A})$ with $d \in \mathsf{dom}(\mathfrak{B})$ whenever c and d realize the same $\mathcal{K}^1$-type from K. Then $\mathfrak{C}$ is well defined by the conjuncts in Points 1, 2, 3, and 7 of the definition of extended types. Set $c^{\mathfrak{C}} = c^{\mathfrak{A}}$ for all constants $c \in \mathsf{cons}(\mathcal{D})$ and $(c')^{\mathfrak{C}} = (c')^{\mathfrak{B}}$ for all constants $c' \in \mathsf{cons}(\mathcal{D}')$ (from the definition of $\mathcal{D}_{\mathbf{a}=\mathbf{b}}$). It remains to define the $\mathcal{K}^2$-type realized by (c,d) in $\mathfrak{C}$ for $c \in \mathsf{dom}(\mathfrak{C}) \setminus \mathsf{dom}(\mathfrak{B})$ and $d \in \mathsf{dom}(\mathfrak{C}) \setminus \mathsf{dom}(\mathfrak{A})$. Assume such a (c,d) is given. Then the type $\mathsf{tp}_{\mathcal{K}}(\mathfrak{B}, d)$ is realized in $\mathfrak{A}$, by the formulas in Point 5 and 6 of the definition of extended types. Then let $d' \in \mathsf{dom}(\mathfrak{A})$ such that $\mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, d') = \mathsf{tp}_{\mathcal{K}}(\mathfrak{B}, d)$. We may assume that $d' \neq c$ as $\mathsf{tp}_{\mathcal{K}}(\mathfrak{B}, d)$ is realized at least twice in both $\mathfrak{A}$ and in $\mathfrak{B}$. Now interpret the relations $R \in \mathsf{sig}(\mathcal{K})$ in $\mathfrak{C}$ in such a way that $\mathsf{tp}_{\mathcal{K}}(\mathfrak{C}, d, c) = \mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, d', c)$.

One can show that $\mathfrak{C}$ is a model of $(\mathcal{O}, \mathcal{D}_{\mathbf{a}=\mathbf{b}})$. It is clear by definition that $\mathfrak{C} \models \mathcal{D}_{\mathbf{a}=\mathbf{b}}$. We prove that for all $c \in \mathsf{dom}(\mathfrak{A}) \cup \mathsf{dom}(\mathfrak{B})$ and all $\varphi(x) \in \mathsf{cl}(\mathcal{K})$ we have $\mathfrak{A} \models \varphi(c)$ iff $\mathfrak{C} \models \varphi(c)$ if $c \in \mathsf{dom}(\mathfrak{A})$ and $\mathfrak{B} \models \varphi(c)$ iff $\mathfrak{C} \models \varphi(c)$ if $c \in \mathsf{dom}(\mathfrak{B})$. It is straightforward to check the cases where $\varphi(x) = R(x)$ for $R \in \mathsf{sig}(\mathcal{K})$ and where $\varphi(x)$ is a boolean combination of formulas subject to the induction hypothesis. Now suppose $\varphi(x) = \exists y\, \varphi'(x,y)$.

1. Suppose $c \in \mathsf{dom}(\mathfrak{A}) \setminus \mathsf{dom}(\mathfrak{B})$.

($\Rightarrow$) Suppose $\mathfrak{A} \models \varphi(c)$. Then $\mathfrak{A} \models \varphi'(c,d)$ for some $d \in \mathsf{dom}(\mathfrak{A})$. Being an FO^2 formula with at most two free variables, $\varphi'(x,y)$ is a boolean combination of formulas of the form $R(x,y)$, $R(y,x)$, $\varphi''(x)$, $\varphi''(y)$ where R is either a relation in $\mathsf{sig}(\mathcal{K})$ or the equality symbol and $\varphi'' \in \mathsf{cl}(\mathcal{K})$. We have $\mathfrak{A} \models R(c,d)$ iff $\mathfrak{C} \models R(c,d)$ for any pair $c, d \in \mathsf{dom}(\mathfrak{A})$ by definition of semantics in $\mathfrak{C}$. For the other types, we know $\mathfrak{A} \models \varphi''(e)$ iff $\mathfrak{C} \models \varphi''(e)$ for any $e \in \mathsf{dom}(\mathfrak{A})$, from the induction hypothesis on formulas with at most 1 free variable. Therefore $\mathfrak{C} \models \varphi(c)$.

($\Leftarrow$) Suppose $\mathfrak{C} \models \varphi(c)$. Then $\mathfrak{C} \models \varphi'(c,d)$ for some $d \in \mathsf{dom}(\mathfrak{C})$. Suppose $d \in \mathsf{dom}(\mathfrak{A})$. The same argument as in the above subcase giving $\mathfrak{A} \models \varphi(c) \Rightarrow \mathfrak{C} \models \varphi(c)$ also gives the converse when $c \in \mathsf{dom}(\mathfrak{A}) \setminus \mathsf{dom}(\mathfrak{B})$ and $d \in \mathsf{dom}(\mathfrak{A})$. Suppose $d \in \mathsf{dom}(\mathfrak{B}) \setminus \mathsf{dom}(\mathfrak{A})$. Keep the notation for subformulas of $\varphi'(x,y)$. By definition of $\mathfrak{C}$, there exists $d' \in \mathsf{dom}(\mathfrak{A})$ such that $\mathsf{tp}_{\mathcal{K}}(\mathfrak{B}, d) = \mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, d')$ and $\mathsf{tp}_{\mathcal{K}}(\mathfrak{C}, c, d) = \mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, c, d')$. Then $\mathfrak{C} \models R(c,d)$ iff $\mathfrak{A} \models R(c,d')$. By induction hypothesis, $\mathfrak{C} \models \varphi''(c)$ implies $\mathfrak{A} \models \varphi''(c)$ and $\mathfrak{C} \models \varphi''(d)$ implies $\mathfrak{B} \models \varphi''(d)$, thus $\mathfrak{A} \models \varphi''(d')$ since $\mathsf{tp}_{\mathcal{K}}(\mathfrak{B}, d) = \mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, d')$. Therefore we do have $\mathfrak{A} \models \varphi'(c,d')$, so $\mathfrak{A} \models \varphi(c)$.

2. Suppose $c \in \mathsf{dom}(\mathfrak{B}) \setminus \mathsf{dom}(\mathfrak{A})$.

($\Rightarrow$) Suppose $\mathfrak{B} \models \varphi(c)$. We get $\mathfrak{C} \models \varphi(c)$ via the same proof as for $\mathfrak{A} \models \varphi(c) \Rightarrow \mathfrak{C} \models \varphi(c)$ when $c \in \mathsf{dom}(\mathfrak{A}) \setminus \mathsf{dom}(\mathfrak{B})$.

($\Leftarrow$) Suppose $\mathfrak{C} \models \varphi(c)$. Then $\mathfrak{C} \models \varphi'(c,d)$ for some $d \in \mathsf{dom}(\mathfrak{C})$. Suppose $d \in \mathsf{dom}(\mathfrak{B})$. We get $\mathfrak{C} \models \varphi(c)$ via the same proof as the proof of $\mathfrak{C} \models \varphi'(c,d) \Rightarrow \mathfrak{A} \models \varphi(c)$ when $c \in \mathsf{dom}(\mathfrak{A}) \setminus \mathsf{dom}(\mathfrak{B})$ and $d \in \mathsf{dom}(\mathfrak{A})$. Suppose $d \in \mathsf{dom}(\mathfrak{A}) \setminus \mathsf{dom}(\mathfrak{B})$. Keep the notation for subformulas of $\varphi'(x,y)$. By definition of $\mathfrak{C}$, there exists $c' \in \mathsf{dom}(\mathfrak{A})$ such that $\mathsf{tp}_{\mathcal{K}}(\mathfrak{B}, c) = \mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, c')$ and $\mathsf{tp}_{\mathcal{K}}(\mathfrak{C}, c, d) = \mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, c', d)$. Then $\mathfrak{C} \models R(c,d)$ iff $\mathfrak{A} \models R(c',d)$. By induction hypothesis, $\mathfrak{C} \models \varphi''(d)$ iff $\mathfrak{A} \models \varphi''(d)$ and $\mathfrak{C} \models \varphi''(c)$ iff $\mathfrak{B} \models \varphi''(c)$, thus $\mathfrak{A} \models \varphi''(c')$ since $\mathsf{tp}_{\mathcal{K}}(\mathfrak{B}, c) = \mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, c')$. Therefore we do have $\mathfrak{A} \models \varphi(c')$, thus $\mathfrak{B} \models \varphi(c)$, since $\mathsf{tp}_{\mathcal{K}}(\mathfrak{B}, c) = \mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, c')$.

3. Suppose $c \in \mathsf{dom}(\mathfrak{A}) \cap \mathsf{dom}(\mathfrak{B})$.

There are again two subcases to distinguish. By definition of $\mathfrak{C}$, elements in the intersection are either of types realized once, or equal to $a_i^{\mathfrak{C}}$ for $i \in \{1,2\}$. If $c \in \{a_1^{\mathfrak{C}}, a_2^{\mathfrak{C}}\}$, then $\mathsf{tp}_{\mathcal{K}}(\mathfrak{C}, c)$ is possibly realized more than once.

3.1. Suppose $\mathsf{tp}_{\mathcal{K}}(\mathfrak{C}, c)$ is realized once.

($\Rightarrow$) Suppose $\mathfrak{A} \models \varphi(c)$. Then $\mathfrak{A} \models \varphi'(c,d)$ for some $d \in \mathsf{dom}(\mathfrak{A})$. Same proof as for the implication $\mathfrak{A} \models \varphi(c) \Rightarrow \mathfrak{C} \models \varphi(c)$ when $c \in \mathsf{dom}(\mathfrak{A}) \setminus \mathsf{dom}(\mathfrak{B})$. Suppose

$\mathfrak{B} \models \varphi(c)$. By the same argument, we get $\mathfrak{C} \models \varphi(c)$.

($\Leftarrow$) Suppose $\mathfrak{C} \models \varphi(c)$. We want to prove both $\mathfrak{A} \models \varphi(c)$ and $\mathfrak{B} \models \varphi(c)$. We have $\mathfrak{C} \models \varphi'(c,d)$ for some $d \in \mathsf{dom}(\mathfrak{C})$. Suppose $d \in \mathsf{dom}(\mathfrak{A}) \setminus \mathsf{dom}(\mathfrak{B})$. The proof is then the same as the proof of $\mathfrak{C} \models \varphi(c) \Rightarrow \mathfrak{A} \models \varphi(c)$ when $c \in \mathsf{dom}(\mathfrak{A}) \setminus \mathsf{dom}(\mathfrak{B})$ and $d \in \mathsf{dom}(\mathfrak{A})$. We get $\mathfrak{A} \models \varphi(c)$, thus also $\mathfrak{B} \models \varphi(c)$ since $\mathsf{tp}_{\mathcal{K}}(\mathfrak{A},c) = \mathsf{tp}_{\mathcal{K}}(\mathfrak{B},c)$. Suppose $d \in \mathsf{dom}(\mathfrak{B}) \setminus \mathsf{dom}(\mathfrak{A})$. The same proof as above applies to give $\mathfrak{B} \models \varphi(c)$ and then $\mathfrak{A} \models \varphi(c)$ since $\mathsf{tp}_{\mathcal{K}}(\mathfrak{A},c) = \mathsf{tp}_{\mathcal{K}}(\mathfrak{B},c)$.
Now suppose $d \in \mathsf{dom}(\mathfrak{B}) \cap \mathsf{dom}(\mathfrak{A})$.

a) Suppose $\mathsf{tp}_{\mathcal{K}}(\mathfrak{C},d)$ is realized once. Keep the notation for subformulas of $\varphi'(x,y)$. On unary subformulas $\varphi''(c)$, by induction hypothesis, $\mathfrak{C} \models \varphi''(c)$ iff $\mathfrak{A} \models \varphi''(c)$ and iff $\mathfrak{B} \models \varphi''(c)$ and same for d. As for binary relations and equality, $\mathfrak{C} \models R(c,d)$ implies either $\mathfrak{A} \models R(c,d)$ or $\mathfrak{B} \models R(c,d)$, by definition of semantics in $\mathfrak{C}$. It is sufficient to complete the proof to show that $\mathfrak{A} \models R(c,d)$ and $\mathfrak{B} \models R(c,d)$ are equivalent. Suppose $\mathfrak{A} \models R(c,d)$. Since a and b have the same extended type, it holds that $\mathfrak{B} \models \exists x y \bigwedge \mathsf{tp}_{\mathcal{K}}(\mathfrak{A},c,d) \wedge \mathsf{tp}_{\mathcal{K}}(\mathfrak{A},c) \wedge \mathsf{tp}_{\mathcal{K}}(\mathfrak{A},d)$, point 7 of the definition of extended type. Any element in the intersection $\mathsf{dom}(\mathfrak{A}) \cap \mathsf{dom}(\mathfrak{B})$ has its type realized only once in $\mathfrak{C}$, so c is the only element in $\mathfrak{B}$ satisfying $\mathsf{tp}_{\mathcal{K}}(\mathfrak{A},c)$ and d is the only element in $\mathfrak{B}$ satisfying $\mathsf{tp}_{\mathcal{K}}(\mathfrak{A},d)$, since their types are both realized only once. Thus, $\mathsf{tp}_{\mathcal{K}}(\mathfrak{B},c,d) = \mathsf{tp}_{\mathcal{K}}(\mathfrak{A},c,d)$, therefore $\mathfrak{B} \models R(c,d)$. The converse is proved the same way. We have then proved that $\mathfrak{C} \models \varphi(c)$ implies both $\mathfrak{A} \models \varphi(c)$ and $\mathfrak{B} \models \varphi(c)$.

b) Suppose $d = a_i^{\mathfrak{C}}$ for some $i = 1,2$. Keep the notation for subformulas of $\varphi'(x,y)$. On unary subformulas $\varphi''(c)$, by induction hypothesis, $\mathfrak{C} \models \varphi''(c)$ iff $\mathfrak{A} \models \varphi''(c)$ and iff $\mathfrak{B} \models \varphi''(c)$ and same for d. As for binary relations and equality, $\mathfrak{C} \models R(c,d)$ implies either $\mathfrak{A} \models R(c,d)$ (i.e $\mathfrak{A} \models R(c,a_i^{\mathfrak{A}})$) or $\mathfrak{B} \models R(c,d)$ (i.e $\mathfrak{B} \models R(c,b_i^{\mathfrak{B}})$), by definition of semantics in $\mathfrak{C}$. It is sufficient to complete the proof to show that $\mathfrak{A} \models R(c,d)$ and $\mathfrak{B} \models R(c,d)$ are equivalent. We get $\mathsf{tp}_{\mathcal{K}}(\mathfrak{B},c,d) = \mathsf{tp}_{\mathcal{K}}(\mathfrak{A},c,d)$ the equivalence by condition 2 and 3 of the definition of extended type, thus the equivalence. We have then proved that $\mathfrak{C} \models \varphi(c)$ implies both $\mathfrak{A} \models \varphi(c)$ and $\mathfrak{B} \models \varphi(c)$.

3.2. Suppose $c \in \{a_1^{\mathfrak{C}}, a_2^{\mathfrak{C}}\}$ and $\mathsf{tp}_{\mathcal{K}}(\mathfrak{C},c)$ is realized more than once.

($\Rightarrow$) By the same argument as in the other case, we have that $\mathfrak{A} \models \varphi(c)$ and $\mathfrak{B} \models \varphi(c)$ both imply $\mathfrak{C} \models \varphi(c)$.

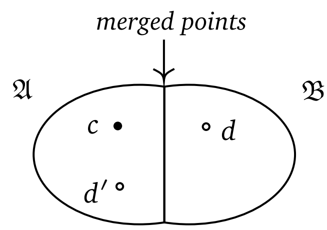
($\Leftarrow$) Suppose $\mathfrak{C} \models \varphi(c)$. We want to prove both $\mathfrak{A} \models \varphi(c)$ and $\mathfrak{B} \models \varphi(c)$. We have $\mathfrak{C} \models \varphi'(c,d)$ for some $d \in \mathsf{dom}(\mathfrak{C})$. Suppose $d \in \mathsf{dom}(\mathfrak{A}) \setminus \mathsf{dom}(\mathfrak{B})$. The proof is then the same as the proof of $\mathfrak{C} \models \varphi(c) \Rightarrow \mathfrak{A} \models \varphi(c)$ when $c \in \mathsf{dom}(\mathfrak{A}) \cap \mathsf{dom}(\mathfrak{B})$ and $d \in \mathsf{dom}(\mathfrak{A}) \setminus \mathsf{dom}(\mathfrak{B})$ and $\mathsf{tp}_{\mathcal{K}}(\mathfrak{C},c)$ is realized once. We get $\mathfrak{A} \models \varphi(c)$,

thus also $\mathfrak{B} \models \varphi(c)$ since $\mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, c) = \mathsf{tp}_{\mathcal{K}}(\mathfrak{B}, c)$. Suppose $d \in \mathsf{dom}(\mathfrak{B}) \setminus \mathsf{dom}(\mathfrak{A})$. The same proof as above applies to give $\mathfrak{B} \models \varphi(c)$ and then $\mathfrak{A} \models \varphi(c)$ since $\mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, c) = \mathsf{tp}_{\mathcal{K}}(\mathfrak{B}, c)$.
Now suppose $d \in \mathsf{dom}(\mathfrak{B}) \cap \mathsf{dom}(\mathfrak{A})$.

a) Suppose $\mathsf{tp}_{\mathcal{K}}(\mathfrak{C}, d)$ is realized once. Keep the notation for subformulas of $\varphi'(x, y)$. On unary subformulas $\varphi''(c)$, by induction hypothesis, $\mathfrak{C} \models \varphi''(c)$ iff $\mathfrak{A} \models \varphi''(c)$ and iff $\mathfrak{B} \models \varphi''(c)$ and same for d. As for binary relations and equality, $\mathfrak{C} \models R(c, d)$ implies either $\mathfrak{A} \models R(c, d)$ (i.e $\mathfrak{A} \models R(a_i^{\mathfrak{A}}, d)$) or $\mathfrak{B} \models R(c, d)$ (i.e $\mathfrak{B} \models R(b_i^{\mathfrak{B}}, d)$), by definition of semantics in $\mathfrak{C}$. It is sufficient to complete the proof to show that $\mathfrak{A} \models R(c, d)$ and $\mathfrak{B} \models R(c, d)$ are equivalent. We get $\mathsf{tp}_{\mathcal{K}}(\mathfrak{B}, c, d) = \mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, c, d)$ the equivalence by condition 2 and 3 of the definition of extended type, thus the equivalence. We have then proved that $\mathfrak{C} \models \varphi(c)$ implies both $\mathfrak{A} \models \varphi(c)$ and $\mathfrak{B} \models \varphi(c)$.

b) Suppose $d = a_i^{\mathfrak{C}}$ for some $i = 1, 2$. Keep the notation for subformulas of $\varphi'(x, y)$. On unary subformulas $\varphi''(c)$, by induction hypothesis, $\mathfrak{C} \models \varphi''(c)$ iff $\mathfrak{A} \models \varphi''(c)$ and iff $\mathfrak{B} \models \varphi''(c)$ and same for d. As for binary relations and equality, $\mathfrak{C} \models R(c, d)$ implies either $\mathfrak{A} \models R(c, d)$ or $\mathfrak{B} \models R(c, d)$, by definition of semantics in $\mathfrak{C}$. It is sufficient to complete the proof to show that $\mathfrak{A} \models R(c, d)$ and $\mathfrak{B} \models R(c, d)$ are equivalent. We get $\mathsf{tp}_{\mathcal{K}}(\mathfrak{B}, c, d) = \mathsf{tp}_{\mathcal{K}}(\mathfrak{A}, c, d)$ by condition 1 of the definition of extended type since both are constants, thus the equivalence. We have then proved that $\mathfrak{C} \models \varphi(c)$ implies both $\mathfrak{A} \models \varphi(c)$ and $\mathfrak{B} \models \varphi(c)$.

$\dashv$



Through similar intermediate equivalences to the ones omitted in the context of **GF**, strongly separating formulas are of size at most $2^{2^{p(||\mathcal{O}||)}}$ (instead of simply $2^{2^{p(||\mathcal{K}||)}}$), p a polynomial. Once more, complexity bounds follow from the link between separability and satisfiability. Recall that satisfiability of **FO**^{2}-knowledge bases is NExp-complete in combined complexity and NP-complete in data complexity.

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

Chapter **4**

Restricted weak separability

Starting from this section, we now add a signature to the input and require separating formulas to be expressed in that signature. This makes it possible to exclude features that one would consider irrelevant. We consider description logics in $\mathbf{DL}_{\mathcal{IO}}$ as well as decidable fragments of **FO** like **GF** and $\mathbf{FO}^2$. Recall from the Preliminaries chapter that in the context of restricted signatures, one can still define a notion of "projectivity" as the allowance of symbols outside of the knowledge base's signature.

Many results from the full signature case do not hold in the restricted case. It was shown in Chapter 2 that in the full signature case, unions of conjunctive queries had the same weakly-separating power as first-order formulas (Thm. 2.1) and, under $\mathcal{ALCI}$ ontologies, as projective $\mathcal{ALCI}$ formulas (Cor. 2.29). With restricted signatures, those equivalences do not hold:

4.1. *Example.* Let $\mathcal{K}_1 = (\mathcal{O}_1, \mathcal{D}_1)$ and $\mathcal{K}_2 = (\mathcal{O}_2, \mathcal{D}_2)$, where $\mathcal{O}_1 = \{A \sqsubseteq \exists R.B \sqcap \exists R.\neg B\}$, $\mathcal{O}_2 = \{A \sqsubseteq \forall R.B\}$, $\mathcal{D}_1 = \{A(a), R(b,c)\}$ and $\mathcal{D}_2 = \{A(a), C(b)\}$. Then $(\mathcal{K}_1, \{a\}, \{b\})$ is not $\mathbf{UCQ}(\{R\})$-separable but is weakly separated by the $\mathbf{FO}(\{R\})$-formula $\exists y \exists y'(R(x,y) \wedge R(x,y') \wedge \neg(y = y'))$. $(\mathcal{K}_2, \{a\}, \{b\})$ is not $\mathbf{UCQ}(\{R,B\})$-separable but is weakly separated by $\forall R.B$, an $\mathcal{ALC}(\{R,B\})$-concept.

In Section 4.1 we give model-theoretic characterisations for projective separability in $\mathbf{DL}_{\mathcal{IO}}$ and discuss some corollaries. In Section 4.2 we observe that it follows from our main characterisation theorem that projective $\mathcal{ALCI}$ and $\mathcal{ALCO}$-separability can be non-projectively captured by a language combining UCQs and DLs. In Section 4.3, we make the crucial observation that for the DLs we consider, restricted separability is tightly connected to the problem of deciding conservative extensions. That reduction implies 2Exp-hardness for projective $(\mathcal{ALC}, \mathcal{ALC})$ and $(\mathcal{ALCI}, \mathcal{ALCI})$-separability. In Section 4.4, we prove a matching 2EXP upper bound via tree automata and the bisimulation characterisation. In Section 4.5, we prove undecidability of projective weak $(\mathcal{ALC}, \mathcal{ALCO})$-separability if constants are allowed as helper symbols. This is done via reduction from an undecidable tiling problem. We then use the same tiling problem to show that $(\mathcal{ALC}, \mathcal{ALCFIO})$-separability is undecidable, both in the projective and non-projective case.

Listed below are combined complexity results in the case where the ontology language and the separation language coincide. The results we obtain in the following section are contrasted with the ones from the weak, full signature case.

Conservative extensions are undecidable in **GF** [JLMSW17], but the strategy we use in $\mathbf{DL}_{\mathcal{IO}}$ to bound the complexity of separability from below with deciding conservative extensions cannot be applied in **GF** to obtain undecidability of separability. It is shown nonetheless in [JLPW21] that (non-)projective weak restricted $(\mathcal{L}_O, \mathcal{L}_S)$-separability is undecidable for any $\mathcal{L}_O$ that contains $\mathbf{GF}^3$ and $\mathcal{L}_S$ that contains $\mathcal{ALC}$, by adapting the proof used in [JLMSW17] to show undecidability of conservative extensions in **GF**.

	prj+full	full	prj+rstr
$\mathcal{ALC}$	NEXP	?	2EXP
$\mathcal{ALCI}$	NEXP	NEXP	2EXP
$\mathcal{ALCO}$	?	?	3EXP [JLPW21]
GF	2EXP	2EXP	Undec [JLPW21]
$\mathbf{FO}^2$	Undec	Undec	Undec

§ 4.1. SEMANTIC CHARACTERISATION IN DL

It is straightforward to translate the bisimulation-based charaterisation from the full signature case (see Theorem 2.16) to the restricted case, both for projective and non-projective separability. However, unlike in the full signature case, the condition we obtain for restricted projective separability (in (2) of Theorem 4.2) is inconvenient for studying its complexity, so we also reformulate it (in (3)) in a way that is well-suited to automata-based decision procedures. That reformulation does not hold for $\mathcal{ALCIO}$ as separation language, as infinite outdegree forest models can be forced. We do not provide such a reformulation for the non-projective case as its complexity is out of the scope of this thesis: it appears from Chapter 2 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 technical challenges, we leave the non-projective case open.

As in the full signature case, we assume without loss of generality that the set of negative examples is a singleton.

4.2. Theorem. *Let $\mathcal{L} \in \mathbf{DL}_{\mathcal{IO}} \setminus \mathcal{ALCIO}$. Let $(\mathcal{K}, E^+, \{b\})$ be a labeled $\mathcal{L}$-knowledge base and $\Sigma \subseteq \mathsf{sig}(\mathcal{K})$. The following conditions are equivalent.*

1. $(\mathcal{K}, E^+, \{b\})$ *is projectively $\mathcal{L}(\Sigma)$-separable using only* rel_1 *as helper symbols.*
2. *There exists an $\mathcal{L}$-forest model $\mathfrak{A} \models \mathcal{K}$ of finite outdegree and $\Sigma' \subseteq \mathsf{rel}_1 \setminus \mathsf{sig}(\mathcal{K})$ such that for all $\mathfrak{B} \models \mathcal{K}, a \in E^+$, $(\mathfrak{B}, a^{\mathfrak{B}}) \not\sim_{\mathcal{L},\Sigma\cup\Sigma'} (\mathfrak{A}, b^{\mathfrak{A}})$.*
3. *There exists an $\mathcal{L}$-forest model $\mathfrak{A} \models \mathcal{K}$ of finite outdegree such that for all $\mathfrak{B} \models \mathcal{K}, a \in E^+$, $(\mathfrak{B}, a^{\mathfrak{B}}) \not\sim^f_{\mathcal{L},\Sigma} (\mathfrak{A}, b^{\mathfrak{A}})$.*

Proof. (1) ⇒ (2) and (2) ⇒ (1) are proved as in the full signature case.
For (2) ⇒ (3), take an $\mathcal{L}$-forest model $\mathfrak{A}$ and Σ' such that (2) holds. We show that (3) holds for $\mathfrak{A}$ as well. Suppose for contradiction that there exists $\mathfrak{B} \models \mathcal{K}$, $a \in E^+$ and a functional Σ-bisimulation f witnessing $(\mathfrak{B}, a^{\mathfrak{B}}) \sim^f_{\mathcal{L},\Sigma} (\mathfrak{A}, b^{\mathfrak{A}})$. As $\Sigma' \cap \mathsf{sig}(\mathcal{K}) = \emptyset$, we may assume without loss of generality that $\mathfrak{B}$ does not interpret any symbol in Σ'. We expand $\mathfrak{B}$ into a model $\mathfrak{B}'$ by setting $A^{\mathfrak{B}'}$ to be all $d \in \mathsf{dom}(f)$ such that $f(d) \in A^{\mathfrak{A}}$, for all $A \in \Sigma'$. The contradiction ensues, as f clearly witnesses $(\mathfrak{B}', a^{\mathfrak{B}'}) \sim_{\mathcal{L},\Sigma\cup\Sigma'} (\mathfrak{A}, b^{\mathfrak{A}})$.

(3) ⇒ (2). Take an $\mathcal{L}$-forest model $\mathfrak{A}$ of $\mathcal{K}$ of finite outdegree such that (3) holds. We may assume without loss of generality that $\mathfrak{A}$ only interprets symbols in $\mathsf{sig}(\mathcal{K})$. Define $\mathfrak{A}'$ by expanding $\mathfrak{A}$ as follows. Take for any $d \in \mathsf{dom}(\mathfrak{A})$ a fresh concept name A_d and set $A_d^{\mathfrak{A}'} = \{d\}$. Then (2) holds for $\mathfrak{A}'$ and $\Sigma' = \{A_d \mid d \in \mathsf{dom}(\mathfrak{A})\}$. ⊣

Theorems 4.2 fails for $\mathcal{ALCIO}$. A counterexample is given by the following.

4.3. *Example.* Let $\mathcal{K} = (\mathcal{O}, \mathcal{D})$, with

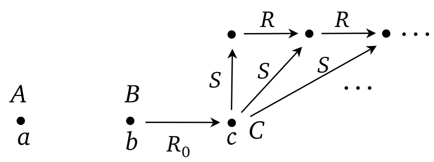
$$\Sigma = \{c, R_0, S, R\}$$

$$\mathcal{D} = \{A(a), B(b), C(c), R_0(b, c)\}$$

$$\mathcal{O} = \left\{ \begin{array}{lcl} C & \sqsubseteq & \exists R_0^-.A \rightarrow A_0 \\ C & \sqsubseteq & \exists S.\top \sqcap \forall S.(E \sqcap \exists R.(E \sqcap \exists S^-.\top)) \\ A_0 & \sqsubseteq & \exists S.\exists S^-.(A_0 \rightarrow \exists S.\exists R.\exists S^-.\neg A_0) \\ C & \sqsubseteq & \neg A \sqcap \neg B \\ E & \equiv & \neg A \sqcap \neg B \sqcap \neg C \end{array} \right\}$$

1. $(\mathcal{K}, \{a\}, \{b\})$ is non-projectively weakly separated by the $\mathcal{ALCIO}(\Sigma)$ concept $D = \neg\exists R_0.\forall S.(\forall S^-.\{c\} \sqcap \forall R.\forall S^-.\{c\})$: suppose for contradiction that there exists $\mathfrak{A} \models \mathcal{K}$ such that $\mathfrak{A} \not\models D(a)$. Then $(a^{\mathfrak{A}}, c^{\mathfrak{A}}) \in R_0^{\mathfrak{A}}$. That, by $\mathcal{O}$, implies $c^{\mathfrak{A}} \in A_0^{\mathfrak{A}}$ and thus $c^{\mathfrak{A}} \in (\exists S.\exists S^-.(A_0 \rightarrow \exists S.\exists R.\exists S^-.\neg A_0))^{\mathfrak{A}}$, contradicting $c^{\mathfrak{A}} \in (\forall S.(\forall S^-.\{c\} \sqcap \forall R.\forall S^-.\{c\}))^{\mathfrak{A}}$. Then, $\mathcal{K} \not\models D(b)$ is straightforwardly witnessed by the model depicted below. An explicit definition is given by $\text{dom}(\mathfrak{A}) = \{a, b, c\} \cup \{a_i : i \geqslant 0\}$ and

$$\begin{array}{ll} A_0^{\mathfrak{A}} = \emptyset & (B')^{\mathfrak{A}} = \emptyset \\ A^{\mathfrak{A}} = \{a\} = a^{\mathfrak{A}} & R_0^{\mathfrak{A}} = \{b, c\} \\ B^{\mathfrak{A}} = \{b\} = b^{\mathfrak{A}} & R^{\mathfrak{A}} = \{(a_i, a_{i+1}) : i \geq 0\} \\ C^{\mathfrak{A}} = \{c\} = c^{\mathfrak{A}} & S^{\mathfrak{A}} = \{(c, a_i) : i \geq 0\} \end{array}$$



2. 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 $(\mathfrak{B}, a^{\mathfrak{B}}) \sim^f_{\mathcal{ALCIO},\Sigma} (\mathfrak{A}, b^{\mathfrak{A}})$. Let $\mathfrak{A} \models \mathcal{K}$. We construct $\mathfrak{B}$ as follows. Let $\text{dom}(\mathfrak{B}) = \text{dom}(\mathfrak{A})$, $a^{\mathfrak{B}} = b^{\mathfrak{B}} = b^{\mathfrak{A}}$, $c^{\mathfrak{B}} = c^{\mathfrak{A}}$, $A^{\mathfrak{B}} = \{a^{\mathfrak{B}}\}$, $A_0^{\mathfrak{B}} = C^{\mathfrak{B}} = \{c^{\mathfrak{B}}\}$, and $B^{\mathfrak{B}} = B^{\mathfrak{A}}$. Let $\rho^{\mathfrak{B}} = \rho^{\mathfrak{A}}$ for all role names ρ. Clearly, $(\mathfrak{A}, b^{\mathfrak{A}}) \simeq_\Sigma (\mathfrak{B}, a^{\mathfrak{B}})$, as $\mathfrak{B}$ only differs from $\mathfrak{A}$ with respect to symbols outside of Σ. It is clear that $\mathfrak{B} \models \mathcal{D}$. We then check that $\mathfrak{B} \models \mathcal{O}$. 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 $\mathfrak{A}$. Assume the third inclusion is not satisfied. Then, by $A_0^{\mathfrak{B}} = \{c^{\mathfrak{B}}\}$, 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 a cycle between nodes distinct from interpretations of constants) or it does not have finite $\mathcal{ALCIO}$-outdegree, witnessed by $c^{\mathfrak{A}}$.

With $\mathcal{ALCIO}$, condition (2) from Theorem 4.2 is too strong to be equivalent to (1), but removing the finite outdegree requirement from (2) would make it too weak, as shown by the following example.

4.4. *Example.* Let $\Sigma = \{c, R_0, S, R\}$ and $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ with

$$\mathcal{D} = \{A(a), B(b), C(c), R_0(b, c)\}$$

$$\mathcal{O} = \left\{ \begin{array}{lcl} C & \sqsubseteq & \exists R_0^-.A \rightarrow A_0 \\ A_0 & \sqsubseteq & (\exists S.\top \sqcap \forall S.\exists R.\exists S^-.A_0) \rightarrow \exists S.B' \\ B' & \sqsubseteq & \exists R^-.B' \\ C & \sqsubseteq & \neg A \sqcap \neg B \end{array} \right\}$$

1. $(\mathcal{K}, \{a\}, \{b\})$ is not weakly projectively $\mathcal{ALCIO}(\Sigma)$-separable using concept names as helper symbols.

Let $\mathfrak{A} \models \mathcal{K}$. We show there exists $\mathfrak{B} \models \mathcal{K}$ such that $(\mathfrak{B}, a^{\mathfrak{B}}) \equiv_{\mathcal{ALCIO},\Sigma\cup\Sigma'} (\mathfrak{A}, b^{\mathfrak{A}})$, where $\Sigma' = \mathsf{rel}_1 \setminus \mathsf{sig}(\mathcal{K})$. Let $\mathfrak{B}_0$ be defined 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 $B'^{\mathfrak{B}_0}$ according to a case distinction. Suppose $c^{\mathfrak{B}_0} \notin (\exists S.\top \sqcap \forall S.\exists R.\exists S^-.A_0)^{\mathfrak{B}_0}$. Then set $B'^{\mathfrak{B}_0} = \emptyset$. Then $\mathfrak{B}_0$ is a model of $\mathcal{K}$ and the identity is a Σ-isomorphism between $\mathfrak{B}_0$ and Suppose $c^{\mathfrak{B}_0} \in (\exists S.\top \sqcap \forall S.\exists R.\exists S^-.A_0)^{\mathfrak{B}_0}$. 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, $\mathfrak{B} \models \mathcal{K}$ and $(\mathfrak{A}, b^{\mathfrak{A}}) \equiv_{\mathcal{ALCIO},\Sigma\cup\Sigma'} (\mathfrak{B}, a^{\mathfrak{B}})$, which concludes the proof.

2. There exists $\mathfrak{A} \models \mathcal{K}$ such that $(\mathfrak{B}, a^{\mathfrak{B}}) \not\sim_{\mathcal{ALCIO},\Sigma} (\mathfrak{A}, b^{\mathfrak{A}})$ for all $\mathfrak{B} \models \mathcal{K}$.

Consider the model $\mathfrak{A}$ depicted in Example 4.3. It is immediate that $\mathfrak{A} \models \mathcal{K}$. If $(\mathfrak{B}, a^{\mathfrak{B}}) \sim_{\mathcal{ALCIO},\Sigma} (\mathfrak{A}, b^{\mathfrak{A}})$, then, as $c, S, R, R_0 \in \Sigma$, $b^{\mathfrak{A}} \in (\exists R_0.(\{c\} \sqcap \exists S.\top \sqcap \forall S.\exists R.\exists S^-.\{c\}))^{\mathfrak{A}}$ implies $a^{\mathfrak{B}} \in (\exists R_0.(\{c\} \sqcap \exists S.\top \sqcap \forall S.\exists R.\exists S^-.\{c\}))^{\mathfrak{B}}$ and $a^{\mathfrak{B}} \in (\exists R_0.(A_0 \sqcap \exists S.\top \sqcap \forall S.\exists R.\exists S^-.A_0))^{\mathfrak{B}}$ as $c^{\mathfrak{B}} \in A_0^{\mathfrak{B}}$, in virtue of the fact that $\{A(a), C(c), R_0(a,c)\} \subseteq \mathcal{D}$ and $C \sqsubseteq \exists R_0^-.A \rightarrow A_0 \in \mathcal{O}$. Then, by $\mathcal{O}$'s second inclusion we get $a^{\mathfrak{B}} \in (\exists R_0.\exists S.B')^{\mathfrak{B}}$ 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 Σ-bisimulation including $(a^{\mathfrak{B}}, b^{\mathfrak{A}})$ is then impossible, as $\{R_0, S, R\} \subseteq \Sigma$.

In $\mathbf{DL}_{\mathcal{IQ}}$, Theorem 4.2 also holds if role names are allowed as helper symbols.

4.5. Theorem. *Let $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$. Restricted weak projective $(\mathcal{L}, \mathcal{L})$-separability is invariant under the addition of role names as helper symbols.*

Proof. We employ the characterization of projective separability given in Theorem 4.2. Observe that the following conditions are equivalent.

1. There exists an $\mathcal{L}(\Sigma \cup ((\mathsf{rel}_1 \cup \mathsf{rel}_2) \setminus \mathsf{sig}(\mathcal{K})))$-concept C such that $\mathcal{K} \models C(a)$ for all $a \in E^+$ and $\mathcal{K} \not\models C(b)$.
2. There exists an $\mathcal{L}$-forest model $\mathfrak{A}$ of $\mathcal{K}$ of finite outdegree and a set $\Sigma' \subseteq (\mathsf{rel}_1 \cup \mathsf{rel}_2) \setminus \mathsf{sig}(\mathcal{K})$ such that $(\mathfrak{B}, a^{\mathfrak{B}}) \not\sim_{\mathcal{L},\Sigma\cup\Sigma'} (\mathfrak{A}, b^{\mathfrak{A}})$ for any $\mathfrak{B} \models \mathcal{K}$ and

$a \in E^+$.

It suffices to show that (2) is equivalent to (3) of Theorem 4.2. The right to left direction is proved exactly as in the context of Theorem 4.2. For a proof of the converse by contradiction, assume that there exists an $\mathcal{L}$-forest model $\mathfrak{A}$ of $\mathcal{K}$ satisfying (2) above for Σ' but not (3) of Theorem 4.2. Let $\mathfrak{B} \models \mathcal{K}$ and f witness $(\mathfrak{B}, a^{\mathfrak{B}}) \sim^f_{\mathcal{L},\Sigma} (\mathfrak{A}, b^{\mathfrak{A}})$ for some $a \in E^+$. We modify $\mathfrak{B}$ to obtain a model $\mathfrak{B}'$ of $\mathcal{K}$ such that $(\mathfrak{B}', a^{\mathfrak{B}'}) \sim_{\mathcal{L},\Sigma\cup\Sigma'} (\mathfrak{A}, b^{\mathfrak{A}})$ and thus obtain a contradiction. $\mathfrak{B}'$ is obtained from $\mathfrak{B}$ by first assuming without loss of generality that $\mathfrak{B}$ does not interpret any symbol in Σ' and then

- taking the disjoint union $\mathfrak{B} \uplus \mathfrak{A}'$ of $\mathfrak{B}$ and a copy $\mathfrak{A}'$ of $\mathfrak{A}$ that does not interpret any symbol in Σ';
- observing that the function $g = f \cup \mathsf{id}$, where id maps every node in $\mathfrak{A}'$ to its copy in $\mathfrak{A}$, 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 $A \in \mathsf{rel}_1 \cap \Sigma'$ and $R^{\mathfrak{B}'} = g^{-1}(R^{\mathfrak{A}})$ for all $R \in \mathsf{rel}_2 \cap \Sigma'$.

Then g is a (functional) $\mathcal{L}(\Sigma \cup \Sigma')$-bisimulation between $\mathfrak{B}'$ and $\mathfrak{A}$. ⊣

The above result fails when nominals are involved in the separation language, *i.e.* adding role names to the definition of projectivity can turn inseparable labeled KBs into separable ones.

4.6. *Example.* Let $\mathcal{K} = (\mathcal{O}, \mathcal{D})$, where $\mathcal{O} = \{A_0 \sqcap \exists R.\top \sqsubseteq \bot, B \sqsubseteq \forall R.A\}$, $\mathcal{D} = \{R(c,a), A_0(a), A_0(b)\}$ and $\Sigma = \{c, B, A\}$. Then, assuming $R_I \notin \mathsf{sig}(\mathcal{K})$, $(\mathcal{K}, \{a\}, \{b\})$ is not projectively $\mathcal{ALCIO}(\Sigma)$-separable with only rel_1 for helper symbols, but the $\mathcal{ALCO}(\Sigma \cup \{R_I\})$-concept $\exists R_I.(\{c\} \sqcap B) \rightarrow A$ separates $(\mathcal{K}, \{a\}, \{b\})$.

However, in that case, a single additional role name suffices to capture separability.

4.7. Theorem. *Let $\mathcal{L} \in \{\mathcal{ALCO}, \mathcal{ALCIO}\}$, $(\mathcal{K}, E^+, \{b\})$ be a labeled $\mathcal{L}$-knowledge base and $\Sigma \subseteq \mathsf{sig}(\mathcal{K})$. Let R_I be a fresh role name and let $\mathcal{K}'$ be the extension of $\mathcal{K}$ with $\exists R_I.\top \sqsubseteq \exists R_I.\top$. Then the following conditions are equivalent.*

1. *$(\mathcal{K}, E^+, \{b\})$ is projectively $\mathcal{L}(\Sigma)$-separable with $\mathsf{rel}_1 \cup \mathsf{rel}_2$ as helper symbols.*
2. *$(\mathcal{K}', E^+, \{b\})$ is projectively $\mathcal{L}(\Sigma \cup \{R_I\})$-separable with only rel_1 as helper symbols.*

Proof. Assume that $\mathcal{L} = \mathcal{ALCO}$. As (2) trivially implies (1), we show the converse. Assume (1). Let $\mathfrak{A} \models \mathcal{K}$ be an $\mathcal{L}$-forest model of finite outdegree and let $\Sigma' \subseteq (\mathsf{rel}_1 \cup \mathsf{rel}_2) \setminus \mathsf{sig}(\mathcal{K})$ such that $(\mathfrak{B}, a^{\mathfrak{B}}) \not\sim_{\mathcal{L},\Sigma\cup\Sigma'} (\mathfrak{A}, b^{\mathfrak{A}})$ for all $\mathfrak{B} \models \mathcal{K}$

and $a \in E^+$. Assume for contradiction that there does not exist any such model if $\Sigma \cup \Sigma'$ is replaced by $\Sigma \cup \{R_I\}$. Obtain $\mathfrak{A}'$ from $\mathfrak{A}$ by setting $R^{\mathfrak{A}'} = \emptyset$ for all $R \in \mathsf{rel}_2 \cap \Sigma'$ and $R_I^{\mathfrak{A}'} = \{(b^{\mathfrak{A}}, c^{\mathfrak{A}}) \mid c \in \mathsf{cons}(\mathcal{D})\} \cup \bigcup_{R \in \mathsf{rel}_2} R^{\mathfrak{A}}$. Then $\mathfrak{A}'$ is an $\mathcal{L}$-forest model of $\mathcal{K}$ of finite outdegree. Thus, by assumption there exists $\mathfrak{B} \models \mathcal{K}$ and $a \in E^+$ 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, as $\mathfrak{A}$ is a forest model with only one connected component (all constants are connected via R_I). Now obtain $\mathfrak{B}'$ from $\mathfrak{B}$ by setting $A^{\mathfrak{B}'} = f^{-1}(A^{\mathfrak{A}})$ for all $A \in \mathsf{rel}_1 \cap \Sigma'$ and $R^{\mathfrak{B}'} = f^{-1}(R^{\mathfrak{A}})$ for all $R \in \mathsf{rel}_2 \cap \Sigma'$. Then f is a functional $\mathcal{L}(\Sigma \cup \Sigma')$-bisimulation between $(\mathfrak{B}', a^{\mathfrak{B}'})$ and $(\mathfrak{A}, b^{\mathfrak{A}})$, hence the contradiction.

Assume that $\mathcal{L} = \mathcal{ALCIO}$. Then we cannot use Theorem 4.2 as it does not hold for $\mathcal{ALCIO}$. However, if one replaces $\mathcal{L}$-forest models of finite outdegree by ω-saturated models, then Theorem 4.2 holds for $\mathcal{ALCIO}$. Now exactly the same proof can be done for $\mathcal{ALCIO}$ as for $\mathcal{ALCO}$ using ω-saturated models instead of forest models. ⊣

§ 4.2. HYBRID UCQS

We introduce, for $\mathcal{L}$ a description logic, the languages $\mathbf{CQ}^{\mathcal{L}}$ and $\mathbf{UCQ}^{\mathcal{L}}$, as well as their rooted counterparts $\mathbf{CQ}_r^{\mathcal{L}}$ and $\mathbf{UCQ}_r^{\mathcal{L}}$. They consist of (resp. rooted) UCQs where atomic predicates are replaced by $\mathcal{L}$-concepts. The added expressive power of UCQs to $\mathcal{L}$ allows to non-projectively separate instances that would otherwise only be projectively separable. The canonical example of a loop on the positive example, which requires projectivity for any of the considered DLs, is easily dealt with by UCQs. We show in Theorems 4.16 and 4.18 that for the selected combinations of ontology and separation languages, those projectively separable instances are in fact exactly all that can non-projectively be separated by $\mathbf{UCQ}_r^{\mathcal{L}}$.

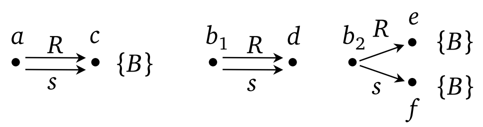
4.8. *Definition.* Let $\mathcal{L}$ be a description logic. Let $\mathbf{CQ}^{\mathcal{L}}$ denote the language of all **FO**-formulas $\varphi(x) = \exists \mathbf{y}\, \psi(x, \mathbf{y})$ where ψ is a conjunction of atoms $C(t)$, C an $\mathcal{L}$-concept, or $R(t_1, t_2)$ with t, t_1, t_2 variables, and x is the single free variable of $\varphi(x)$. Let $\mathbf{UCQ}^{\mathcal{L}}$ denote the language of all finite disjunctions of $\mathbf{CQ}^{\mathcal{L}}$ formulas sharing the same unique free variable. Let $\mathbf{CQ}_r^{\mathcal{L}}$ denote the formulas $\varphi(x)$ in $\mathbf{CQ}^{\mathcal{L}}$ that are $\mathcal{L}$-rooted in x and similarly for $\mathbf{UCQ}_r^{\mathcal{L}}$.

4.9. Note that $\mathbf{UCQ}^{\mathcal{L}}$, like **UCQ**, is not syntactically closed under conjunction. We still safely treat it as an **FO**-fragment, as every conjunction is equivalent to a $\mathbf{UCQ}^{\mathcal{L}}$-formula.

The following example shows how languages of the form $\mathbf{UCQ}^{\mathcal{L}}_r$ can also help give more elegant non-projective solutions to instances that cannot be separated non-projectively in a 'natural way' by description logics in $\mathbf{DL}_{\mathcal{IOQ}}$.

4.10. *Example.* Let $\Sigma = \{R, S, T, A\}$ and $\mathcal{K} = (\mathcal{O}, \mathcal{D})$, where $\mathcal{O} = \{B \sqsubseteq \forall T.A\}$ and

$$\mathcal{D} = \big\{R(a,c), S(a,c), R(b_1,d), S(b_1,d),\\ R(b_2,e), S(b_2,e), B(c), B(e), B(f)\big\} \text{ (cf. figure)}$$



Then $(\mathcal{K}, \{a\}, \{b_1, b_2\})$ is Σ-separated by $\exists y\, R(x,y) \wedge S(x,y) \wedge (\forall T.A)(y) \in \mathbf{CQ}^{\mathcal{ALC}}_r$. The 'simplest' $\mathcal{ALC}$-concept Σ-separating $(\mathcal{K}, \{a\}, \{b_1, b_2\})$ is $(\exists R.\forall T.A) \sqcap (\forall R.X \rightarrow \exists S.X)$, where X is fresh.

For the main result we need the following basic properties, combining homomorphisms and bisimulations to semantically characterize logical implication in $\mathbf{CQ}^{\mathcal{L}}$ and $\mathbf{CQ}^{\mathcal{L}}_r$.

4.11. *Definition.* Let $(\mathfrak{A}, a), (\mathfrak{B}, b)$ be two pointed models, and D such that $a \in D \subseteq \mathsf{dom}(\mathfrak{A})$. Let $\mathcal{L} \in$ and Σ a signature. Then a $\mathbf{CQ}^{\mathcal{L}}(\Sigma)$-*homomorphism* with domain D between $(\mathfrak{A}, a)$ and $(\mathfrak{B}, b)$ is a Σ-homomorphism $h : \mathfrak{A}_{|D} \rightarrow \mathfrak{B}$ such that $h(a) = b$ and $(\mathfrak{A}, a') \sim_{\mathcal{L},\Sigma} (\mathfrak{B}, h(a'))$ for all $a' \in D$. We write $(\mathfrak{A}, a) \rightarrow_{D,\mathcal{L},\Sigma} (\mathfrak{B}, b)$.

We write

1. $(\mathfrak{A}, a) \Rightarrow_{\mathbf{CQ}^{\mathcal{L}},\Sigma} (\mathfrak{B}, b)$ if $\mathfrak{A} \models \varphi(a)$ implies $\mathfrak{B} \models \varphi(b)$ for all $\varphi(x)$ in $\mathbf{CQ}^{\mathcal{L}}(\Sigma)$,
2. $(\mathfrak{A}, a) \Rightarrow_{\mathbf{CQ}^{\mathcal{L}}_r,\Sigma} (\mathfrak{B}, b)$ if $\mathfrak{A} \models \varphi(a)$ implies $\mathfrak{B} \models \varphi(b)$ for all $\varphi(x)$ in $\mathbf{CQ}^{\mathcal{L}}_r(\Sigma)$,
3. $(\mathfrak{A}, a) \Rightarrow^{\text{mod}}_{\mathbf{CQ}^{\mathcal{L}},\Sigma} (\mathfrak{B}, b)$ if $(\mathfrak{A}, a) \rightarrow_{D,\mathcal{L},\Sigma} (\mathfrak{B}, b)$ for all finite $D \subseteq \mathsf{dom}(\mathfrak{A})$,
4. $(\mathfrak{A}, a) \Rightarrow^{\text{mod}}_{\mathbf{CQ}^{\mathcal{L}}_r,\Sigma} (\mathfrak{B}, b)$ if $(\mathfrak{A}, a) \rightarrow_{D,\mathcal{L},\Sigma} (\mathfrak{B}, b)$ for all finite $D \subseteq \mathsf{dom}(\mathfrak{A})$ such that the Σ-reduct of $\mathfrak{A}_{|D}$ is $\mathcal{L}$-rooted in a.

4.12. Lemma. *Let $\mathcal{L} \in \mathbf{DL}_{\mathcal{IO}}$, $(\mathfrak{A}, a)$ and $(\mathfrak{B}, b)$ be pointed models and Σ a signature.*

1. $(\mathfrak{A}, a) \Rightarrow^{\text{mod}}_{\mathbf{CQ}^{\mathcal{L}},\Sigma} (\mathfrak{B}, b)$ *implies* $(\mathfrak{A}, a) \Rightarrow_{\mathbf{CQ}^{\mathcal{L}},\Sigma} (\mathfrak{B}, b)$.
2. $(\mathfrak{A}, a) \Rightarrow^{\text{mod}}_{\mathbf{CQ}^{\mathcal{L}}_r,\Sigma} (\mathfrak{B}, b)$ *implies* $(\mathfrak{A}, a) \Rightarrow_{\mathbf{CQ}^{\mathcal{L}}_r,\Sigma} (\mathfrak{B}, b)$.

Proof. The proof for (2) contains the proof for (1). Assume the premice of (2) and let $\varphi(x)$ be a formula in $\mathbf{CQ}^{\mathcal{L}}_r(\Sigma)$ such that $\mathfrak{A} \models \varphi(a)$. Then there exists a mapping h from the set $\mathsf{var}(\varphi)$ of variables in $\varphi(x)$ to $\mathfrak{A}$ such that $h(x)=a$, if $R(y,z)$ is a conjunct of $\varphi(x)$, then $(h(y),h(z))\in R^{\mathfrak{A}}$, and if $C(y)$ is a conjunct of $\varphi(x)$, then $h(y)\in C^{\mathfrak{A}}$. Let $D=h[\mathsf{var}(\varphi)]$. Then the Σ-reduct of $\mathfrak{A}_{|D}$ is $\mathcal{L}$-rooted in a and, by definition of $\Rightarrow^{\text{mod}}_{\mathbf{CQ}^{\mathcal{L}},\Sigma}$, we have a Σ-homomorphism $h':\mathfrak{A}_{|D}\to\mathfrak{B}$ such that $h'(a)=b$ and $(\mathfrak{A},c)\sim_{\mathcal{L},\Sigma}(\mathfrak{B},h'(c))$ for all $c\in D$. Then, by Lemma 1.25, $h'\circ h(y)\in C^{\mathfrak{B}}$ if $C(y)$ is a conjunct of φ. Thus, $\mathfrak{B}\models\varphi(b)$, as required. The proof for $\mathbf{CQ}^{\mathcal{L}}$ is the same except that the one does not need to observe that the Σ-reduct of $\mathfrak{A}_{|D}$ is $\mathcal{L}$-rooted in a. ⊣

4.13. Lemma. *Let $\mathcal{L}\in\mathbf{DL}_{\mathcal{IO}}$, $(\mathfrak{A},a)$ and $(\mathfrak{B},b)$ be pointed models and Σ a signature.*

1. $(\mathfrak{A},a)\Rightarrow^{\text{mod}}_{\mathbf{CQ}^{\mathcal{L}},\Sigma}(\mathfrak{B},b)$ *iff* $(\mathfrak{A},a)\Rightarrow_{\mathbf{CQ}^{\mathcal{L}},\Sigma}(\mathfrak{B},b)$ *if $\mathfrak{B}$ is ω-saturated.*
2. $(\mathfrak{A},a)\Rightarrow^{\text{mod}}_{\mathbf{CQ}^{\mathcal{L}}_r,\Sigma}(\mathfrak{B},b)$ *iff* $(\mathfrak{A},a)\Rightarrow_{\mathbf{CQ}^{\mathcal{L}}_r,\Sigma}(\mathfrak{B},b)$ *if $\mathfrak{B}$ is ω-saturated or has finite outdegree.*

Proof. The proof for (2) contains the proof for (1). Given the previous Lemma, it is sufficient to prove only one implication. Assume that $(\mathfrak{A},a)\Rightarrow_{\mathbf{CQ}^{\mathcal{L}}_r,\Sigma}(\mathfrak{B},b)$. To show that $(\mathfrak{A},a)\Rightarrow^{\text{mod}}_{\mathbf{CQ}^{\mathcal{L}}_r,\Sigma}(\mathfrak{B},b)$, let D be such that the Σ-reduct of $\mathfrak{A}_{|D}$ is $\mathcal{L}$-rooted at a. Consider the set of 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 $(\mathfrak{A},a)\Rightarrow_{\mathbf{CQ}^{\mathcal{L}}_r,\Sigma}(\mathfrak{B},b)$ 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 ω-saturated, then $q^{\mathfrak{A}}_D$ is satisfied in $\mathfrak{B}$. If $\mathfrak{B}$ has finite outdegree then this can be shown directly using the condition that the Σ-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 Σ-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 $\mathbf{CQ}^{\mathcal{L}}$ also follows, as saturation of $\mathfrak{B}$ implies satisfaction of $q^{\mathfrak{A}}_D$ regardless of whether the Σ-reduct of $\mathfrak{A}_{|D}$ is rooted in a. ⊣

From this point on, we ignore any separation language containing $\mathcal{ALCIO}$ and any ontology language containing $\mathcal{ALCO}$: our aim is to show that hybrid UCQs can non-projectively capture projective separability in DL. The characterisation from 4.2 is crucial to establish that connection. As shown in Section 4.1 we are still unaware of any such characterisation for $\mathcal{ALCIO}$, and we show that the main theorem linking hybrid UCQs to DLs fails when the ontology contains constants.

Prior to stating the main theorem, it now suffices to characterize separability in rooted hybrid UCQs using the model-theoretic tools introduced above.

4.14. Proposition. *Let $(\mathcal{L}_O, \mathcal{L}_S)$ be either $(\mathcal{ALCI}, \mathcal{ALCI})$ or $(\mathcal{ALC}, \mathcal{ALCO})$. Let $(\mathcal{K}, E^+, \{b\})$ be a labeled $\mathcal{L}_S$-knowledge base and $\Sigma \subseteq \mathsf{sig}(\mathcal{K})$. The following conditions are equivalent.*

1. *$(\mathcal{K}, E^+, \{b\})$ is non-projectively $\mathbf{UCQ}_r^{\mathcal{L}_S}(\Sigma)$-separable;*
2. *There exists an $\mathcal{L}_S$-forest $\mathfrak{A} \models \mathcal{K}$ of finite $\mathcal{L}_S$-outdegree such that $(\mathfrak{B}, a^{\mathfrak{B}}) \not\Rightarrow^{\mathrm{mod}}_{\mathbf{CQ}_r^{\mathcal{L}_S},\Sigma}$ $(\mathfrak{A}, b^{\mathfrak{A}})$ for all $\mathfrak{B} \models \mathcal{K}$ and $a \in E^+$.*

Proof.

(1) ⇒ (2). Let $\varphi(x)$ in $\mathbf{UCQ}_r^{\mathcal{L}_S}(\Sigma)$ such that $\mathcal{K} \models \varphi(a)$ for all $a \in E^+$ and $\mathcal{K} \not\models \varphi(b)$. Then there exists $\mathfrak{A} \models \mathcal{K}$ such that $\mathfrak{A} \not\models \varphi(b)$. Using the same arguments as in Lemma 1.25, we may assume w.l.o.g. that $\mathfrak{A}$ is an $\mathcal{L}_S$-forest model of finite $\mathcal{L}_S$-outdegree. Then, $(\mathfrak{B}, a^{\mathfrak{B}}) \not\Rightarrow_{\mathbf{CQ}_r^{\mathcal{L}_S},\Sigma} (\mathfrak{A}, b^{\mathfrak{A}})$ implies (2) by Lemma 4.12.

(2) ⇒ (1). Suppose ¬(1). Let $\mathfrak{A}$ be any $\mathcal{L}_S$-forest model of $\mathcal{K}$ of finite $\mathcal{L}_S$-outdegree and set $\Gamma = \mathcal{K} \cup \{\neg\varphi(x) \mid \varphi(x) \in \mathbf{UCQ}_r^{\mathcal{L}_S}(\Sigma), \mathfrak{A} \models \neg\varphi(b^{\mathfrak{A}})\}$. We show that, by compactness, Γ is satisfiable with $x = a$ for some $a \in E^+$. Assume for contradiction that it is not satisfiable. Then for any $a \in E^+$ there exists a finite subset Γ'_a of Γ such that Γ'_a is not satisfiable with $x = a$. Then $\Gamma' = \bigcup_{a \in E^+} \Gamma'_a$ is not satisfiable with $x = a$, for any $a \in E^+$. We may assume that $\Gamma' = \mathcal{K} \cup \{\neg\varphi_1(x), \ldots, \neg\varphi_n(x)\}$. Then $\mathcal{K} \models \bigvee_i \varphi_i(a)$ for all $a \in E^+$. Observe that $\bigvee_i \varphi_i \in \mathbf{UCQ}_r^{\mathcal{L}_S}$. Thus, as we assume ¬(1), $\mathcal{K} \models \bigvee_i \varphi_i(b)$. Hence $\mathfrak{A} \models \bigvee_i \varphi_i(b)$ and so there exists i such that $\mathfrak{A} \models \varphi_i(b)$. We have derived a contradiction. Let $\mathfrak{B}$ be a model of $\mathcal{K}$ satisfying Γ in some $a \in E^+$. We can assume without loss of generality that $\mathfrak{B}$ is ω-saturated (cf. Thm 1.7), as if it is not we can simply take one of its elementary extensions. By definition, $(\mathfrak{B}, a^{\mathfrak{B}}) \Rightarrow_{\mathbf{CQ}_r^{\mathcal{L}_S},\Sigma} (\mathfrak{A}, b^{\mathfrak{A}})$. By Lemma 4.12, $(\mathfrak{B}, a^{\mathfrak{B}}) \Rightarrow^{\mathrm{mod}}_{\mathbf{CQ}_r^{\mathcal{L}_S},\Sigma} (\mathfrak{A}, b^{\mathfrak{A}})$, *i.e.* ¬(2). ⊣

We now come to the central result of this section: that $\mathbf{UCQ}_r^{\mathcal{L}_S}$ can non-projectively capture projective separability with $\mathcal{L}_S$, for some DL instances of $\mathcal{L}_O, \mathcal{L}_S$. We start with Theorem 4.16, where $\mathcal{L}_O = \mathcal{L}_S = \mathcal{ALCI}$. To prove it, we require the notion of '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.

4.15. *Definition.* We define the *k-unfolding* $\mathfrak{B}_d^{\leqslant k}$ of a model $\mathfrak{B}$ at $d \in \mathsf{dom}(\mathfrak{B})$ as follows, for any $k > 0$.

1. The domain of $\mathfrak{B}_d^{\leqslant k}$ 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 $i_j \leqslant k$ for all $j \leqslant n$, where all $R_i \in \mathsf{rel}_2^-$. Let $\mathsf{tail}(w) = d_n$.
2. $A^{\mathfrak{B}_d^{\leqslant k}} = \left\{w \in W : \mathsf{tail}(w) \in A^{\mathfrak{B}}\right\}$ for all $A \in \mathsf{rel}_1$.

3. $R^{\mathfrak{B}_d^{\leqslant k}} = \{(w_1, w_2) \in W^2 : \exists d \in \mathsf{dom}(\mathfrak{B}), i \leqslant k \text{ with } w_2 = w_1 R(d,i)\}$ for all $R \in \mathsf{rel}_2^-$.

4.16. Theorem. *Let $(\mathcal{K}, E^+, \{b\})$ be a labeled $\mathcal{ALCI}$-knowledge base and $\Sigma \subseteq \mathsf{sig}(\mathcal{K})$. Then the following conditions are equivalent.*

1. *$(\mathcal{K}, E^+, \{b\})$ is projectively $\mathcal{ALCI}(\Sigma)$-separable;*
2. *$(\mathcal{K}, E^+, \{b\})$ is non-projectively $\mathbf{UCQ}_r^{\mathcal{ALCI}}(\Sigma)$-separable.*

Proof. It suffices to show that (3) of Theorem 4.2 and (2) of Lemma 4.14 are equivalent.

($\Rightarrow$) Assume first that (3) of Theorem 4.2 holds for $\mathfrak{A}$. For a model $\mathfrak{C}$ of $\mathcal{K}$ we denote by $\mathcal{D}_\Sigma^{\mathfrak{C},a}$ the maximal Σ-connected component of $a^{\mathfrak{C}}$ in $\mathfrak{C}_{|\mathcal{D}^{\mathfrak{C}}}$, where $\mathcal{D}^{\mathfrak{C}} = \{c^{\mathfrak{C}} \mid c \in \mathsf{cons}(\mathcal{D})\}$. To show that (2) of Lemma 4.14 holds, we show that $(\mathfrak{B}, a^{\mathfrak{B}}) \not\rightarrow_{\mathsf{dom}(\mathcal{D}_\Sigma^{\mathfrak{B},a}),\mathcal{ALCI},\Sigma} (\mathfrak{A}, b^{\mathfrak{A}})$ for all $\mathfrak{B} \models \mathcal{K}$ and all $a \in E^+$. For contradiction assume there exist $\mathfrak{B} \models \mathcal{K}$, $a \in E^+$ and h witnessing the negation. We aim to respectively convert $\mathfrak{B}$ and h into a new model $\mathfrak{B}'$ of $\mathcal{K}$ and a functional bisimulation witnessing $(\mathfrak{B}', a^{\mathfrak{B}'}) \sim^f_{\mathcal{L},\Sigma} (\mathfrak{A}, b^{\mathfrak{A}})$, thus deriving a contradiction. Let k be the maximum over the $\mathcal{ALCI}$-outdegrees of the nodes in $\mathfrak{A}$. We define $\mathfrak{B}'$ from $\mathfrak{B}$ by removing all nodes d not in $\mathcal{D}^{\mathfrak{B}}$ from it and attaching the k-unfolding $\mathfrak{B}_d^{\leqslant k}$ to d, for any $d \in \mathcal{D}^{\mathfrak{B}}$. Now there is a functional $\mathcal{ALCI}(\Sigma)$-bisimulation f between $(\mathfrak{B}', a^{\mathfrak{B}'})$ and $(\mathfrak{A}, b^{\mathfrak{A}})$: to define f, extend h with, for every $d \in \mathsf{cons}(\mathcal{D}_\Sigma^{\mathfrak{B},a})$, functional bisimulations witnessing $(\mathfrak{B}_d^{\leqslant k}, d) \sim^f_{\mathcal{ALCI},\Sigma} (\mathfrak{A}, h(d))$.

($\Leftarrow$) Assume that (3) of Theorem 4.2 does not hold and let $\mathfrak{A}$ be an $\mathcal{ALCI}$-forest model of $\mathcal{K}$ of finite outdegree. Then, there exists $\mathfrak{B} \models \mathcal{K}$ and $a \in E^+$ 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 $\mathsf{dom}(\mathfrak{B})$ as a Σ-homomorphism h such that clearly $(\mathfrak{B}, c) \sim_{\mathcal{ALCI},\Sigma} (\mathfrak{A}, h(c))$ for all $c \in D$. Thus, $(\mathfrak{B}, a^{\mathfrak{B}}) \rightarrow_{D,\mathcal{ALCI},\Sigma} (\mathfrak{A}, b^{\mathfrak{A}})$. But then (2) of Lemma 4.14 does not hold for $\mathfrak{A}$. ⊣

Theorem 4.16 also applies to $(\mathcal{ALC}, \mathcal{ALCO})$-separability. We thus need an analogous notion of k-unfolding, adapted to the $\mathcal{ALCO}$ case, i.e. that does not create distinct copies of an element interpreting the same constant.

4.17. *Definition.* We define the *directed k-unfolding omitting Σ-individuals*, $\mathfrak{B}_d^{\Sigma,\leqslant k}$, of a model $\mathfrak{B}$ at $d \in \mathsf{dom}(\mathfrak{B})$ as follows, for any $k > 0$.

1. 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 $i_j \leqslant k$ for all $j \leqslant n$, where all $R_i \in \mathsf{rel}_2$ and $d_n \notin \{c^{\mathfrak{B}} : c \in \mathsf{cons} \cap \Sigma\}$ if $R_0, \ldots, R_{n-1} \in \Sigma$. Let $\mathsf{tail}(w) = d_n$.
2. $A^{\mathfrak{B}_d^{\Sigma,\leqslant k}} = \{w \in W : \mathsf{tail}(w) \in A^{\mathfrak{B}}\}$ for all $A \in \mathsf{rel}_1$.

3. $R^{\mathfrak{B}_d^{\Sigma,\leqslant k}} = \{(w_1, w_2) \in W^2 : \exists d \in \mathsf{dom}(\mathfrak{B}), i \leqslant k \text{ with } w_2 = w_1 R(d,i)\}$ for all $R \in \mathsf{rel}_2$.

4.18. Theorem. *Let $(\mathcal{K}, E^+, \{b\})$ be a labeled $\mathcal{ALC}$-knowledge base and $\Sigma \subseteq \mathsf{sig}(\mathcal{K})$ a signature. Then the following conditions are equivalent.*

1. *$(\mathcal{K}, E^+, \{b\})$ is projectively $\mathcal{ALCO}(\Sigma)$-separable;*
2. *$(\mathcal{K}, E^+, \{b\})$ is non-projectively* $\mathbf{UCQ}_r^{\mathcal{ALCO}}(\Sigma)$*-separable.*

Proof. Again it suffices to show that (3) of Theorem 4.2 and (2) of Lemma 4.14 are equivalent. For ($\Leftarrow$) the proof is as for $\mathcal{ALCI}$.

($\Rightarrow$) Assume first that (3) of Theorem 4.2 holds for $\mathfrak{A}$. We show that $\mathfrak{A}$ witnesses (2) of Lemma 4.14. Assume that for all $n > 0$ there exist $\mathfrak{B} \models \mathcal{K}$ and $a \in E^+$ such that for all $D \subseteq \mathsf{dom}(\mathfrak{B})$ of cardinality $\leqslant n$ such that the Σ-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 S denote the set of constants $c \in \mathsf{cons}(\mathcal{D}) \cap \Sigma$ such that there is an $\mathcal{ALC}(\Sigma)$-path from $b^{\mathfrak{A}}$ to $c^{\mathfrak{A}}$ in $\mathfrak{A}$. For any $c \in S$ let n_c be the length of the shortest such path and let $m = \sum_{c\in S} n_c |\mathsf{cons}(\mathcal{D})|$. Let $D_0 \subseteq \mathsf{dom}(\mathfrak{B})$ be minimal such that the Σ-reduct of $\mathfrak{B}_{|D}$ is rooted in $a^{\mathfrak{B}}$ and D_0 contains all $c^{\mathfrak{B}}$ with $c \in \mathsf{cons}(\mathcal{D}) \cap \Sigma$ such that there is an $\mathcal{ALC}(\Sigma)$-path from $a^{\mathfrak{B}}$ to $c^{\mathfrak{B}}$. As $(\mathfrak{B}, a^{\mathfrak{B}}) \sim_{\mathcal{ALCO},\Sigma} (\mathfrak{A}, b^{\mathfrak{A}})$, the individuals we obtain are exactly those in S and $|D_0| \leqslant \sum_{c\in R} n_c$. Obtain D from D_0 by adding all nodes $c^{\mathfrak{B}}$ with $c \in \mathsf{cons}(\mathcal{D})$ 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 $|D| \leqslant m$. Thus, we have $(\mathfrak{B}, a^{\mathfrak{B}}) \to_{D,\mathcal{ALCO},\Sigma} (\mathfrak{A}, b^{\mathfrak{A}})$. Let h be the Σ-homomorphism witnessing this. Now let k be the maximal $\mathcal{ALC}$-outdegree of a node in $\mathfrak{A}$ and define a model $\mathfrak{B}'$ from $\mathfrak{B}$ by

1. removing all nodes d not in $\mathcal{D}^{\mathfrak{B}}$,
2. attaching $\mathfrak{B}_d^{\Sigma,\leqslant k}$ to d for any $d \in \mathcal{D}^{\mathfrak{B}}$,
3. adding $(w, c^{\mathfrak{B}})$ to the interpretation of R if $(\mathsf{tail}(w), c^{\mathfrak{B}}) \in R^{\mathfrak{B}}$ and $c \in \Sigma$.

Then one can show that $(\mathfrak{B}', a^{\mathfrak{B}}) \sim^f_{\mathcal{ALCO},\Sigma} (\mathfrak{A}, b^{\mathfrak{A}})$, witnessed by the extension of h with the functional bisimulations witnessing $(\mathfrak{B}_d^{\Sigma,\leqslant k}, d) \sim^f_{\mathcal{ALCO}} (\mathfrak{A}, h(d))$ for every $d \in D$. ⊣

Theorem 4.18 does not hold when the ontology contains nominals:

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

§ 4.3. Conservative extensions

We now look into the computational complexity of projective restricted $(\mathcal{L},\mathcal{L})$-separability for $\mathcal{L} \in \mathbf{DL}_{\mathcal{IO}}$. While both complexity bounds usually followed from a link between separability and query evaluation in the full signature case, in the restricted case a lower bound is given by the link with deciding conservative extensions. Throughout this section we assume that for projective separability, only concept names are allowed as helper symbols. Recall that it does not impact separability if $\mathcal{L}$ does not contain nominals. In consequence, we also consider that sig only denotes the concept names occurring in a syntactic object, instead of the more general definition given in the Preliminaries.

4.20. *Definition.* 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 $\mathcal{O} \cup \mathcal{O}' \models C \sqsubseteq D$ implies $\mathcal{O} \models C \sqsubseteq D$ for all inclusions $C \sqsubseteq D$ of $\mathcal{L}$-concepts such that $\mathsf{sig}(C) \cup \mathsf{sig}(D) \subseteq \mathsf{sig}(\mathcal{O})$. *Projective conservative extensions in* $\mathcal{L}$ are defined in the same way except that we only require $(\mathsf{sig}(C) \cup \mathsf{sig}(D)) \cap \mathsf{sig}(\mathcal{O} \cup \mathcal{O}') \subseteq \mathsf{sig}(\mathcal{O})$. If $\mathcal{O} \cup \mathcal{O}'$ is not a conservative extension of $\mathcal{O}$ in $\mathcal{L}$, then there exists an $\mathcal{L}(\mathsf{sig}(\mathcal{O}))$-concept C 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 $\mathcal{O}'$.

4.21. Theorem. *Let $\mathcal{L} \in \mathbf{DL}_{\mathcal{IO}}$. Deciding (resp. projective) conservative extensions in $\mathcal{L}$ is polynomial-time reducible to the complement of restricted weak (resp. projective) $(\mathcal{L},\mathcal{L})$-separability.*

Proof. Assume $\mathcal{L}$-ontologies $\mathcal{O}$ and $\mathcal{O}'$ are given. Let

$$\begin{aligned}\mathsf{atom}_{\mathcal{O}} = \{A \mid A \in \mathsf{sig}(\mathcal{O}) \cap \mathsf{rel}_1\} \\ \cup \{\{a\} \mid a \in \mathsf{sig}(\mathcal{O}) \cap \mathsf{cons}\} \\ \cup \{\exists R.\top \mid R \in \mathsf{sig}(\mathcal{O})\}\end{aligned}$$

If $\mathcal{L}$ admits inverse roles, let it also contain $\exists R^{-}.\top$, for $R \in \mathsf{sig}(\mathcal{O})$. We may assume that there exists a concept name $A \in \mathsf{atom}_{\mathcal{O}}$ such that $\mathcal{O} \models A \equiv \neg C$ for some $C \in \mathsf{atom}_{\mathcal{O}}$. Indeed, if no such A exists, pick any $X \in \mathsf{atom}_{\mathcal{O}}$, add $A \sqsubseteq \neg X$, $X \sqsubseteq \neg A$ to $\mathcal{O}$ to obtain $\mathcal{O}_1$. 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 consider the case $\mathcal{L} = \mathcal{ALCO}$, which subsumes the case $\mathcal{L} = \mathcal{ALC}$, and whose development is easily adaptable to the $\mathcal{ALCI}(\mathcal{O})$ case. Recall the definition of relativization $C_{|A}$ to a concept name A of a concept C, and $\mathcal{O}_{|A}$ of an ontology (Def. 2.3). Observe that the relativization of an inclusion to a concept name

A is satisfied in $\mathfrak{A}$ whenever $A^{\mathfrak{A}} = \emptyset$. Fix A, D, D', a, b, S fresh (without loss of generality) and define

$$\begin{aligned}
\mathcal{O}^A &= \mathcal{O}_{|A} \cup \{\{c\} \sqsubseteq A \mid c \in \mathsf{sig}(\mathcal{O}) \cap \mathsf{cons}\} \\
&\quad \cup \{A \sqsubseteq \forall R.A \mid R \in \mathsf{sig}(\mathcal{O}) \cap \mathsf{rel}_2\} \\
(\mathcal{O} \cup \mathcal{O}')^{D'} &= (\mathcal{O} \cup \mathcal{O}')_{|D'} \cup \{D' \sqsubseteq \forall S.D'\} \\
&\quad \cup \{D' \sqsubseteq \forall R.D' \mid R \in \mathsf{sig}(\mathcal{O}) \cap \mathsf{rel}_2\} \\
\mathcal{O}^* &= \mathcal{O}^A \cup (\mathcal{O} \cup \mathcal{O}')^{D'} \\
&\quad \cup \{D \sqcap E \sqsubseteq D' \mid E \in \mathsf{atom}_{\mathcal{O}}\}
\end{aligned}$$

$$\mathcal{D} = \{A(b), D(a), S(a,c) \mid c \in \mathsf{cons} \cap \mathsf{sig}(\mathcal{O} \cup \mathcal{O}')\}$$

$$\mathcal{K} = (\mathcal{O}^*, \mathcal{D})$$

It now suffices to prove the following claim.

(∗) $\mathcal{O} \cup \mathcal{O}'$ is a (resp. projective) conservative extension of $\mathcal{O}$ in $\mathcal{ALCO}$ iff $(\mathcal{K}, \{a\}, \{b\})$ is not (resp. projectively) $\mathcal{ALCO}(\mathsf{sig}(\mathcal{O}))$-separable.

We consider the projective case. The non-projective case is similar and omitted. Consider an $\mathcal{ALCO}(\Sigma)$-concept C, where $\mathsf{sig}(\mathcal{O}) \subseteq \Sigma$ and $\Sigma \setminus \mathsf{sig}(\mathcal{O}) \subseteq \mathsf{rel}_1$. We use and prove the following three claims.

1. C is satisfiable w.r.t. $\mathcal{O}$ iff there exists $\mathfrak{B} \models \mathcal{K}$ such that $b^{\mathfrak{B}} \in C^{\mathfrak{B}}$.

Assume that C is satisfiable w.r.t. $\mathcal{O}$. let $\mathfrak{A} \models \mathcal{O}$ and $d \in C^{\mathfrak{A}}$. We define a model $\mathfrak{B}$ of $\mathcal{K}$ satisfying C in $b^{\mathfrak{B}}$: let $\mathsf{dom}(\mathfrak{B}) = \mathsf{dom}(\mathfrak{A}) \cup \{a\} \cup (\mathsf{cons} \cap (\mathsf{sig}(\mathcal{O}')) \setminus \mathsf{sig}(\mathcal{O}))$, $b^{\mathfrak{B}} = d$, $c^{\mathfrak{B}} = c$ for all $c \in \{a\} \cup (\mathsf{cons} \cap (\mathsf{sig}(\mathcal{O}') \setminus \mathsf{sig}(\mathcal{O})))$, $A^{\mathfrak{B}} = \mathsf{dom}(\mathfrak{A})$, $S^{\mathfrak{B}} = S^{\mathfrak{A}} \cup \{(a, c^{\mathfrak{A}}) \mid c \in \mathsf{cons} \cap (\mathsf{sig}(\mathcal{O}' \cup \mathcal{O}))\}$, $D^{\mathfrak{B}} = \{a\}$ and $D'^{\mathfrak{B}} = \emptyset$. On everything else let $\mathfrak{A}$ be unchanged. Then $\mathfrak{B}$ is a model of $\mathcal{K}$ satisfying C in $b^{\mathfrak{B}}$. The converse direction of (1) is clear.

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

Suppose that C is satisfiable w.r.t. $\mathcal{O} \cup \mathcal{O}'$, witnessed by the pointed model $(\mathfrak{A}, d)$. We define a model $\mathfrak{B}$ of $\mathcal{K}$ satisfying C in $a^{\mathfrak{B}}$: set $\mathsf{dom}(\mathfrak{B}) = \mathsf{dom}(\mathfrak{A})$. We interpret b arbitrarily and set $A^{\mathfrak{B}} = D^{\mathfrak{B}} = D'^{\mathfrak{B}} = \mathsf{dom}(\mathfrak{B})$. Finally, let $S^{\mathfrak{B}} = S^{\mathfrak{A}} \cup \{(a, c^{\mathfrak{A}}) \mid c \in \mathsf{cons} \cap \mathsf{sig}(\mathcal{O} \cup \mathcal{O}')\}$. Then $\mathfrak{B}$ is a model of $\mathcal{O} \cup \mathcal{O}'$ satisfying C in $a^{\mathfrak{B}}$.

3. Let $E \in \mathsf{atom}_{\mathcal{O}}$. 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. $\mathcal{O} \cup \mathcal{O}'$.

Let $E \in \mathsf{atom}_{\mathcal{O}}$ 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. $(\mathcal{O} \cup \mathcal{O}')^{\mathcal{D}'}$.

We now prove $(*)$.

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

($\Leftarrow$) 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 $\mathsf{atom}_{\mathcal{O}}$, there exists $E \in \mathsf{atom}_{\mathcal{O}}$ 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 (1), there exists a model $\mathfrak{A}$ of $\mathcal{K}$ such that $b^{\mathfrak{A}} \in (E \sqcap C)^{\mathfrak{A}}$ and, by (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 $\mathcal{ALCO}(\mathsf{sig}(\mathcal{O}))$-separable, namely by $\neg(E \sqcap C)$.

To adapt the proof to the presence of inverse roles, it suffices to add $A \sqsubseteq \forall R^{-}.A$ to $\mathcal{K}^{A}$ for any $R \in \mathsf{sig}(\mathcal{O}) \cap \mathsf{rel}_2$. ⊣

The lower bound for restricted weak separability immediately follows for $\mathcal{ALC}$ and $\mathcal{ALCI}$: it is known that deciding (non-projective) conservative extensions in $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$ is 2Exp-hard [GLW06, LWW07] and that conservative extensions and projective conservative extensions coincide in logics that enjoy the Craig Interpolation Property [JLMSW17], which $\mathcal{ALC}$ and $\mathcal{ALCI}$ do. For $\mathcal{ALCO}$, which does not (consider the inclusion $\{a\} \sqcap \exists R.\{a\} \sqsubseteq \{b\} \rightarrow \exists R.\{b\}$), it is proved in [JLPW21] that deciding projective conservative extensions is 3Exp-complete and that deciding non-projective conservative extensions is 3Exp-hard. The upper bound is still open.

4.22. Corollary.

1. *Weak (projective) restricted* $(\mathcal{ALC}, \mathcal{ALC})$ *and* $(\mathcal{ALCI}, \mathcal{ALCI})$ *separability are* 2Exp*-hard in combined complexity.*
2. *Weak (projective) restricted* $(\mathcal{ALCO}, \mathcal{ALCO})$*-separability is* 3Exp*-hard in combined complexity.*

§ 4.4. 2Exp upper bound for $\mathcal{ALC}(\mathcal{I})$

We then focus on the upper bound for projective separability. We concentrate on $\mathcal{ALCI}$; the case of $\mathcal{ALC}$ is similar and simpler. The idea is to use two-way alternating tree automata (2ATA) [V98] to decide (3) of Theorem 4.2. More precisely, given $(\mathcal{K}, E^{+}, \{b\})$ and Σ, we construct a 2ATA $\mathcal{A}$ such that the language recognized by $\mathcal{A}$ is non-empty iff there is an $\mathcal{ALCI}$-forest model $\mathfrak{A}$ of $\mathcal{K}$ as described in (3) of Theorem 4.2. A proof of similar nature gives a 3EXP upper bound for $(\mathcal{ALCO}, \mathcal{ALCO})$ in the projective case [JLPW21].

4.4.1. *2ATAs and proof strategy*

4.23. *Definition.* Let a *tree* be 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 $i \in \mathbb{N}$. 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 Θ, a Θ-*labeled tree* is a pair (T, τ) 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*, Θ is the finite *input alphabet*, $q_0 \in Q$ is the *initial state*, δ 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 true and false and transitions of the form q, $\langle - \rangle q$, $[-]q$, $\Diamond q$, $\Box q$ where $q \in Q$.

The semantics are defined in terms of runs in the usual way [V98]: let $\mathcal{A} = (Q, \Theta, q_0, \delta, \Omega)$ be a 2ATA and (T, τ) a Θ-labeled tree. A *run for $\mathcal{A}$ on (T, τ)* is a $T \times Q$-labeled tree (T_r, r) such that $\varepsilon \in T_r$, $r(\varepsilon) = (\varepsilon, q_0)$ and for all $y \in T_r$ with $r(y) = (x, q)$ and $\delta(q, \tau(x)) = \varphi$, there is an assignment v of truth values to the transitions in φ such that v satisfies φ and

1. if $v(p) = 1$, then $r(y') = (x, p)$ for some successor y' of y in T_r;
2. if $v(\langle - \rangle p) = 1$, then $x \neq \varepsilon$ and there is a successor y' of y in T_r with $r(y') = (x \cdot -1, p)$;
3. if $v([-]p) = 1$, then $x = \varepsilon$ or there is a successor y' of y in T_r such that $r(y') = (x \cdot -1, p)$;
4. 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 $r(y') = (x', p)$;
5. 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 $j \geqslant 0$, with q_j the state such that $r(i_0 \cdots i_j) = (x, q_j)$. The path γ 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. We use $L(\mathcal{A})$ to denote the set of all Θ-labeled binary trees accepted by $\mathcal{A}$.

4.24. Theorem ([V98]). *The* emptiness problem*, which asks whether $L(\mathcal{A}) = \emptyset$ for a given 2ATA $\mathcal{A}$, can be decided in exponential time in the number of states of $\mathcal{A}$.*

The use of tree automata is enabled by the fact that (3) of Theorem 4.2 refers to *forest models* of $\mathcal{K}$. Forest models can be encoded in labeled trees using an appropriate alphabet. Intuitively, each node in the tree corresponds to an element in the forest model and the label contains its type, the connection to its predecessor, and connections to individuals from $\mathcal{D}$.

We first devise a 2ATA $\mathcal{B}$ (of polynomial size) that recognizes the finite outdegree forest models of $\mathcal{K}$. It then suffices to construct, for each $a \in E^+$, a 2ATA $\mathcal{A}_a$ such that $\mathcal{A}_a$ accepts $\mathfrak{A}$ iff

$(*_a)$ there exists $\mathfrak{B} \models \mathcal{K}$ such that $(\mathfrak{B}, a^{\mathfrak{B}}) \sim^f_{\mathcal{ALCI},\Sigma} (\mathfrak{A}, b^{\mathfrak{A}})$.

Indeed, a 2ATA that recognizes the language $L(\mathcal{B}) \cap \bigcap_{a\in E^+} \overline{L(\mathcal{A}_a)}$ is as required, where $\overline{L}$ denotes the complement of L:

$L(\mathcal{B}) \cap \bigcap_{a\in E^+} \overline{L(\mathcal{A}_a)} = \emptyset$ iff $(\mathcal{K}, E^+, \{b\})$ is not projectively $\mathcal{ALCI}(\Sigma)$-separable.

We obtain the desired 2ATA $\mathcal{A}$ from $\mathcal{B}$ and the $\mathcal{A}_a$, as complementation and intersection of 2ATAs involve only a polynomial blowup:

4.25. Lemma (Folklore). *Given 2ATAs $\mathcal{A}_1, \mathcal{A}_2$, we can compute in polynomial time a 2ATA*

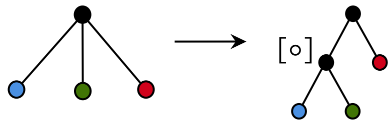
1. *$\overline{\mathcal{A}_1}$ such that $L(\overline{\mathcal{A}_1}) = \overline{L(\mathcal{A}_1)}$ and the number of states of $\overline{\mathcal{A}_1}$ equals the number of states of $\mathcal{A}_1$;*
2. *$\mathcal{A}$ such that $L(\mathcal{A}) = L(\mathcal{A}_1) \cap L(\mathcal{A}_2)$ and the number of states of $\mathcal{A}$ is $1+n_1+n_2$, n_i the number of states of $\mathcal{A}_i$.*

4.4.2. *Encoding models as input*

In order to work with tree automata, we need to encode $\mathcal{ALCI}$-forest models of $\mathcal{K}$ of finite $\mathcal{ALCI}$-outdegree as input to the tree automata. Since 2ATAs run over binary trees, we need to appropriately encode the arbitrary outdegree. More precisely, we use the alphabet Θ defined by

$$\Theta = \{\circ\} \cup (\mathsf{rel}_2^- \cap \mathsf{sig}(\mathcal{K})) \times 2^{\mathsf{cons} \cup (\mathsf{sig}(\mathcal{K}) \cap \mathsf{rel}_1)} \times 2^{\mathcal{F}},$$

where $\mathcal{F} = (\mathsf{rel}_2^- \cap \mathsf{sig}(\mathcal{K})) \times \mathsf{cons}(\mathcal{D})$. 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 $\mathsf{cons}(\mathcal{D})$ 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 ∘-labeled nodes as needed to achieve the required outdegree at each node, as depicted below.



Additional conditions are required from trees to be able to represent forest models. We call such trees *well-formed*:

4.26. *Definition.* Let (T, τ) be a Θ-labeled tree. We call (T, τ) *well-formed* if

1. for every $a \in \mathsf{cons}(\mathcal{D})$, 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.

4.27. *Remark.* 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. A well-formed Θ-labeled tree (T, τ) gives rise to a model $\mathfrak{A}_\tau$ with $\mathsf{dom}(\mathfrak{A}_\tau) = \{w \in T \mid \tau(w) \neq \circ\}$ 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 $a \in \mathsf{cons}$, $A \in \mathsf{rel}_1$, $R \in \mathsf{rel}_2$, and where w_a denotes an arbitrary element of $\mathfrak{A}$ if $a \in \mathsf{cons} \setminus \mathsf{cons}(\mathcal{D})$.

Conversely,

4.28. Lemma. *Every $\mathcal{ALCI}$-forest model $\mathfrak{A} \models \mathcal{K}$ of finite $\mathcal{ALCI}$-outdegree can be encoded (up to isomorphism) as a well-formed Θ-labeled tree.*

Proof. We start with a not-necessarily binary well-formed Θ-labeled tree (T, τ) that encodes $\mathfrak{A}$; (T, τ) can easily be made binary by introducing intermediate ∘-labeled nodes. Let $D = \{a^{\mathfrak{A}} \mid a \in \mathsf{cons}(\mathcal{D})\}$ and associate sets M_d, F_d to every element $d \in \mathsf{dom}(\mathfrak{A})$ by taking

$$M_d = \{A \in \mathsf{sig}(\mathcal{K}) \mid d \in A^{\mathfrak{A}}\} \cup \{a \in \mathsf{cons}(\mathcal{D}) \mid d = a^{\mathfrak{A}}\}$$
$$F_d = \{(R, e) \mid (d, e) \in R^{\mathfrak{A}}, e^{\mathfrak{A}} \in D\}$$

To start the construction of (T, τ), we set $\tau(\varepsilon) = \circ$ and add a successor w_d of ε for every $d \in D$ and label it with $\tau(w_d) = (S, M_d, F_d)$ for an arbitrary fresh 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 an $\mathcal{L}$-forest model of $\mathcal{K}$. Recall that $\mathfrak{A}_d$ is rooted at d. Now (T, τ) 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 $(e, f) \in R^{\mathfrak{A}}$.

⊣

4.4.3. *Recognising adequate models*

By intersection, we first construct a 2ATA that recognises precisely the forest models of $\mathcal{K}$. First, one that recognises the well-formed trees, then one that recognises models of $\mathcal{D}$ among well-formed trees, and finally one that recognises finally models of $\mathcal{O}$ among well-formed trees.

4.29. Lemma. *For any $\mathcal{ALCI}$-knowledge base $\mathcal{K}$, we can construct in time polynomial in $||\mathcal{K}||$ a 2ATA $\mathcal{A}_0$ that recognises exactly the well-formed Θ-labeled trees.*

Proof. We define the 2ATA $\mathcal{A}_0 = (Q, \Theta, q_0, \delta, \Omega)$ as follows. Let Ω send every state to 0, which implies by definition that any infinite path in any run is accepting. Let $Q = \{q_0\} \cup \{q_a \mid a \in \mathsf{cons}(\mathcal{D})\}$. For all $a, b \in \mathsf{cons}(\mathcal{D})$ and $\theta \in \Theta$,

$$\delta(q_0, \theta) = \begin{cases} \Box q_a & \text{if } \theta = (R, M, F) \text{ and } a \in M \\ \Box q_0 & \text{if } \theta = \circ \\ \bot & \text{otherwise.} \end{cases}$$

$$\delta(q_a, \theta) = \begin{cases} \bot & \text{if } \theta = (R, M, F) \text{ and } b \in M \\ \Box q_a & \text{otherwise.} \end{cases}$$

Let (T, τ) be a Θ-labeled tree. If (T, τ) is well-formed, it is straightforward to construct an accepting run (T_r, r) of $\mathcal{A}$ on (T, τ), considering that the nodes w_a (using notation from Def 4.26) can only be descendants of $\circ$-labeled nodes and are unique for each $a \in \mathsf{cons}(\mathcal{D})$. If (T, τ) is not well-formed, it is also routinely checked that any run (T_r, r) of $\mathcal{A}$ on (T, τ) would need to contain a node x such that $r(x) = (w, \bot)$ for some $w \in T$, which would contradict the existence of the run itself. ⊣

4.30. Lemma. *There exists a 2ATA $\mathcal{A}_\mathcal{D}$ that recognises a well-formed Θ-labeled tree (T, τ) iff $\mathfrak{A}_\tau \models \mathcal{D}$.*

Proof. Let $w \in T$ and $(R, M, F) = \tau(w)$ such that there exists $a \in \mathsf{cons}(\mathcal{D}) \cap M$. Denote by (C_a) the condition stating that for all $A \in \mathsf{rel}_1, R \in \mathsf{rel}_2$ and $b \in \mathsf{cons}(\mathcal{D})$, $A(a) \in \mathcal{D}$ implies $A \in M$, $R(a, b) \in \mathcal{D}$ implies $(R, b) \in F$ and $R(b, a) \in \mathcal{D}$ implies $(R^-, b) \in F$. We define the 2ATA $\mathcal{A}_\mathcal{D} = (Q, \Theta, q_0, \delta, \Omega)$ as follows. Let Ω send every state to 0. Let $Q = \{q_0\}$ and for all $(R, M, F) \in \Theta$,

$$\delta(q_0, \circ) = \Box q_0$$

$$\delta(q_0, (R, M, F)) = \begin{cases} \top & \text{if } \exists a \text{ s.t. } \{a\} = \mathsf{cons}(\mathcal{D}) \cap M \text{ and } (C_a) \\ \bot & \text{otherwise.} \end{cases}$$

⊣

Finally, we omit the cumbersome (but of similar essence) proof that one can recognise models of $\mathcal{O}$.

4.31. Lemma. *Let $\mathcal{O}$ be an ontology, that we may assume w.l.o.g. such that $\mathcal{O} = \{\top \sqsubseteq C\}$ for some $\mathcal{ALCI}$ concept C in negation normal form. There exists a 2ATA $\mathcal{A}_\mathcal{O}$ that recognises a well-formed Θ-labeled tree (T, τ) iff $\mathfrak{A}_\tau \models \mathcal{O}$.*

The final block of the proof is now to construct, for any given $a \in E^+$, a 2ATA $\mathcal{A}_a$ that recognises exactly well-formed trees corresponding to models satisfying $(*_a)$. We first replace $(*_a)$ with an equivalent condition for which we are able to devise an appropriate 2ATA.

4.32. *Definition.* For any pointed database $(\mathcal{D}, a)$, we write $(\mathcal{D}_a, a) \to_c^\Sigma (\mathfrak{A}, b^\mathfrak{A})$ if there is a Σ-homomorphism h from the maximal Σ-connected component $\mathcal{D}_a$ 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 $d \in \mathsf{cons}(\mathcal{D}_a)$ such that

(i) there exists $\mathfrak{B}_d \models \mathcal{O}$ with $\mathsf{tp}_\mathcal{K}(\mathfrak{B}_d, d) = t_d$ and $(\mathfrak{B}_d, d) \sim_{\mathcal{ALCI},\Sigma} (\mathfrak{A}, h(d))$;
(ii) $(\mathcal{O}, \mathcal{D}')$ is satisfiable, where $\mathcal{D}' = \mathcal{D} \cup \{C(d) \mid C \in t_d,\ d \in \mathsf{cons}(\mathcal{D}_a)\}$.

We consider an extended notion of database in (ii), *i.e.* any set of the form $\mathcal{D} \cup \{C_1(a_1), \dots, C_n(a_n)\}$ where $a_1, \dots, a_n \in \mathsf{cons}$ and $C_1, \dots, C_n$ are $\mathcal{ALCI}$-concepts. The semantics of extended databases is defined in the expected way.

4.33. Lemma. *For all $\mathcal{ALCI}$-forest models $\mathfrak{A} \models \mathcal{K}$ and $a \in E^+$, condition $(*_a)$ is equivalent to $(\mathcal{D}_a, a) \to_c^\Sigma (\mathfrak{A}, b^\mathfrak{A})$.*

Proof.

($\Leftarrow$) Let $\mathfrak{A} \models \mathcal{K}$ be a forest model and $a \in E^+$ with $(\mathcal{D}_a, a) \to_c^\Sigma (\mathfrak{A}, b^\mathfrak{A})$, witnessed by some Σ-homomorphism h and $\mathcal{K}$-types t_d, $d \in \mathsf{cons}(\mathcal{D})$. For each d let $\mathfrak{B}_d \models \mathcal{O}$ such that $(\mathfrak{B}_d, d) \sim_{\mathcal{ALCI},\Sigma} (\mathfrak{A}, h(d))$. We may assume without loss of generality that the $\mathfrak{B}_d$ are tree-shaped with root d and that the bisimulations are functions f_d (because of tree-shapedness). We then define $\mathfrak{B}$ by attaching $\mathfrak{B}_d$ to $\mathcal{D}_a$ at d for each $d \in \mathsf{cons}(\mathcal{D}_a)$ and taking the disjoint union with a model of $(\mathcal{O}, \mathcal{D} \setminus \mathcal{D}_a)$. Then $\mathfrak{B} \models \mathcal{K}$ and $f = \bigcup_{d \in \mathsf{cons}(\mathcal{D}_a)} f_d$ is a functional $\mathcal{ALCI}(\Sigma)$-bisimulation between $\mathfrak{B}$ and $\mathfrak{A}$.

($\Rightarrow$) Let $\mathfrak{A}, \mathfrak{B} \models \mathcal{K}$ and $a \in E^+$ such that $(\mathfrak{B}, a^\mathfrak{B}) \sim^f_{\mathcal{ALCI},\Sigma} (\mathfrak{A}, b^\mathfrak{A})$, witnessed by some function $\mathcal{ALCI}(\Sigma)$-bisimulation f. Then the restriction $f_{|\mathsf{cons}(\mathcal{D})}$ and types $t_d = \mathsf{tp}_\mathcal{K}(\mathfrak{B}, d^\mathfrak{B})$, for all $d \in \mathsf{cons}(\mathcal{D})$ witness $(\mathcal{D}_a, a) \to_c^\Sigma (\mathfrak{A}, b^\mathfrak{A})$. ⊣

It only remains to show that for each $a \in E^+$, there exists a 2ATA $\mathcal{A}_a$ such that $\mathcal{A}_a$ accepts a well-formed Θ-labeled tree (T, τ) iff $(\mathcal{D}_a, a) \to_c^\Sigma (\mathfrak{A}, b^\mathfrak{A})$. Moreover, $\mathcal{A}_a$ can be constructed in double exponential time in $||\mathcal{K}||$ and has exponentially many states. See [JLPW21] for a sketch of the construction and proof of correctness. The 2EXP upper bound follows as non-emptiness can be decided in exponential time.

§ 4.5. UNDECIDABILITY RESULTS

In [JLPW21], a 3EXP upper bound for weak restricted projective $(\mathcal{ALCO}, \mathcal{ALCO})$-separability is provided assuming constants are not allowed as helper symbols, matching the 3EXP lower bound given in 4.22 by the reduction from deciding projective conservative extensions. We now show that if constants are allowed, complexity goes from 3EXP-complete to undecidable, and that undecidability holds even with $\mathcal{ALC}$ ontologies. A rather direct corollary of that proof is that for any $\mathcal{L}$ that contains $\mathcal{ALCFIO}$, weak restricted projective and non-projective $(\mathcal{ALC}, \mathcal{L})$-separability are also undecidable. Note that to prove this, we could not use the fact that deciding conservative extensions in $\mathcal{ALCFIO}$ is undecidable [LWW07] as the relativization strategy used previously fails in this case.

4.34. Theorem. *Restricted weak projective $(\mathcal{ALC}, \mathcal{ALCO})$-separability is undecidable if constants are admitted as helper symbols.*

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

4.35. *Definition.* A *finite rectangular 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, τ) where $n, m \geqslant 1$ and $\tau : \{0, \dots, n\} \times \{0, \dots, m\} \to \mathcal{T}$ such that the following hold.

$$
\begin{aligned}
&(\tau(i,j), \tau(i+1,j)) \in H && \text{for all } i < n \text{ and } j \leqslant m,\\
&(\tau(i,j), \tau(i,j+1)) \in V && \text{for all } i \leqslant n \text{ and } j < m,\\
&\tau(0,j) \in L \text{ and } \tau(n,j) \in R && \text{for all } 0 \leqslant j \leqslant m,\\
&\tau(i,0) \in B \text{ and } \tau(i,m) \in T && \text{for all } 0 \leqslant i \leqslant n.
\end{aligned}
$$

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

4.36. *Definition.* Let $S = (\mathcal{T}, H, V, R, L, T, B)$ be a finite rectangular tiling system. Let the ontology $\mathcal{O}_S$ consist of the following inclusions.

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

$$
\begin{aligned}
\top &\sqsubseteq \bigsqcup_{t \in T} (t \sqcap \bigsqcap_{t' \in T,\, t' \neq t} \neg t')\\
\top &\sqsubseteq \bigsqcap_{t \in T} (t \to (\bigsqcup_{(t,t') \in H} \forall R_x.t' \sqcap \bigsqcup_{(t,t') \in V} \forall R_y.t'))
\end{aligned}
$$

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

$$
\begin{aligned}
\mathsf{bottom} &\sqsubseteq \neg\mathsf{top} \sqcap \forall R_x.\mathsf{bottom} \sqcap \bigsqcup_{t \in B} t\\
\mathsf{right} &\sqsubseteq \forall R_y.\mathsf{right} \sqcap \bigsqcup_{t \in R} t\\
\mathsf{left} &\sqsubseteq \neg\mathsf{right} \sqcap \forall R_y.\mathsf{left} \sqcap \bigsqcup_{t \in L} t\\
\mathsf{top} &\sqsubseteq \forall R_x.\mathsf{top} \sqcap \bigsqcup_{t \in T} t\\
\neg\mathsf{top} &\equiv \exists R_y.\top\\
\neg\mathsf{right} &\equiv \exists R_x.\top
\end{aligned}
$$

3. There is an infinite outgoing R_x/R_y-path starting at Q or some grid cell

does not close in the part of the model reachable from Q:

$$Q \sqsubseteq \exists R_x.Q \sqcup \exists R_y.Q \sqcup (\exists R_x.\exists R_y.P \sqcap \exists R_y.\exists R_x.\neg P)$$

4. Q is triggered by $A_1 \sqcap D$:

$$A_1 \sqcap D \sqsubseteq Q$$

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

$$\mathcal{D}_S = \{A_1(a), Y(b), D(o), \mathsf{left}(o), \mathsf{bottom}(o)\}$$
$$\Sigma = \{o, R_x, R_y, \mathsf{left}, \mathsf{right}, \mathsf{top}, \mathsf{bottom}\}$$

4.37. Lemma. *If the tiling system S has a solution, then $(\mathcal{K}_S, \{a\}, \{b\})$ is projectively $\mathcal{ALCO}(\Sigma)$-separable, using constants as helper symbols.*

Proof. Let τ be a solution for S, of dimensions n, m. We show the labeled knowledge base $(\mathcal{K}_S, \{a\}, \{b\})$ is separated by $\neg G$, where G is an $\mathcal{ALCO}(\Sigma \cup \mathsf{cons} \setminus \mathsf{cons}(\mathcal{K}))$ concept defined in the following way, for all $i \leqslant n-1, j \leqslant m-1$. Informally, satisfying G means being the bottom left corner of an $n \times m$ grid, regardless of tiles.

$$G = \{o\} \sqcap G_{0,0}$$
$$G_{n,m} = \{a_{n,m}\} \sqcap \bigsqcap_{\substack{i' \leqslant n, j' \leqslant m \\ (i',j') \neq (n,m)}} \neg\{a_{i',j'}\} \sqcap \mathsf{right} \sqcap \mathsf{top}$$
$$G_{n,j} = \{a_{n,j}\} \sqcap \bigsqcap_{\substack{i' \leqslant n, j' \leqslant m \\ (i',j') \neq (n,j)}} \neg\{a_{i',j'}\} \sqcap \exists R_y.G_{i,j+1} \sqcap \forall R_y.G_{i,j+1} \sqcap \mathsf{right}$$
$$G_{i,m} = \{a_{i,m}\} \sqcap \bigsqcap_{\substack{i' \leqslant n, j' \leqslant m \\ (i',j') \neq (i,m)}} \neg\{a_{i',j'}\} \sqcap \exists R_x.G_{i+1,j} \sqcap \forall R_x.G_{i+1,j} \sqcap \mathsf{top}$$
$$G_{i,j} = \{a_{i,j}\} \sqcap \bigsqcap_{\substack{i' \leqslant n, j' \leqslant m \\ (i',j') \neq (i,j)}} \neg\{a_{i',j'}\} \sqcap \exists R_x.G_{i+1,j} \sqcap \forall R_x.G_{i+1,j} \sqcap \exists R_y.G_{i,j+1} \sqcap \forall R_y.G_{i,j+1}$$

1) We first show that $\mathcal{K}_S \models \neg G(a)$. Let $\mathfrak{A} \models G(a)$. Then, $\mathfrak{A} \models \{o\}(a)$, so if $\mathfrak{A} \models \mathcal{K}$ then $\mathfrak{A} \models (A_1 \sqcap D)(a)$ and consequently $\mathfrak{A} \models Q(a)$. Then, by definition of $\mathcal{O}(S)$, either the R_x/R_y connected component of $a^{\mathfrak{A}}$ in $\mathfrak{A}$ contains an infinite path (of elements satisfying Q), or it contains an element satisfying $\exists R_x.\exists R_y.P \sqcap \exists R_y \exists R_x.\neg P$. Both options contradict $\mathfrak{A} \models G(a)$: in any model $\mathfrak{B}$ of G, the connected component of $o^{\mathfrak{B}}$ for the signature $\sigma = \{R_x, R_y, \{a_{i,j}\}_{i \leqslant n, j \leqslant m}\}$ is uniquely determined up to σ-isomorphism by the structure depicted

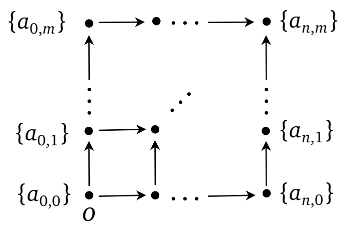
2) We then show that $\mathcal{K}_S \not\models \neg G(b)$. We define a model $\mathfrak{A}$ of $\mathcal{K}_S$ and $G(b)$.

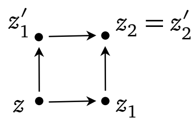
$$\begin{aligned}\mathsf{dom}(\mathfrak{A}) &= \{a_{i,j} : (i,j) \in \{0,\dots,n\} \times \{0,\dots,m\}\}\\ R_x^{\mathfrak{A}} &= \{(a_{i,j}, a_{i+1,j}) : (i,j) \in \{0,\dots,n-1\} \times \{0,\dots,m\}\}\\ R_y^{\mathfrak{A}} &= \{(a_{i,j}, a_{i,j+1}) : (i,j) \in \{0,\dots,n\} \times \{0,\dots,m-1\}\}\end{aligned}$$

Let $o^{\mathfrak{A}} = b^{\mathfrak{A}} = a_{0,0} \neq a^{\mathfrak{A}}$. Let $A_1^{\mathfrak{A}} = \{a^{\mathfrak{A}}\}$, $D^{\mathfrak{A}} = \{b^{\mathfrak{A}}\}$, $a_{i,j}^{\mathfrak{A}} = a_{i,j}$ for all $(i,j) \in \{0,\dots,n\} \times \{0,\dots,m\}$.

$$\begin{aligned}\mathsf{left}^{\mathfrak{A}} &= \{a_{0,j} : 0 \leqslant j \leqslant m\}\\ \mathsf{right}^{\mathfrak{A}} &= \{a_{n,j} : 0 \leqslant j \leqslant m\}\\ \mathsf{top}^{\mathfrak{A}} &= \{a_{i,m} : 0 \leqslant i \leqslant n\}\\ \mathsf{bottom}^{\mathfrak{A}} &= \{a_{i,0} : 0 \leqslant i \leqslant n\}\end{aligned}$$

For every $t \in T$ let $t^{\mathfrak{A}} = \{a_{i,j} : (i,j) \in \tau^{-1}(t)\}$. Clearly $G(b)$, $\mathcal{D}(S)$ are satisfied. The tiling axioms of $\mathcal{O}(S)$ are satisfied as τ is a solution for S. The axioms involving Q, P are satisfied, as $Q^{\mathfrak{A}} \subseteq (A_1 \sqcap D)^{\mathfrak{A}} = \emptyset$. $\dashv$





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

Proof. Let $\mathfrak{A} \models \mathcal{K}_S$. Suppose $b^{\mathfrak{A}} \neq o^{\mathfrak{A}}$. Let $\mathsf{dom}(\mathfrak{B}) = \mathsf{dom}(\mathfrak{A})$, $a^{\mathfrak{B}} = b^{\mathfrak{B}} = b^{\mathfrak{A}}$, $E^{\mathfrak{B}} = E^{\mathfrak{A}}$ for every concept name E, $R^{\mathfrak{B}} = R^{\mathfrak{A}}$ for every role name R. Then it is clear that $\mathfrak{B}$ is a model of $\mathcal{K}_S$ and that $(\mathfrak{A}, b^{\mathfrak{A}})$ is Σ-isomorphic to $(\mathfrak{B}, a^{\mathfrak{B}})$. Now suppose $b^{\mathfrak{A}} = o^{\mathfrak{A}}$. We define an intermediary model $\mathfrak{B}'$ as follows. Let $\mathsf{dom}(\mathfrak{B}') = \mathsf{dom}(\mathfrak{A})$, $a^{\mathfrak{B}'} = b^{\mathfrak{B}'} = b^{\mathfrak{A}}$, $E^{\mathfrak{B}'} = E^{\mathfrak{A}}$ for every concept name E, $R^{\mathfrak{B}'} = R^{\mathfrak{A}}$ for every role name $R \notin \{Q, P\}$ and $Q^{\mathfrak{B}'} = P^{\mathfrak{B}'} = \emptyset$. Let $\mathfrak{B}$ be the model obtained from $\mathfrak{B}'$ after applying the following algorithm: start with $\mathfrak{B} = \mathfrak{B}'$. If there exists an infinite R_x chain from $o^{\mathfrak{B}}$ then update $Q^{\mathfrak{B}}$ by adding every element of that chain. Else, if there exists an infinite R_y chain from $o^{\mathfrak{B}}$ then update $Q^{\mathfrak{B}}$ by adding every element of that chain. If there are no such chains, then there exists an R_x chain $(o^{\mathfrak{B}}, a_{1,0}, \dots, a_{n,0})$ of maximal length and an R_y chain $(o^{\mathfrak{B}}, a_{0,1}, \dots, a_{0,m})$ of maximal length in $\mathfrak{B}$. In that case, let $a_{0,0} = o^{\mathfrak{B}}$ and

```
i ← 1, j ← 1
  while P^𝔅 = ∅ and j ⩽ m,
    while P^𝔅 = ∅ and i ⩽ n,
      if ∃d ≠ d' with (a_{i−1,j}, d) ∈ R_x^𝔅 and (a_{i,j−1}, d) ∈ R_y^𝔅,
           P^𝔅 ← P^𝔅 ∪ {d}
      otherwise, pick d s.t.  (a_{i−1,j}, d) ∈ R_x^𝔅 and (a_{i,j−1}, d) ∈ R_y^𝔅
           d ← a_{i,j}
           i ← i + 1 if i < n
    j ← j + 1 if j < m and i = n
    i ← 1
```

If $i = n$ and $j = m$, then the restriction of $\mathfrak{B}$ to $\{a_{i,j} : 0 \leqslant i \leqslant n, 0 \leqslant j \leqslant m\}$ induces a solution for S, given by the tiling function mapping each (i, j) to the unique $t \in T$ such that $a_{i,j} \in t^{\mathfrak{B}}$. As we assumed S had no solution, (i, j) then cannot reach the value (n, m) after termination of the algorithm. So, either there exists an infinite R_x (or R_y) chain starting from $o^{\mathfrak{B}}$, or $P^{\mathfrak{B}} \neq \emptyset$. In the latter case, by definition of the algorithm, there exist $a_{i,j}$ and d, d' such that $d \neq d'$, $(a_{i+1,j}, d) \in R_y^{\mathfrak{B}}$, $(a_{i,j+1}, d') \in R_x^{\mathfrak{B}}$, $d \in P^{\mathfrak{B}}$, $d' \notin P^{\mathfrak{B}}$. Therefore, $\mathfrak{B}$ satisfies the axioms of $\mathcal{K}_S$ involving Q, P. The tiling axioms of $\mathcal{K}_S$ are also satisfied by $\mathfrak{B}$, as they are by $\mathfrak{A}$ and as $(\mathfrak{A}, b^{\mathfrak{A}}) \simeq_\Sigma (\mathfrak{B}, a^{\mathfrak{B}})$. The Σ-isomorphism follows from $\mathfrak{B}$'s construction and the fact that $Q, P \notin \Sigma$. ⊣

The same tiling problem as for $\mathcal{ALCO}$ can be used to show undecidability of separability with any extension of $\mathcal{ALCFIO}$ under $\mathcal{ALC}$-ontologies. By adding the letter $\mathcal{F}$ we denote the extension of a DL with unqualified number restrictions of the form $\leqslant 1.R.\top$, expressing functionality.

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

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. 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. Let $G = F \sqcap E$, where F is the obvious concept stating that $(\leqslant 1\ R)$ holds for $R \in \{R_x, R_y, R_y^-, R_x^-\}$ for all nodes reachable in no more than $2(n+m)$ steps along roles R_x, R_x, R_y, R_x^-, and

$$E = \{o\} \sqcap \forall R_x^{n+1}.\bot \sqcap \forall R_x^{\leqslant n}.\mathsf{bottom} \sqcap \forall R_y^{m+1}.\bot \sqcap \forall R_y^{\leqslant m}.\mathsf{left} \sqcap \bigsqcap_{\substack{w \in \{R_x, R_y\}^* \\ |w|_{R_x} < n, |w|_{R_y} < m}} \exists (w \cdot R_x R_y R_x^- R_y^- \cdot \overleftarrow{w}).\{o\}$$

It is readily checked that G indeed enforces a grid, as announced. We show that $\mathcal{K}_S \models \neg G(a)$ and $\mathcal{K}_S \not\models \neg G(b)$, 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. ⊣

To obtain the desired reduction, we only need the implication given in the above lemma, as its converse already follows from Lemma 4.38. Undecidability ensues.

4.40. Theorem. *For any* **FO***-fragment $\mathcal{L}$ containing $\mathcal{ALCFIO}$, (non-)projective restricted weak $(\mathcal{ALC}, \mathcal{L})$-separability is undecidable.*

Chapter **5**

Restricted strong separability

In this section we focus on the case of strong separability, still assuming the signature of a separating concept to be part of the input. We mainly look at **FO**, $\mathbf{DL}_{\mathcal{IO}}$, **GF** and $\mathbf{FO}^2$.

As in the full signature case and by virtue of the same arguments (Prop. 1.60), projectivity does not affect strong separability. We first observe that restricted strong separability can be seen as an interpolation existence problem between two formulas encoding the knowledge base. It can then also be seen as an entailment problem if the separation language has the Craig Interpolation Property. If not, we can use recent results on the complexity of interpolant existence [AJMOW21, JW21]. Listed below are combined complexity results in the case where the ontology language and the separation language coincide. The results we obtain in the following section are contrasted with the ones from the strong, full signature case.

	FULL	RESTRICTED
$\mathcal{ALC}$	EXP	2EXP
$\mathcal{ALCI}$	EXP	2EXP
$\mathcal{ALCO}$	?	2EXP
GF	2EXP	3EXP
$\mathbf{FO}^2$	NEXP	[2EXP, coN2EXP]
GNF	2EXP	2EXP

§ 5.1. Interpolation

We show that restricted strong separability can be reduced to a problem of interpolation existence.

5.1. *Definition.* Given **FO**-formulas $\varphi(x), \psi(x)$ and a fragment $\mathcal{L}$ of **FO**, we say that an $\mathcal{L}$-formula $\chi(x)$ is an *$\mathcal{L}$-interpolant of* φ, ψ if $\varphi(x) \models \chi(x)$, $\chi(x) \models \psi(x)$, and $\mathsf{sig}(\chi) \subseteq \mathsf{sig}(\varphi) \cap \mathsf{sig}(\psi)$. We say that $\mathcal{L}$ *has the CIP (Craig Interpolation Property)* if for all $\mathcal{L}$-formulas $\varphi(x), \psi(x)$ such that $\varphi(x) \models \psi(x)$, there exists an $\mathcal{L}$-interpolant of φ, ψ.

As in the strong full signature case, we work without loss of generality with labeled knowledge bases containing only one positive example a and one negative example b (Rem. 1.62). We show that a separating formula can then be seen as an interpolant between two formulas $\varphi_{\mathcal{K},\Sigma,a}$ and $\neg\varphi_{\mathcal{K},\Sigma,b}$, where $\varphi_{\mathcal{K},\Sigma,a}$ and $\varphi_{\mathcal{K},\Sigma,b}$ encode the knowledge base; one from the positive example's "point of view" and the other from the negative example's.

5.2. *Definition.* Let $(\mathcal{K}, \{a\}, \{b\})$ be a labeled **FO**-knowledge base and $\Sigma \subseteq \mathsf{sig}(\mathcal{K})$. Obtain $\mathcal{K}_{\Sigma,a}$ from $\mathcal{K}$ by

1. replacing all $X \in (\mathsf{rel}_1 \cup \mathsf{rel}_2) \setminus \Sigma$ by fresh symbols X_a,
2. replacing all constants $c \notin \Sigma \cup \{a\}$ by fresh variables x_c,
3. replacing a by x for a single fresh variable x,
4. adding $x = a$ if $a \in \Sigma$.

Let $\varphi_{\mathcal{K},\Sigma,a}(x) = \exists \mathbf{z}(\bigwedge \mathcal{K}_{\Sigma,a})$, where $\mathbf{z}$ 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 $\mathcal{K}_{\Sigma,a}$. $\mathcal{K}_{\Sigma,b}(x)$ and $\varphi_{\mathcal{K},\Sigma,b}(x)$ are defined in the same way, with a replaced by b.

5.3. *Example.* To illustrate Proposition 5.4, let $\Sigma = \{R\}$ and $\mathcal{K} = (\mathcal{O}, \mathcal{D})$, with $\mathcal{O} = \{A \sqsubseteq \forall R.\neg A\}$ and $\mathcal{D} = \{A(a), R(b,b)\}$. Then, $\neg R(x,x)$ strongly Σ-separates $(\mathcal{K}, \{a\}, \{b\})$ and is an interpolant for $\varphi_{\mathcal{K},\Sigma,a}, \neg\varphi_{\mathcal{K},\Sigma,b}$ where

$$\begin{aligned}
\varphi_{\mathcal{K},\Sigma,a}(x) &= \exists x_b\, R(x_b, x_b) \wedge A_a(x) \wedge \forall yz(R(y,z) \wedge A_a(y) \rightarrow \neg A_a(z)) \\
\varphi_{\mathcal{K},\Sigma,b}(x) &= \underbrace{\exists y_a\, R(x,x) \wedge A_b(y_a)}_{\mathcal{D}} \wedge \underbrace{\forall yz\,(R(y,z) \wedge A_b(y) \rightarrow \neg A_b(z))}_{\mathcal{O}}.
\end{aligned}$$

The reduction is then straightforward from the definition.

5.4. Proposition. *Let $(\mathcal{K}, \{a\}, \{b\})$ be a labeled* **FO***-knowledge base, $\Sigma \subseteq \mathsf{sig}(\mathcal{K})$ a signature, and $\mathcal{L}_S$ a fragment of* **FO***. Then, for any formula $\varphi(x)$ in $\mathcal{L}_S$, φ strongly Σ-separates $(\mathcal{K}, \{a\}, \{b\})$ iff φ is an $\mathcal{L}_S$-interpolant for $\varphi_{\mathcal{K},\Sigma,a}(x), \neg\varphi_{\mathcal{K},\Sigma,b}(x)$.*

Proof. Suppose φ strongly separates and let us show that it is an $\mathcal{L}_S$-interpolant. First, $\mathsf{sig}(\varphi) \subseteq \Sigma \subseteq \mathsf{sig}(\varphi_{\mathcal{K},\Sigma,a}) \cap \mathsf{sig}(\varphi_{\mathcal{K},\Sigma,b})$ as φ can be assumed w.l.o.g to be non-projectively separating. To show $\varphi_{\mathcal{K},\Sigma,a}(x) \models \varphi(x)$, suppose $\mathfrak{A} \models \varphi_{\mathcal{K},\Sigma,a}(d)$ for some $d \in \mathsf{dom}(\mathfrak{A})$. Then, $\mathfrak{A}' \models \mathcal{K}$ where $\mathfrak{A}'$ is defined as $\mathfrak{A}$ except for $a^{\mathfrak{A}'} = d$ and $c^{\mathfrak{A}'} = v(x_c)$ where v is an assignment witnessing $\mathfrak{A} \models \varphi_{\mathcal{K},\Sigma,a}(d)$. Then $\mathfrak{A}' \models \varphi(a^{\mathfrak{A}'})$ as $\mathcal{K} \models \varphi(a)$, so $\mathfrak{A} \models \varphi(d)$. The same argument shows that if $\mathfrak{A} \models \varphi_{\mathcal{K},\Sigma,b}(d)$ for some pointed model $(\mathfrak{A}, d)$, then $\mathfrak{A} \models \neg\varphi(d)$ using the fact that $\mathcal{K} \models \neg\varphi(b)$, hence $\varphi(x) \models \neg\varphi_{\mathcal{K},\Sigma,b}(x)$. Conversely, suppose φ is an interpolant. Then, for any $\mathfrak{A} \models \mathcal{K}$ we have $\mathfrak{A} \models \varphi_{\mathcal{K},\Sigma,a}(a^{\mathfrak{A}})$, thus $\mathfrak{A} \models \varphi(a^{\mathfrak{A}})$. Similarly, $\mathfrak{A} \models \neg\varphi(b^{\mathfrak{A}})$. ⊣

As **FO** has the CIP [Cr57], the existence of an **FO**-interpolant (thus of a separating **FO**-formula) can be reduced to an entailment problem between the two encoding formulas.

5.5. Theorem. *Let $(\mathcal{K}, \{a\}, \{b\})$ a labeled* **FO***-knowledge base and $\Sigma \subseteq \mathsf{sig}(\mathcal{K})$. Then $(\mathcal{K}, \{a\}, \{b\})$ is strongly* **FO**(Σ)*-separable iff $\varphi_{\mathcal{K},\Sigma,a}(x) \models \neg\varphi_{\mathcal{K},\Sigma,b}(x)$.*

If $\mathcal{O}$ is in **GNF**, then $\varphi_{\mathcal{K},\Sigma,a}$ and $\varphi_{\mathcal{K},\Sigma,a}$ are in **GNF**. In that context, if constants are excluded from Σ, we can reduce the existence of a **GNF** separating formula (and thus of an **FO** one) to **GNF**-satisfiability by Theorem 5.5, as **GNF** without constants has the CIP [BBC13]. We thus get a 2EXP upper bound that matches the lower bound, which also follows from satisfiability (Rem. 1.63).

5.6. Corollary. *If constants are excluded, $\mathsf{sep}_{\Sigma,s}$(**GNF**, **GNF**) $=$ $\mathsf{sep}_{\Sigma,s}$(**GNF**, **FO**) for all $\Sigma \subseteq \mathsf{sig}(\mathcal{K})$, and the associated decision problem is* 2EXP*-complete in combined complexity.*

If $\mathcal{O}$ is expressed in $\mathcal{L} \in \mathbf{DL}_{\mathcal{IO}}$, the encodings $\varphi_{\mathcal{K},\Sigma,a}$ and $\varphi_{\mathcal{K},\Sigma,b}$ are not guaranteed to be in $\mathcal{L}$. Then, the above corollary does not immediately apply. We next show it is possible to work around this problem by adjusting the encodings.

§ 5.2. $\mathbf{DL}_{\mathcal{IO}}$ ONTOLOGIES

We now restrict the ontology language to $\mathbf{DL}_{\mathcal{IO}}$. We first show that, in this context, Theorem 5.5 provides an EXP upper bound on $(\mathcal{L}, \mathbf{FO})$-separability as we can translate $\varphi_{\mathcal{K},\Sigma,a}(x) \models \neg\varphi_{\mathcal{K},\Sigma,b}(x)$ into a satisfiability condition in $\mathcal{ALCIO}^u$, decidable in exponential time. It matches the EXP lower bound given by satisfiability of $\mathcal{ALC}$-knowledge bases (Rem. 1.63).

5.7. Corollary. *For any $\mathcal{L} \in \mathbf{DL}_{\mathcal{IO}}$, restricted strong $(\mathcal{L}, \mathbf{FO})$ is* EXP*-complete.*

Proof. First observe that $\varphi_{\mathcal{K},\Sigma,a}(x) \models \neg\varphi_{\mathcal{K},\Sigma,b}(x)$ iff $\varphi_{\mathcal{K},\Sigma,a}(x) \wedge \varphi_{\mathcal{K},\Sigma,b}(x)$ is not satisfiable. Then, obtain $C_{\mathcal{K},\Sigma,a}$ from $\mathcal{K}$ by taking the conjunction of all $\forall u.(C \to D)$ for any $C \sqsubseteq D \in \mathcal{O}$, all $\forall u.(\{c\} \to \exists R.\{d\})$ for any $R(c,d) \in \mathcal{D}$, all $\forall u.(\{c\} \to A)$ for any $A(c) \in \mathcal{D}$ and then replacing all concept and role names $X \notin \Sigma$ by fresh and distinct symbols X_a, all constants $c \notin \Sigma \cup \{a\}$ by fresh and distinct constants c_a, the constant a by a fresh constant m if $a \in \Sigma$, and adding $\{m\} \leftrightarrow \{a\}$ 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 the $\mathcal{ALCIO}^u$-concept $\{m\} \wedge C_{\mathcal{K},\Sigma,a} \wedge C_{\mathcal{K},\Sigma,b}$ is satisfiable. ⊣

5.2.1. *Boolean Hybrid CQs*

It follows from Theorem 5.5 that one can use **FO** theorem provers such as *Vampire* [HHKV12] to compute strongly separating formulas. **FO** is arguably too powerful, however, to serve as a useful separation language for labeled description logic knowledge bases. It is then natural to look for a fragment of **FO** that is needed to obtain a strongly separating formula in case that there is a strongly separating formula in **FO**. By taking the closure $\mathbf{BoCQ}^{\mathcal{ALCIO}}$ of $\mathbf{CQ}^{\mathcal{ALCIO}}$ under Boolean connectors one obtains a sufficiently powerful language, at least if the knowledge base does not admit nominals. As $\mathbf{BoCQ}^{\mathcal{ALCIO}}$ is included in **GNF**, it is decidable. Note that **FO** is considered here with constants. This subsection is dedicated to proving the following theorem.

5.8. Theorem. *Let $(\mathcal{K}, \{a\}, \{b\})$ be a labeled $\mathcal{ALCI}$-knowledge base and $\Sigma \subseteq \mathrm{sig}(\mathcal{K})$. Then $(\mathcal{K}, \{a\}, \{b\})$ is strongly* $\mathbf{FO}(\Sigma)$*-separable iff it is strongly* $\mathbf{BoCQ}^{\mathcal{ALCIO}}(\Sigma)$*-separable.*

We use the next Lemma to prove Theorem 5.8.

5.9. Lemma. *Let $(\mathcal{K}, \{a\}, \{b\})$ be a labeled $\mathcal{ALCI}$-knowledge base and $\Sigma \subseteq \mathrm{sig}(\mathcal{K})$. If $(\mathcal{K}, \{a\}, \{b\})$ is not* $\mathbf{BoCQ}^{\mathcal{ALCIO}}(\Sigma)$*-separable, then there exist pointed models $(\mathfrak{A}, d)$ 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 $(\mathfrak{A}, d) \Leftrightarrow^{mod}_{\mathbf{CQ}^{\mathcal{ALCIO}},\Sigma} (\mathfrak{B}, e)$.*

Proof. Suppose $(\mathcal{K}, \{a\}, \{b\})$ is not $\mathbf{BoCQ}^{\mathcal{ALCIO}}(\Sigma)$-separable. Let Γ be the set of all $\psi(x)$ in $\mathbf{BoCQ}^{\mathcal{ALCIO}}(\Sigma)$ such that $\varphi_{\mathcal{K},\Sigma,a}(x) \models \psi(x)$. Then $\Gamma \cup \{\varphi_{\mathcal{K},\Sigma,b}(x)\}$ is satisfiable: if not, then by compactness there exists a conjunction φ of formulas in Γ such that $\varphi \models \neg\varphi_{\mathcal{K},\Sigma,b}$. But then φ is a separating $\mathbf{BoCQ}^{\mathcal{ALCIO}}(\Sigma)$-formula, as $\varphi \in \Gamma$ implies $\mathcal{K} \models \varphi(a)$ and $\varphi \models \neg\varphi_{\mathcal{K},\Sigma,b}$ implies $\mathcal{K} \models \neg\varphi(b)$. Let $(\mathfrak{B}, e)$ witness it. Let Ψ be the set of all $\psi(x)$ in $\mathbf{BoCQ}^{\mathcal{ALCIO}}(\Sigma)$ such that $\mathfrak{B} \models \psi(e)$. By compactness and assumption we can assume $(\mathfrak{B}, e)$ is wlog such that $\Psi \cup$

$\{\varphi_{\mathcal{K},\Sigma,a}(x)\}$ is satisfiable by some $(\mathfrak{A},d)$ (that is, that there exists at least one such $(\mathfrak{B},e)$). Otherwise, $\Gamma \cup \{\varphi_{\mathcal{K},\Sigma,b}(x)\} \models \varphi_{\mathcal{K},\Sigma,a}(x)$, i.e. $\varphi_{\mathcal{K},\Sigma,b}(x) \models \varphi_{\mathcal{K},\Sigma,a}(x)$, contradicting inseparability. By definition, $(\mathfrak{A},d)$ and $(\mathfrak{B},e)$ are such that $\mathfrak{A} \models \varphi_{\mathcal{K},\Sigma,a}(d)$ and $\mathfrak{B} \models \varphi_{\mathcal{K},\Sigma,b}(e)$ and $(\mathfrak{A},d) \Leftrightarrow_{\mathbf{CQ}^{\mathcal{ALCIO}},\Sigma} (\mathfrak{B},e)$. Finally, we conclude by assuming without loss of generality (modulo taking elementary extensions) that $\mathfrak{A}$ and $\mathfrak{B}$ are ω-saturated (cf. Theorem 1.7). ⊣

Proof of Theorem 5.8. The "if" direction is trivial. For the converse direction, assume that the left condition holds. By assumption and Theorem 5.5, $\varphi_{\mathcal{K},\Sigma,a}(x) \models \neg\varphi_{\mathcal{K},\Sigma,b}(x)$. Assume there does not exist a separating formula in $\mathbf{BoCQ}^{\mathcal{ALCIO}}(\Sigma)$. By Lemma 5.9, there exist pointed models $(\mathfrak{A},d)$ 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 $(\mathfrak{A},d) \Leftrightarrow^{\mathrm{mod}}_{\mathbf{CQ}^{\mathcal{ALCIO}},\Sigma} (\mathfrak{B},e)$. 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 Σ-homomorphisms

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

It also immediately follows from $(\mathfrak{A},d) \Leftrightarrow^{\mathrm{mod}}_{\mathbf{CQ}^{\mathcal{ALCIO}},\Sigma} (\mathfrak{B},e)$ that for any $c \in \mathrm{dom}(\mathfrak{A})$ there exists $c' \in \mathrm{dom}(\mathfrak{B})$ such that $(\mathfrak{A},c) \sim_{\mathcal{ALCIO},\Sigma} (\mathfrak{B},c')$, and vice versa. We use this fact to merge $\mathfrak{A}$ and $\mathfrak{B}$ to a single model $\mathfrak{C}$ and show in Lemma 5.11 that it witnesses inseparability. ⊣

5.10. *Definition.* We define the *bisimulation product* $\mathfrak{C}$ of $\mathfrak{A}$ and $\mathfrak{B}$ as follows. $\mathrm{dom}(\mathfrak{C}) = \{(c,c') \in \mathrm{dom}(\mathfrak{A}) \times \mathrm{dom}(\mathfrak{B}) \mid (\mathfrak{A},c) \sim_{\mathcal{ALCI},\Sigma} (\mathfrak{B},c')\}$

$$
\begin{array}{lr}
(c,c') \in A^{\mathfrak{C}} \text{ if } c \in A^{\mathfrak{A}} \text{ (equiv. if } c' \in A^{\mathfrak{B}}) & \text{for all } A \in \Sigma \\
(c,c') \in A^{\mathfrak{C}} \text{ if } c \in A^{\mathfrak{A}} & \text{for all } A \in \mathrm{sig}(\varphi_{\mathcal{K},\Sigma,a}) \setminus \Sigma \\
(c,c') \in A^{\mathfrak{C}} \text{ if } c' \in A^{\mathfrak{B}} & \text{for all } A \in \mathrm{sig}(\varphi_{\mathcal{K},\Sigma,b}) \setminus \Sigma \\
((c_1,c_1'),(c_2,c_2')) \in R^{\mathfrak{C}} \text{ if } (c_1,c_2) \in R^{\mathfrak{A}} \text{ and } (c_1',c_2') \in R^{\mathfrak{B}} & \text{for all } R \in \Sigma \\
((c_1,c_1'),(c_2,c_2')) \in R^{\mathfrak{C}} \text{ if } (c_1,c_2) \in R^{\mathfrak{A}} & \text{for all } R \in \mathrm{sig}(\varphi_{\mathcal{K},\Sigma,a}) \setminus \Sigma \\
((c_1,c_1'),(c_2,c_2')) \in R^{\mathfrak{C}} \text{ if } (c_1',c_2') \in R^{\mathfrak{B}} & \text{for all } R \in \mathrm{sig}(\varphi_{\mathcal{K},\Sigma,b}) \setminus \Sigma \\
c^{\mathfrak{C}} = (c^{\mathfrak{A}}, c^{\mathfrak{B}}) & \text{for all } c \in \Sigma
\end{array}
$$

5.11. Lemma. $\mathfrak{C} \models (\varphi_{\mathcal{K},\Sigma,a} \wedge \varphi_{\mathcal{K},\Sigma,b})(d,e)$*, i.e. the pointed model* $(\mathfrak{C},(d,e))$ *witnesses inseparability of* $(\mathcal{K},\{a\},\{b\})$.

Proof. We can view $\mathcal{K}_{\Sigma,a}$ as the union of $\mathcal{O}_{\Sigma,a}$ and $\mathcal{D}_{\Sigma,a}$, where $\mathcal{O}_{\Sigma,a}$ is a copy of $\mathcal{O}$ in which all concept and role names $X \notin \Sigma$ have been replaced by fresh symbols X_a, and $\mathcal{D}_{\Sigma,a}$ 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 $a \in \Sigma$. $\mathcal{K}_{\Sigma,b}$ can be viewed accordingly with a replaced by b. By taking the conjunction of all members of $\mathcal{D}_{\Sigma,a}$ and existentially quantifying over all variables distinct from x we obtain a formula in $\mathbf{CQ}^{\mathcal{ALCIO}}$. We first show that $\mathfrak{C} \models \mathcal{O}_{\Sigma,a} \cup \mathcal{O}_{\Sigma,b}$, for which it suffices to show that (1) the projection $p_a : \mathfrak{C} \to \mathfrak{A}$ defined by $p_a(c,c') = c$ is an $\mathcal{ALCIO}(\mathsf{sig}(\varphi_{\mathcal{K},\Sigma,a}))$-bisimulation between $\mathfrak{C}$ and $\mathfrak{A}$ and (2) the projection $p_b : \mathfrak{C} \to \mathfrak{B}$ defined by $p_b(c,c') = c'$ is an $\mathcal{ALCIO}(\mathsf{sig}(\varphi_{\mathcal{K},\Sigma,b}))$-bisimulation between $\mathfrak{C}$ and $\mathfrak{B}$. The proof of (1) and (2) is straightforward and omitted. It follows from (1) and (2), $\mathfrak{A} \models \mathcal{O}_{\Sigma,a}$ and $\mathfrak{B} \models \mathcal{O}_{\Sigma,b}$ that $\mathfrak{C} \models \mathcal{O}_{\Sigma,a} \cup \mathcal{O}_{\Sigma,b}$. Next we lift the variable assignments v_a and v_b from $\mathfrak{A}$ (resp. $\mathfrak{B}$) to $\mathfrak{C}$:

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

It is then clear that $\mathfrak{C} \models_{\bar{v}_a} \mathcal{D}_{\Sigma,a}(d,e)$ and $\mathfrak{C} \models_{\bar{v}_b} \mathcal{D}_{\Sigma,b}(d,e)$, which, with $\mathfrak{C} \models \mathcal{O}_{\Sigma,a} \cup \mathcal{O}_{\Sigma,b}$, implies $\mathfrak{C} \models (\varphi_{\mathcal{K},\Sigma,a} \wedge \varphi_{\mathcal{K},\Sigma,b})(d,e)$. ⊣

5.2.2. *2Exp-completeness for* $\mathbf{DL}_{\mathcal{IO}}$

We now show that for all $\mathcal{L} \in \mathbf{DL}_{\mathcal{IO}}$, restricted strong $(\mathcal{L},\mathcal{L})$-separability is 2EXP-complete in combined complexity. The foundation for the proofs of both bounds is given by results from [AJMOW21] on the complexity of interpolant existence in description logics. As mentioned above, one may not immediately apply those complexity results in combination with Theorem 5.8, as $\varphi_{\mathcal{K},\Sigma,a}$ and $\varphi_{\mathcal{K},\Sigma,b}$ are not necessarily in $\mathcal{L}$. We thus introduce a different notion of interpolant. We first prove the upper bounds.

5.12. *Definition.* Let $\mathcal{L} \in \mathbf{DL}_{\mathcal{IO}}$. Let $\mathcal{O}_1, \mathcal{O}_2$ be $\mathcal{L}$-ontologies, C_1, C_2 be $\mathcal{L}$-concepts, and $\Sigma = \mathsf{sig}(\mathcal{O}_1, C_1) \cap \mathsf{sig}(\mathcal{O}_2, C_2)$. 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 $\mathcal{O}_1 \models C_1 \sqsubseteq C$ and $\mathcal{O}_2 \models C \sqsubseteq C_2$.

5.13. Theorem ([AJMOW21]). *Let $\mathcal{L} \in \{\mathcal{ALCO}, \mathcal{ALCIO}\}$. Then deciding whether an $\mathcal{L}$-interpolant exists for $\mathcal{L}$-tuples $(\mathcal{O}_1, \mathcal{O}_2, C_1, C_2)$ is* 2EXP*-complete.*

It now suffices to introduce the following notation.

5.14. *Definition.* Let $\mathcal{L} \in \mathbf{DL}_{\mathcal{IO}}$, $(\mathcal{K}, \{a\}, \{b\})$ be an $\mathcal{L}$-knowledge base with $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ and $\Sigma \subseteq \mathsf{sig}(\mathcal{K})$. Let $\mathcal{LO}$ denote the extension of $\mathcal{L}$ by nominals (if $\mathcal{L}$ contains nominals already then set $\mathcal{LO} = \mathcal{L}$). We define an $\mathcal{LO}$-ontology $\mathcal{O}_{\mathcal{K},\Sigma,a}$ from $\mathcal{K}$, that consists of the ontology

$$\begin{aligned} &\mathcal{O} \cup \{\{c\} \sqsubseteq \exists R.\{d\} \mid R(c,d) \in \mathcal{D}\} \\ &\quad \cup \{\{c\} \sqsubseteq A \mid A(c) \in \mathcal{D}\} \end{aligned}$$

from which are replaced all concept or role names $X \notin \Sigma$ by a fresh symbol X_a, all constants $c \notin \Sigma \cup \{a\}$ by fresh and distinct constants c_a, and the constant a by a fresh constant m_a. If $a \in \Sigma$ then $\{m_a\} \equiv \{a\}$ is added. $\mathcal{O}_{\mathcal{K},\Sigma,b}$ is obtained from $\mathcal{K}$ in the same way by replacing a by b.

5.15. Theorem. *For all $\mathcal{L} \in \mathbf{DL}_{\mathcal{IO}}$, restricted strong $(\mathcal{L}, \mathcal{L})$-separability is in* 2Exp.

Proof. 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 (*i.e.* $\mathcal{LO} = \mathcal{L}$), then the upper bounds follow immediately from Theorem 5.13. If $\mathcal{L}$ does not contain nominals, then we may assume that Σ does not contain constants. Then $\mathcal{O}_{\mathcal{K},\Sigma,a}$ and $\mathcal{O}_{\mathcal{K},\Sigma,b}$ do not share any constants 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 $\mathcal{O}_{\mathcal{K},\Sigma,a}$, $\mathcal{O}_{\mathcal{K},\Sigma,b}$, $m_a, \neg m_b$; the upper bound follows again. ⊣

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

5.16. Lemma. *Let $\mathcal{L} \in \mathbf{DL}_{\mathcal{IO}}$, $(\mathcal{K}, \{a\}, \{b\})$ be a labeled $\mathcal{L}$-knowledge base and $\Sigma \subseteq \mathsf{sig}(\mathcal{K})$. Then $(\mathcal{K}, \{a\}, \{b\})$ is strongly $\mathcal{L}(\Sigma)$-separable iff $(\mathfrak{A}, a^{\mathfrak{A}}) \not\sim_{\mathcal{L},\Sigma} (\mathfrak{B}, b^{\mathfrak{B}})$ for all $\mathfrak{A}, \mathfrak{B} \models \mathcal{K}$.*

Proof. The left to right implication is immediate as bisimulation implies logical equivalence. For the converse, assume $(\mathcal{K}, \{a\}, \{b\})$ is not strongly $\mathcal{L}(\Sigma)$-separable. Let

$$\begin{aligned} \Gamma_a &= \{C \in \mathcal{L}(\Sigma) \mid \mathcal{K} \models C(a)\} \\ \Gamma_b &= \{C \in \mathcal{L}(\Sigma) \mid \mathcal{K} \models C(b)\} \end{aligned}$$

For any set of concepts Γ we say that Γ is satisfiable in a w.r.t. $\mathcal{K}$ if $(\mathcal{O}, \mathcal{D} \cup \{C(a) \mid C \in \Gamma\})$ is satisfiable. We show that $\Gamma_a \cup \Gamma_b$ is satisfiable in a and b w.r.t. $\mathcal{K}$. We only show it for a as the proof is dual for b. Assume it is not satisfiable. Then $(\mathcal{O}, \mathcal{D} \cup \{C(a) \mid C \in \Gamma_b\})$ is not satisfiable. By compactness there exists $D \in \Gamma_b$ such that $\mathcal{K} \models \neg D(a)$. But $\mathcal{K} \models D(b)$ by definition, so $\neg D$ is a separating concept, which is the desired contradiction. Now let $\Gamma_0 = \Gamma_a \cup \Gamma_b$ and let $C_1, C_2, \ldots$ be an enumeration of all $\mathcal{L}(\Sigma)$-concepts such that $C_i \notin \Gamma_0$. For all $i \geq 0$ let

$$\Gamma_{i+1} = \begin{cases} \Gamma_i \cup \{C_{i+1}\} & \text{if } \Gamma_i \cup \{C_{i+1}\} \text{ is satisfiable in } a \text{ w.r.t. } \mathcal{K}, \\ \Gamma_i \cup \{\neg C_{i+1}\} & \text{otherwise.} \end{cases}$$

We now show that, for all $i \geq 1$, either $\Gamma_i \cup \{C_{i+1}\}$ is satisfiable in a and b w.r.t. $\mathcal{K}$ or $\Gamma_i \cup \{\neg C_{i+1}\}$ is satisfiable in a and b w.r.t. $\mathcal{K}$. Assume that is proved for $i-1$. Assume w.l.o.g. that $\Gamma_i = \Gamma_0 \cup \{C_1, \ldots, C_i\}$. Assume w.l.o.g. for contradiction that $\Gamma_i \cup \{C_{i+1}\}$ is not satisfiable in a w.r.t. $\mathcal{K}$ and that $\Gamma_i \cup \{\neg C_{i+1}\}$ is not satisfiable in b w.r.t. $\mathcal{K}$. By compactness, there exists $D \in \Gamma_b$ such that $\mathcal{K} \models D'(a)$, where $D' = (D \sqcap C_1 \sqcap \cdots \sqcap C_i) \rightarrow \neg C_{i+1}$. Then $D' \in \Gamma_a$ by definition. Then $D' \in \Gamma_i$, so Γ_i is not satisfiable in b w.r.t $\mathcal{K}$, which is the desired contradiction. Let $\Gamma_\infty = \bigcup_{i>0} \Gamma_i$. Then there exists $\mathfrak{A}, \mathfrak{B} \models \mathcal{K}$ such that $\mathcal{A} \models C(a^{\mathfrak{A}})$ and $\mathfrak{B} \models C(b^{\mathfrak{B}})$ for all $C \in \Gamma_\infty$. Thus, $(\mathfrak{A}, a^{\mathfrak{A}}) \equiv_{\mathcal{L},\Sigma} (\mathfrak{B}, b^{\mathfrak{B}})$. We can assume $\mathfrak{A}$ and $\mathfrak{B}$ are ω-saturated by Theorem 1.7 and the fact that elementary extensions preserve types. Then, $(\mathfrak{A}, a^{\mathfrak{A}}) \sim_{\mathcal{L},\Sigma} (\mathfrak{B}, b^{\mathfrak{B}})$ by Lemma 1.25. ⊣

As part of the lower bound proof for interpolant existence in [AJMOW21], it is proved that for an $\mathcal{L}$-ontology $\mathcal{O}$ and database $\mathcal{D}$ of the form $\{R(a,a)\}$ it is 2Exp-hard to decide whether 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 $d \neq a^{\mathfrak{B}}$. 2Exp-hardness of separability immediately follows.

5.17. Theorem. *For all $\mathcal{L} \in \mathbf{DL}_{\mathcal{IO}}$, restricted strong $(\mathcal{L}, \mathcal{L})$-separability is* 2Exp-*hard in combined complexity.*

Proof. Suppose such $\mathcal{L}$, $\mathcal{K} = (\mathcal{O}, \mathcal{D})$, Σ, and a are given. Let b be a fresh constant 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 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 $d \neq a^{\mathfrak{B}}$ iff there exist models $\mathfrak{A}$ and $\mathfrak{B}$ of $(\mathcal{O}', \mathcal{D}')$ such that $(\mathfrak{A}, a^{\mathfrak{A}}) \sim_{\mathcal{L},\Sigma} (\mathfrak{B}, b^{\mathfrak{B}})$. ⊣

§ 5.3. GF AND FO^2 ONTOLOGIES

We now look at restricted strong **FO** and **GF** (resp. **FO**^{2})-separability of **GF** (resp. **FO**^{2})-labeled knowledge bases. Assume a labeled $\mathcal{L} \in \{\mathbf{GF}, \mathbf{FO}^2\}$-knowledge base $(\mathcal{K}, \{a\}, \{b\})$ with $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ is given. Let Σ be a signature. Then $\varphi_{\mathcal{K},\Sigma,a}$ and $\varphi_{\mathcal{K},\Sigma,b}$ defined in 5.2 are also in $\mathcal{L}$. Upper bounds for $(\mathbf{GF}, \mathbf{FO})$ and $(\mathbf{FO}^2, \mathbf{FO})$-separability then follow, by CIP of **FO**, from the complexity of their respective satisfiability problems. Matching lower bounds are also guaranteed by satisfiability, as usual (Rem. 1.63).

5.18. Theorem.

- *Restricted strong* $(\mathbf{GF}, \mathbf{FO})$*-separability is* 2EXP*-complete.*
- *Restricted strong* $(\mathbf{FO}^2, \mathbf{FO})$*-separability is* CONEXP*-complete.*

Now, **GF** [HM02] and $\mathbf{FO}^2$ [Co69] do not enjoy the CIP so we cannot replicate Corollary 5.6 to get complexity bounds for $(\mathbf{GF}, \mathbf{GF})$ and $(\mathbf{FO}^2, \mathbf{FO}^2)$-separability. Instead, we use recent results by [JW21], namely that interpolant existence in **GF** is decidable in 3EXP, and interpolant existence in $\mathbf{FO}^2$ is decidable in CON2EXP. The lower bounds follow from a simple reduction from interpolation to separability. Then, Theorems 5.20 and 5.21 focus on the upper bounds.

5.19. *Reduction from interpolation to separability.* Let $\varphi(x), \psi(x)$ be **GF**-formulas (resp. $\mathbf{FO}^2$), $\Sigma = \mathsf{sig}(\varphi) \cap \mathsf{sig}(\psi)$, $\mathcal{D} = \{A(a), B(b)\}$ with $A, B \notin \Sigma$ and $\mathcal{O} = \{\forall x A(x) \to \varphi(x), \forall x B(x) \to \neg\psi(x)\}$. It is then straightforward that if $\varphi(x)$ and $\psi(x)$ admit an interpolant, then $((\mathcal{O}, \mathcal{D}), \{a\}, \{b\})$ is strongly $\mathbf{GF}(\Sigma)$-separable (resp. $\mathbf{FO}^2$) by that interpolant. Conversely, if some $\chi(x)$ Σ-separates, then it follows from $\mathcal{K} \models \chi(a)$ and $A, B \notin \mathsf{sig}(\chi)$ that $\varphi(x) \models \chi(x)$ and $\neg\psi(x) \models \neg\chi(x)$.

We move on to the upper bounds. As we aim to apply results on the complexity of interpolant existence that have been proved for **GF** and $\mathbf{FO}^2$ *without* constants, we cannot use constants in the construction of the encodings. Then, the previous encodings are not in **GF** (resp. $\mathbf{FO}^2$) and we devise new ones.

5.20. Theorem. *Restricted strong* $(\mathbf{GF}, \mathbf{GF})$*-separability is* 3EXP*-complete.*

Proof. Assume a labeled **GF**-knowledge base $(\mathcal{K}, \{a\}, \{b\})$ with $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ is given and Σ is a relational signature. Consider the formulas $\varphi_{\mathcal{K},\Sigma,a}$ and $\varphi_{\mathcal{K},\Sigma,b}$ from Definition 5.2. To obtain **GF**-formulas $\varphi^{\mathbf{GF}}_{\mathcal{K},\Sigma,a}$ and $\varphi^{\mathbf{GF}}_{\mathcal{K},\Sigma,b}$, take fresh relation symbols $R_{\mathcal{D},a}, R_{\mathcal{D},b} \in \mathsf{rel}_{|\mathsf{cons}(\mathcal{D})|}$. Add $R_{\mathcal{D},a}(\mathbf{y})$ to $\mathcal{K}_{\Sigma,a}$ when constructing $\varphi_{\mathcal{K},\Sigma,a}(x)$, where $\mathbf{y}$ is an enumeration of the variables in $\mathcal{K}_{\Sigma,a}$. Do the same to construct $\varphi^{\mathbf{GF}}_{\mathcal{K},\Sigma,b}(x)$, using $R_{\mathcal{D},b}$ instead of $R_{\mathcal{D},a}$. The formulas $\varphi^{\mathbf{GF}}_{\mathcal{K},\Sigma,a}$ and $\neg\varphi^{\mathbf{GF}}_{\mathcal{K},\Sigma,b}$ 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 φ, φ strongly Σ-separates $(\mathcal{K}, \{a\}, \{b\})$ iff φ is an interpolant for $\varphi^{\mathbf{GF}}_{\mathcal{K},\Sigma,a}(x), \neg\varphi^{\mathbf{GF}}_{\mathcal{K},\Sigma,b}(x)$. The complexity upper bound now follows from [JW21]. ⊣

5.21. Theorem. *Restricted strong* $(\mathbf{FO}^2, \mathbf{FO}^2)$*-separability is in* CON2EXP *and* 2EXP*-hard.*

Proof. For every $c \in \mathsf{cons}(\mathcal{D})$ let $A_c \in \mathsf{rel}_1 \setminus \mathsf{sig}(\mathcal{K})$. Let $\varphi^2_{\mathcal{K},\Sigma,a} = \bigwedge \mathcal{O} \wedge \mathcal{D}_{\mathsf{enc}}$, where $\mathcal{D}_{\mathsf{enc}}$ is the conjunction of the following formulas in which every $R \in$

$\mathsf{rel} \cap (\mathsf{sig}(\mathcal{K}) \setminus \Sigma)$ is replaced by a fresh symbol R_a.

$$
\begin{array}{ll}
\exists x A_c(x) \wedge \forall x \forall y (A_c(x) \wedge A_c(y) \rightarrow x = y) & \text{for all } c \in \mathsf{cons}(\mathcal{D}) \\
\exists x \exists y R(x,y) \wedge A_c(x) \wedge A_{c'}(y) & \text{for all } R(c,c') \in \mathcal{D} \\
\exists x A(x) \wedge A_c(x) & \text{for all } A(c) \in \mathcal{D} \\
A_a(x) &
\end{array}
$$

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 constants 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 **FO**^{2} and a formula φ strongly Σ-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 **FO**^{2} is decidable in N2EXP [JW21]. ⊣

Conclusion

In this thesis I have presented model-theoretic and computational perspectives on the separability of labeled data points by logical formulas in an ontology-mediated setting. This work tackles fundamental questions in the area of logical supervised learning. It is arguably the first heavy effort in that direction. Equivalence or undecidability results can be of immediate interest to applied research. Non-trivial observations on the links between separability and well-known decision problems offer much potential to be exploited beyond what is achieved here. Numerous other languages remain to be discussed, both as ontology or separation languages. Below are other possible directions for future work.

Unique Name Assumption. The UNA is not made anywhere except in the full weak projective case for description logics admitting number restrictions. While it has no effect on some results, as mentioned in the introduction to Chapter 2, other results fail under the UNA. Notably, the characterisation of full weak (**FO**, **FO**)-separability given in Theorem 2.1 fails in Example 1.54 if b and a are swapped: $(\mathcal{K}, \{b\}, \{a\})$ is then weakly separated by $\exists y \exists z (y \neq z) \wedge \mathsf{citizen\_of}(x, y) \wedge \mathsf{born\_in}(x, z)$. In that same example, if $\mathcal{O}$ is augmented with an axiom stating that born_in and citizen_of are functional, then, under UNA, $(\mathcal{K}_2, \{a\}, \{b\})$ is strongly separated by $\exists y \, \mathsf{born\_in}(x, y) \wedge \mathsf{citizen\_of}(x, y)$. Then, the characterisation of strong **FO**-separability given in Theorem 3.1 also fails under UNA. On the other hand, it remains open whether our results for $\mathcal{ALCQ}(\mathcal{I})$ hold *without* the UNA.

Constants. Another generalization can be made to the case where constants are admitted in the ontology or separation languages, which we do not consider here except for description logics containing nominals in Chapter 4. It is an arguably relevant approach to look for differences between positive and negative examples in terms of how they relate to specific other individuals. Clearly one would still need to exclude constants from $E^+ \cup E^-$, which would otherwise trivialize the problem.

Length restrictions. Separability can be investigated with formula length or quantifier depth as part of the input parameters, or fixed beforehand. Shorter concepts are preferable for better generalization and readability. [F19] looked at the decision problem of full weak $(\mathcal{EL}, \mathcal{EL})$-separability with concepts of fixed role depth k. In combined complexity, the problem is polynomial for $k = 0$ and NP-complete for $k \geqslant 1$.

Other possible directions include looking for instances of separability where at least data complexity is tractable –possibly by considering inexpressive languages– and imposing some restrictions on E^+, E^- such as assuming that they partition the space of examples (*definability* of E^+) or ask further that E^+ be a singleton (existence of a *referring expression* for it, cf. [TW19] for related work).

The following appendices summarize all non-trivial results involving separating power (appendix 1) and complexity (appendix 2).

Appendix 1: *Results on separating power*

I. *Full weak separability case*
For $\mathcal{L}_S, \mathcal{L}'_S \supseteq \mathbf{UCQ}$,

1. $\mathsf{sep}_w(\mathbf{FO}, \mathcal{L}_S) = \mathsf{sep}^p_w(\mathbf{FO}, \mathcal{L}_S) = \mathsf{sep}^p_w(\mathbf{FO}, \mathcal{L}'_S)$
2. $\mathsf{sep}^p_w(\mathbf{FO}^2, \mathbf{FO}^2) \neq \mathsf{sep}_w(\mathbf{FO}^2, \mathbf{FO})$
3. $\mathsf{sep}^p_w(\mathcal{ALCI}, \mathcal{ALCI}) = \mathsf{sep}^p_w(\mathcal{ALCI}, \mathcal{L}_S)$
4. $\mathsf{sep}^p_w(\mathbf{GF}, \mathbf{GF}) = \mathsf{sep}^p_w(\mathbf{GF}, \mathbf{oGF}) = \mathsf{sep}^p_w(\mathbf{GF}, \mathcal{L}_S)$
5. Whenever a labeled $\mathcal{ALCI}$-knowledge base is weakly projectively $\mathcal{ALCI}$ separable, then a single fresh concept name suffices for separation.
6. Whenever a labeled **GF**-knowledge base is weakly projectively **GF** separable, then a single fresh relation symbol suffices for separation.

II. *Full strong separability case*
For $\mathcal{L}_S \supseteq \mathbf{UCQ}$,

1. $\mathsf{sep}_s(\mathbf{FO}, \mathcal{L}_S) = \mathsf{sep}_s(\mathbf{FO}, \mathbf{FO})$
2. $\mathsf{sep}_s(\mathcal{ALC}, \mathcal{ALC}) = \mathsf{sep}_s(\mathcal{ALC}, \mathcal{L}_S)$
3. $\mathsf{sep}_s(\mathcal{ALCI}, \mathcal{ALCI}) = \mathsf{sep}_s(\mathcal{ALCI}, \mathcal{L}_S)$
4. $\mathsf{sep}_s(\mathbf{GF}, \mathbf{GF}) = \mathsf{sep}_s(\mathbf{GF}, \mathbf{oGF}) = \mathsf{sep}_s(\mathbf{GF}, \mathcal{L}_S)$
5. $\mathsf{sep}_s(\mathbf{FO}^2, \mathbf{FO}^2) = \mathsf{sep}_s(\mathbf{FO}^2, \mathcal{L}_S)$

III. *Restricted weak separability case*

1. Let $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}\}$. Restricted weak projective $(\mathcal{L}, \mathcal{L})$-separability is invariant under the addition of role names as helper symbols.
2. Whenever a labeled $\mathcal{ALCO}$-knowledge base is weakly projectively $\mathcal{ALCO}$ separable using $\mathsf{rel}_1 \cup \mathsf{rel}_2$ as helper symbols, then a single fresh role name suffices for separation.
3. $\mathsf{sep}^p_{w,\Sigma}(\mathcal{ALCI}, \mathcal{ALCI}) = \mathsf{sep}_{w,\Sigma}(\mathcal{ALCI}, \mathbf{UCQ}_r^{\mathcal{ALCI}})$ for all Σ
4. $\mathsf{sep}^p_{w,\Sigma}(\mathcal{ALC}, \mathcal{ALCO}) = \mathsf{sep}_{w,\Sigma}(\mathcal{ALC}, \mathbf{UCQ}_r^{\mathcal{ALCO}})$ for all Σ

IV. *Restricted strong separability case*

$\mathsf{sep}_{s,\Sigma}(\mathcal{ALCI}, \mathbf{FO}) = \mathsf{sep}_{s,\Sigma}(\mathcal{ALCI}, \mathbf{UCQ}_r^{\mathcal{ALCIO}})$ for all Σ

Appendix 2: *Results on complexity*

I. *Full weak separability case*
For $\mathcal{L}_S \supseteq$ **UCQ**,

1. (Non-)proj. $(\mathbf{GNF}, \mathcal{L}_S)$ 2Exp-complete
2. (Non-)proj. $(\mathbf{FO}^2, \mathbf{FO}^2)$ undecidable
3. (Non-)proj. $(\mathbf{FO}^2, \mathbf{FO})$ undecidable
4. Proj. $(\mathcal{ALC}, \mathcal{ALC})$ NExp-complete, PSpace-complete in data
5. (Non-)proj. $(\mathcal{ALCI}, \mathcal{L}_S)$ NExp-complete (combined & data)
6. Proj. $(\mathcal{ALCQ}, \mathcal{ALCQ})$ NExp-complete (with UNA)
7. Proj. $(\mathcal{ALCQI}, \mathcal{ALCQI})$ Exp-complete NExp-complete (with UNA)
8. (Non-)proj. $(\mathbf{GF}, \mathcal{L}_S)$ 2Exp-complete (combined & data)
9. $(\mathcal{ALC}, \mathcal{EL}(\mathcal{I}))$ undecidable

II. *Full strong separability case*
For $\mathcal{L}_S \supseteq$ **UCQ**,

1. $(\mathcal{ALC}, \mathcal{L}_S)$ Exp-complete, coNP-complete in data
2. $(\mathcal{ALCI}, \mathcal{L}_S)$ Exp-complete, coNP-complete in data
3. $(\mathbf{GF}, \mathcal{L}_S)$ 2Exp-complete, coNP-complete in data
4. $(\mathbf{FO}^2, \mathcal{L}_S)$ NExp-complete, coNP-complete in data

III. *Restricted weak separability case*

1. Proj. $(\mathcal{ALC}, \mathcal{ALC})$ 2Exp-complete
2. Proj. $(\mathcal{ALCI}, \mathcal{ALCI})$ 2Exp-complete
3. Proj. $(\mathcal{ALCO}, \mathcal{ALCO})$ 3Exp-complete ($\mathsf{rel}_1 \cup \mathsf{rel}_2$ as helper symbols)
4. Proj. $(\mathcal{ALC}, \mathcal{ALCO})$ undecidable ($\mathsf{rel}_1 \cup \mathsf{rel}_2 \cup \mathsf{cons}$ as helper symbols)
5. (Non-)proj. $(\mathcal{ALC}, \mathcal{L}_S)$ undecidable for $\mathcal{L}_S \supseteq \mathcal{ALCFIO}$

IV. *Restricted strong separability case*
For $\mathcal{L} \in \mathbf{DL}_{\mathcal{IO}}$,

1. $(\mathcal{L}, \mathbf{FO})$ Exp-complete
2. $(\mathcal{L}, \mathcal{L})$ 2Exp-complete
3. $(\mathbf{GF}, \mathbf{FO})$ 2Exp-complete
4. $(\mathbf{FO}^2, \mathbf{FO})$ coNExp-complete
5. $(\mathbf{GF}, \mathbf{GF})$ 3Exp-complete
6. $(\mathbf{FO}^2, \mathbf{FO}^2)$ in coN2Exp, 2Exp-hard

References

[ABN98] H. ANDREKA, J. VAN BENTHEM, I. NEMETI (1998). *Modal languages and bounded fragments of predicate logic*, Journal of Philosophical Logic 27 (3) 217-274.

[ABM99] C. ARECES, P. BLACKBURN, AND M. MARX (1999). *A road-map on complexity for hybrid logics*, Volume 1683 of Lecture Notes in Computer Science, 307-321. Springer.

[AD16] M. ARENAS, G. I. DIAZ (2016). *The exact complexity of the first-order logic definability problem*, Association for Computing Machinery Transactions on Database Systems, 41 (2) 13:1-13:14.

[ADK16] M. ARENAS, G. I. DIAZ, E. V. KOSTYLEV (2016). *Reverse engineering SPARQL queries*, Proceedings of the 25th International World Wide Web Conference, 239-249.

[AEF08] C. D'AMATO, F. ESPOSITO, N. FANIZZI (2008). *DL-FOIL: concept learning in description logics*, Proceedings of the 18th Conference on Inductive Logic Programming, 107-121.

[AEFR18a] C. D'AMATO, F. ESPOSITO, N. FANIZZI, G. RIZZO (2018). *DLFoil: Class expression learning revisited*, Proceedings of the 21st International Conference on Knowledge Engineering and Knowledge Management, 98-113.

[AEFR18b] C. D'AMATO, F. ESPOSITO, N. FANIZZI, G. RIZZO (2018). *A framework for tackling myopia in concept learning on the web of data*, Proceedings of the 21st International Conference on Knowledge Engineering and Knowledge Management, 338-354.

[AFR20] C. D'AMATO, N. FANIZZI, G. RIZZO (2020). *Class expression induction as concept space exploration: From dl-foil to dl-focl*, Future Generation Computer Systems 108, 256-272.

[AJMOW21] A. ARTALE, J. C. JUNG, A. MAZZULLO, A. OZAKI, F. WOLTER (2021). *Living without Beth and Craig: Definitions and interpolants in description logics with nominals and role inclusions*, Proceedings of the 35th Conference of the Association for the Advancement of Artificial Intelligence, 6193-6201.

[AKS18] C. ARECES, A. KOLLER, K. STRIEGNITZ (2018). *Referring expressions as formulas of description logic*, Proceedings of the 5th International Natural Language Generation Conference.

[AMOW21] A. ARTALE, A. MAZZULLO, A. OZAKI, F. WOLTER (2021). *On free description logics with definite descriptions*, Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning.

[B66] R. Berger (1966). *The undecidability of the domino problem*, Memoirs of the American Mathematical Society 66.

[BBC13] V. Bárány, M. Benedikt, B. ten Cate (2013). *Rewriting guarded negation queries*, Proceedings of the International Symposium on Mathematical Foundations of Computer Science.

[BBC18] V. Bárány, M. Benedikt, B. ten Cate (2018). *Some model theory of guarded negation*, Journal of Symbolic Logic 83 (4) 1307-1344.

[BBLPW18] S. Bin, L. Buhmann, J. Lehmann, P. Westphal (2018). *DL-learner - structured machine learning on semantic web data*, Proceedings of the 27th International World Wide Web Conference, 467-471.

[BCH92] A. Borgida, W. W. Cohen, H. Hirsh (1992). *Computing least common subsumers in description logics*, Proceedings of the 10th Conference of the Association for the Advancement of Artificial Intelligence, 754-760.

[BCLW14] M. Bienvenu, B. ten Cate, C. Lutz, F. Wolter (2014). *Ontology-based data access: A study through disjunctive datalog, CSP, and MMSNP*, Association for Computing Machinery Transactions on Database Systems 39 (4) 33:1-33:44.

[BCMNP03] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, P. F. Patel-Schneider (Eds.) (2003). *The Description Logic Handbook*, Cambridge University Press.

[BCO12] V. Bárány, B. ten Cate, M. Otto (2012). *Queries with guarded negation*, Proceedings of the 47th International Conference on Very Large Data Bases, 1328-1339.

[BCS15] V. Bárány, B. ten Cate, L. Segoufin (2015). *Guarded negation*, Journal of the Association for Computing Machinery 62 (3) 22:1-22:26.

[BFS13] B. ten Cate, E. Franconi, I. Seylan (2013). *Beth definability in expressive description logics*. Journal of Artificial Intelligence Research 48:347-414.

[BGG97] E. Borger, E. Grädel, Y. Gurevich (1997). *The Classical Decision Problem*, Perspectives in Mathematical Logic, Springer.

[BGO14] V. Bárány, G. Gottlob, M. Otto (2014). *Querying the Guarded Fragment*, Logical Methods in Computer Science 10 (2).

[BHLS17] F. Baader, I. Horrocks, C. Lutz, U. Sattler (2017). *An Introduction to description logics*, Cambridge University Press.

[BKLRWZ16] E. Botoeva, B. Konev, C. Lutz, V. Ryzhikov, F. Wolter, M. Zakharyaschev (2016). *Inseparability and conservative extensions* *of description logic ontologies: A survey*. Proceedings of the 12th Reasoning Web Summer School, 27-89. Springer.

[BKLRWZ19] E. Botoeva, R. Kontchakov, C. Lutz, V. Ryzhikov, F. Wolter, M. Zakharyaschev (2019). *Query inseparability for ALC ontologies*, Artificial Intelligence Journal 272 1-51.

[BKM99] F. Baader, R. Kusters, R. Molitor (1999). *Computing least common subsumers in description logics with existential restrictions*, Proceedings of the 16th International Joint Conference on Artificial Intelligence, 96-103.

[BLW16] L. Buhmann, J. Lehmann, P. Westphal (2016). *DL-learner - A framework for inductive learning on the semantic web*, Journal of Web Semantics 39 (2016) 15-24.

[BN00] L. Badea, S. Nienhuys-Cheng (2000). *A refinement operator for description logics*, Proceedings of the 10th International Conference on Inductive Logic Programming, 40-59.

[BR17] P. Barcelo, M. Romero (2017). *The complexity of reverse engineering problems for conjunctive queries*, Proceedings of the 20th International Conference on Database Theory, 7:1-7:17.

[BRV01] P. Blackburn, M. de Rijke, Y. Venema (2001). *Modal Logic*, Cambridge University Press.

[BTW16] A. Borgida, D. Toman, G. E. Weddell (2016). *On referring expressions in query answering over first order knowledge bases*, Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning, 319-328.

[Co69] S. D. Comer (1969). *Classes without the amalgamation property*. Pacific Journal of Mathematics 28(2):309-318.

[Cr57] W. Craig (1957). *Three uses of the herbrand-gentzen theorem in relating model theory and proof theory*. Journal of Symbolic Logic 22(3):269-285.

[CCL21] G. Cima, F. Croce, M. Lenzerini (2021). *QDEF and its approximations in OBDM*, Proceedings of the 30th ACM International Conference on Information and Knowledge Management, 271-280.

[CCL22] G. Cima, F. Croce, M. Lenzerini (2022). *Separability and its approximations in Ontology-Based Data Management*, Semantic Web Journal.

[CD15] B. ten Cate, V. Dalmau (2015). *The product homomorphism problem and applications,*Proceedings of the 18th International Conference on Database Theory, 161-176.

[CD21] B. ten Cate, V. Dalmau (2021). *Conjunctive queries: Unique characterizations and exact learnability*, Proceedings of the 24th International Conference on Database Theory, 9:1-9:24.

[CGL98] D. Calvanese, G. De Giacomo, M. Lenzerini (1998). *On the decidability of query containment under constraints*, Proceedings of the 17th Symposium on Principles of Database Systems, 149-158.

[CKo81] A. K. Chandra, D. C. Kozen, L. J. Stockmeyer (1981). *Alternation*. Journal of the Association for Computing Machinery 28:114-133.

[CKe98] C. Chang, H. J. Keisler (1998). *Model Theory*, Elsevier.

[CM77] A. K. Chandra, P. M. Merlin (1977). *Optimal implementation of conjunctive queries in relational data bases*, Proceedings of the 9th Annual ACM Symposium on Theory of Computing, 77-90.

[CPT09] C. Chan, S. Parthasarathy, Q. T. Tran (2009). *Query by output*, Proceedings of the 28th Symposium on Principles of Database Systems, 535-548.

[CW17] S. Cohen, Y. Y. Weiss (2017). *Reverse engineering spj-queries from examples*, Proceedings of the 36th Symposium on Principles of Database Systems, 151-166.

[CS13] B. ten Cate, L. Segoufin (2013). *Unary negation*. Logical Methods in Computer Science 9(3).

[DG19] D. Deutch, A. Gilad (2019). *Reverse-engineering conjunctive queries from provenance examples*, Proceedings of the 22nd International Conference on Extending Database Technology, 277-288.

[DGMT17] J. Dietrich, H. W. Guesgen, S. Marsland, A. C. Tran (2017). *Parallel symmetric class expression learning*, Journal of Machine Learning Research 18 64:1-64:34.

[DHNN18] A. R. Divroodi, Q. Ha, L. A. Nguyen, H. S. Nguyen (2018). *On the possibility of correct concept learning in description logics*, Vietnam Journal of Computer Science 5 (1) 3-14.

[DK12] K. van Deemter, E. Krahmer (2012). *Computational generation of referring expressions: A survey*, Computational Linguistics 38 (1) 173-218.

[E97] P. van Emde Boas (1997). *The convenience of tilings*. Complexity, Logic, and Recursion Theory, 331-363.

[EPSZ13] H. Elmeleegy, C. M. Procopiuc, D. Srivastava, M. Zhang (2013). *Reverse engineering complex join queries*, Proceedings of the 2013 ACM SIGMOD Conference on Management of Data, 809-820.

[F19] M. Funk (2019). *Concept-By-Example in $\mathcal{EL}$-knowledge bases*, MSc thesis.

[FIP07] N. Fanizzi, L. Iannone, I. Palmisano (2007). *An algorithm based on counterfactuals for concept learning in the semantic web*, Proceedings of the 18th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems 26 (2) 139-159.

[FJLPW19] M. Funk, J. C. Jung, C. Lutz, H. Pulcini, F. Wolter (2019). *Learning description logic concepts: When can positive and negative examples be separated?*, Proceedings of the 27th International Joint Conference on Artificial Intelligence, 1682-1688.

[G99] E. Gradel (1999). *On the restraining power of guards*. Journal of Symbolic Logic 64(4):1719-1742.

[GHKS08] B. C. Grau, I. Horrocks, Y. Kazakov, U. Sattler (2008). *Modular reuse of ontologies: Theory and practice*. Journal of Artificial Intelligence Research 31:273-318.

[GHKP19] B. C. Grau, I. Horrocks, E. V. Kostylev, A. Petrova (2019). *Query-based entity comparison in knowledge graphs revisited*. Proceedings of the 18th International Semantic Web Conference, 558-575. Springer.

[GHPS17] B. C. Grau, I. Horrocks, A. Petrova, E. Sherkhonov (2017). *Entity comparison in RDF graphs*, Proceedings of the 16th International Semantic Web Conference, 526-541.

[GIJ18] V. Gutiérrez-Basulto, Y. Ibanez-Garcia, J. C. Jung (2018). *Answering regular path queries over SQ ontologies*, Proceedings of the 32nd Conference of the Association for the Advancement of Artificial Intelligence.

[GJS18] V. Gutiérrez-Basulto, J. C. Jung, L. Sabellek (2018). *Reverse engineering queries in ontology-enriched systems: The case of expressive Horn description logic ontologies*, Proceedings of the 27th International Joint Conference on Artificial Intelligence.

[GK72] Y. Gurevich, I. Koryakov (1972) *Remarks on berger's paper on the domino problem*, Siberian Mathematics Journal 13, 319-321.

[GKV97] E. Gradel, P. G. Kolaitis, M. Y. Vardi (1997). *On the decision problem for two-variable first-order logic*, Bulletin of Symbolic Logic 3 (1) 53-69.

[GLW06] S. Ghilardi, C. Lutz, F. Wolter (2006). *Did I damage my ontology? A case for conservative extensions in description logics*, Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning, 187-197.

[GM05] D. GABBAY, L. MAKSIMOVA (2005). *Interpolation and Definability in Modal and Intuitionistic Logics*. Clarendon Press.

[GO07] V. GORANKO, M. OTTO (2007). *Model theory of modal logic*, Handbook of Modal Logic, Elsevier, 249-329.

[GO14] E. GRADEL, M. OTTO (2014). *The freedoms of (guarded) bisimulation*, Johan van Benthem on Logic and Information Dynamics, Springer International Publishing, 3-31.

[H08] S. HELLMANN (2008). *Comparison of concept learning algorithms (with emphasis on ontology engineering for the Semantic Web)*, MSc thesis.

[HHKV12] K. HODER, A. HOLZER, L. KOVACS, A. VORONKOV (2012). *A vampire-based tool for interpolation*. Proceedings of the 10th Asian Symposium on Programming Languages and Systems, 148-156. Springer.

[HHNST12] Q. HA, T. HOANG, L. A. NGUYEN, H. S. NGUYEN, A. SZALAS, T. TRAN (2012). *A bisimulation-based method of concept learning for knowledge bases in description logics*, Proceedings of the 3rd International Symposium on Information and Communication Technology, 241-249.

[HL09] C. HAASE, J. LEHMANN (2009). *Ideal downward refinement in the EL description logic*, Proceedings of the 19th International Conference on Inductive Logic Programming, 73-87.

[HL10] P. HITZLER, J. LEHMANN (2010). *Concept learning in description logics using refinement operators*, Machine Learning 78, 203-250.

[HLPW20] A. HERNICH, C. LUTZ, F. PAPACCHINI, F. WOLTER (2020). *Dichotomies in ontology-mediated querying with the Guarded Fragment*, ACM Trans. Comput. Log. 21 (3) 1-47.

[HM02] E. HOOGLAND, M. MARX (2002). *Interpolation and definability in guarded fragments*. Stud. Log. 70(3):373-409.

[HNT15] T. HOANG, L. A. NGUYEN, T. TRAN (2015). *Bisimulation-based concept learning for information systems in description logics*, Vietnam Journal of Computer Science 2 (3) 149-167.

[HS19] P. HITZLER, M. K. SARKER (2019). *Efficient concept induction for description logics*, Proceedings of the 33th Conference of the Association for the Advancement of Artificial Intelligence, 3036-3043.

[JK84] D. S. JOHNSON, A. C. KLUG (1984). *Testing containment of conjunctive queries under functional and inclusion dependencies*, Journal of Computer and System Sciences 28 (1) 167-189.

[JLPW20] J. C. Jung, C. Lutz, H. Pulcini, F. Wolter (2020). *Logical separability of incomplete data under ontologies*, Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning.

[JLPW21] J. C. Jung, C. Lutz, H. Pulcini, F. Wolter (2021). *Separating data examples by description logic concepts with restricted signatures*, Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning.

[JLPW22] J. C. Jung, C. Lutz, H. Pulcini, F. Wolter (2022). *Logical separability of labeled data examples under ontologies* , Artificial Intelligence, Volume 313, December 2022, 103785.

[JLW20] J. C. Jung, C. Lutz, F. Wolter (2020). *Least general generalizations in description logic: Verification and existence*, Proceedings of the 34th Conference of the Association for the Advancement of Artificial Intelligence, 2854-2861.

[JLZ20] J. C. Jung, C. Lutz, T. Zeume (2020). *On the decidability of expressive description logics with transitive closure and regular role expressions*. Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning.

[JLMSW17] J. C. Jung, C. Lutz, M. Martel, T. Schneider, F. Wolter (2017). *Conservative extensions in guarded and two-variable fragments*, Proceedings of the 44th International Colloquium on Automata, Languages and Programming, 108:1-108:14.

[JW21] J. C. Jung, F. Wolter (2021). *Living without Beth and Craig: Definitions and interpolants in the guarded and two-variable fragments*, Proceedings of the 36th Annual Symposium on Logic in Computer Science 21, 1-14.

[KM22] P. Krogmeier, P. Madhusudan (2022). *Learning Formulas in Finite Variable Logics*, Proceedings of the ACM on Programming Languages, 6. Art. 10.

[KLS18] D. V. Kalashnikov, L. V. Lakshmanan, D. Srivastava (2018). *Fastqre: Fast query reverse engineering*, Proceedings of SIGMOD 18, 337-350.

[KLWW09] B. Konev, C. Lutz, D. Walther, F. Wolter (2009). *Formal properties of modularisation. In Modular Ontologies*, volume 5445 of Lecture Notes in Computer Science. Springer. 25-66.

[KR18] B. Kimelfeld, C. Ré (2018). *A relational framework for classifier engineering*, Association for Computing Machinery Transactions on Database Systems 43 (3) 11:1-11:36.

[Li04] L. LIBKIN (2004). *Elements of Finite Model Theory*, Springer.

[Li12] F. A. LISI (2012). *A formal characterization of concept learning in description logics*, Proceedings of the 25th International Workshop on Description Logics.

[Li16] F. A. LISI (2016). *A model+solver approach to concept learning*, Proceedings of the 15th International Conference of the Italian Association for Artificial Intelligence, 266-279.

[Lu02] C. LUTZ (2002). *The Complexity of Reasoning with Concrete Domains*, PhD thesis, RWTH Achen.

[Lu07] C. LUTZ (2007). *Inverse roles make conjunctive queries hard*, Proceedings of the 20th International Workshop on Description Logics.

[Lu08] C. LUTZ (2008). *The complexity of conjunctive query answering in expressive description logics*, Proceedings of the 4th International Joint Conference on Automated Reasoning, 179-193.

[LPW11] C. LUTZ, R. PIRO, F. WOLTER (2011). *Description logic TBoxes: Model-theoretic characterizations and rewritability*, Proceedings of the 20th International Joint Conference on Artificial Intelligence.

[LS15] F. A. LISI, U. STRACCIA (2015). *Learning in description logics with fuzzy concrete domains*, Fundamenta Informaticae 140 (3-4) 373-391.

[LV14] J. LEHMANN, J. VÖLKER (Eds.) (2014). *Perspectives on Ontology Learning*, Studies on the Semantic Web 18.

[LWW07] C. LUTZ, D. WALTHER, F. WOLTER (2007). *Conservative extensions in expressive description logics*. Proceedings of the 16th International Joint Conference on Artificial Intelligence, 453-458.

[Mark02] D. MARKER (2002). *Model Theory: an Introduction*, Springer.

[Mart19] D. M. L. MARTINS (2019). *Reverse engineering database queries from examples: State-of-the-art, challenges, and research opportunities*, Information Systems 83, 89-100.

[Mo75] M. MORTIMER (1975). *On languages with two variables*, Mathematical Logic Quarterly 21 (1) 135-140.

[Mu91] S. MUGGLETON (1991). *Inductive Logic Programming*, New Generation Computing, 8(4):295-318

[MS01] A. MAEDCHE, S. STAAB (2001). *Ontology learning for the semantic web*. IEEE Intelligent Systems, 16(2):7279.

[MS15] M. MUCCI, U. STRACCIA (2015). *pFOIL-DL: Learning (fuzzy) EL concept descriptions from crisp OWL data using a probabilistic ensemble* *estimation*, Proceedings of the 30th Annual ACM Symposium on Applied Computing, 345-352.

[N90] B. NEBEL (1990). *Reasoning and Revision in Hybrid Representation Systems*, Springer.

[O19] M. ORTIZ (2019). *Ontology-mediated queries from examples: a glimpse at the DL-Lite case*, Proceedings of the 5th Global Conference on Artificial Intelligence, 1-14.

[Pi71] D. PIGOZZI (1971). *Amalgamation, congruence-extension, and interpolation properties in algebras*, Algebra Universalis (1):269-349.

[Pr09] I. PRATT-HARTMANN (2009). *Data-complexity of the two-variable fragment with counting quantifiers*, Information and Computation 207 (8) 867-888.

[Ro07] R. ROSATI (2007). *The limits of querying ontologies*, Proceedings of the 11th International Conference on Database Theory, 164-178.

[Ro11] R. ROSATI (2011). *On the finite controllability of conjunctive query answering in databases under open-world assumption*, Journal of Computer and System Sciences 77 (3) 572-594.

[Ru86] V. RUTHENBURG (1986). *Complexity of generalized graph coloring.*, Proceedings of the International Symposium on Mathematical Foundations of Computer Science, 573-581.

[Sa70] W. J. SAVITCH (1970). *Relationships between nondeterministic and deterministic tape complexities*. Journal of Computer and System Sciences, 4(2): 177-192.

[Sc91] K. SCHILD (1991). *A correspondence theory for terminological logics: Preliminary report*. Proceedings of the 12th International Joint Conference on Artificial Intelligence, 466-471.

[Sp93] E. SPAAN (1993). *Complexity of Modal Logics*. PhD thesis, University of Amsterdam.

[SW12] S. STAWORKO, P. WIECZOREK (2012). *Learning twig and path queries*, Proceedings of the 16th International Conference on Database Theory, pp. 140-154.

[TW19] D. TOMAN, G. E. WEDDELL (2019). *Finding ALL answers to OBDA queries using referring expressions*, Proceedings of the 32nd Australasian Joint Conference on Artificial Intelligence, 117-129.

[TZ13] A. TURHAN, B. ZARRIESS (2013). *Most specific generalizations w.r.t. general EL-TBoxes*, Proceedings of the 22th International Joint Conference on Artificial Intelligence, 1191-1197.

[V98] M. Y. Vardi (1998). *Reasoning about the past with two-way automata*, Proceedings of the 25th International Colloquium on Automata, Languages and Programming, 628-641.