Logical Separability of Incomplete Data under Ontologies

Jean Christoph Jung[1] , **Carsten Lutz**[1] , **Hadrien Pulcini**[2] , **Frank Wolter**[2]

[1]{jeanjung,clu}@uni-bremen.de, University of Bremen, Germany
[2]{h.pulcini,wolter}@liverpool.ac.uk, University of Liverpool, UK

Abstract

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

1 Introduction

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

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

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

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

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

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

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

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

2 Related Work and Applications

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

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

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

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

3 Preliminaries

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

4 Fundamental Results

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

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

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

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

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

The following example illustrates the definition.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

5 Results on Separability

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

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

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

We start, however, with two observations on projective separability. It is shown in (Funk et al. 2019) that a labeled $\mathcal{ALCI}$-KB $(\mathcal{K}, P, N)$ is projectively $\mathcal{ALCI}$-separable iff Condition 4 from Theorem 1 holds. We thus obtain the following.[2]

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

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

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

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

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

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

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

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

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

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

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

We use bisimulations between pointed structures, defined in the standard way but restricted to a signature Σ, see e.g. (Lutz, Piro, and Wolter 2011; Goranko and Otto 2007) for details. With $\mathfrak{A}, a \sim_{\mathcal{ALCI},\Sigma} \mathfrak{B}, b$, we indicate that there is a Σ-bisimulation between $\mathfrak{A}$ and $\mathfrak{B}$ that contains (a,b).

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

For a KB $\mathcal{K}$, we use $\mathsf{cl}(\mathcal{K})$ to denote the set of concepts in $\mathcal{K}$ and the concepts $\exists R.\top$ and $\exists R^-.\top$ for all role names $R \in \mathsf{sig}(\mathcal{K})$, closed under subconcepts and single negation. A *$\mathcal{K}$-type* is a set $t \subseteq \mathsf{cl}(\mathcal{K})$ such that there exists a model $\mathfrak{A}$ of $\mathcal{K}$ and an $a \in \mathrm{dom}(\mathfrak{A})$ with $\mathrm{tp}_{\mathcal{K}}(\mathfrak{A}, a) = t$ where

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

5.2 Separability of GF-KBs

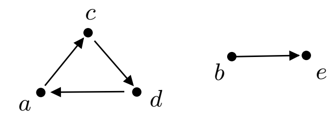
We study projective and non-projective (GF, GF)-separability which turns out to behave similarly to the $\mathcal{ALCI}$ case in many ways. The results are, however, significantly more difficult to establish.

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

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

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

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



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

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

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

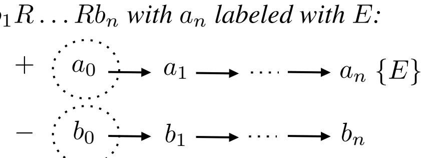
Theorem 6 *(GF, GF)-separability coincides with (GF, openGF)-separability, both in the projective and in the non-projective case.*

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

Example 6 *Let*

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

and let $\mathcal{D}$ contain two R-paths of length n, $a_0 R a_1 R \ldots R a_n$ and $b_0 R b_1 R \ldots R b_n$ with a_n labeled with E:



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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

5.3 Separability of FO^2-KBs

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

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

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

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

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

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

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

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

6 Strong Separability

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

7 Results on Strong Separability

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

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

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

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

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

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

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

7.2 Strong Separability of GF-KBs

We start with observing a counterpart of Theorem 6.

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

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

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

7.3 Strong Separability of FO^2-KBs

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

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

8 Conclusion

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

Acknowledgements

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

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Lutz, C. 2008b. Two upper bounds for conjunctive query answering in $\mathcal{SHIQ}$. In *Proc. of DL*.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

A Preliminaries

We start by introducing tree-shaped structures and forest models for $\mathcal{ALCI}$-KBs. Assume $\mathfrak{A}$ is a structure such that $R^{\mathfrak{A}} = \emptyset$ for any relation symbol R of arity > 2. We associate with $\mathfrak{A}$ an undirected graph $G_{\mathfrak{A}}$ that has the set of vertices dom($\mathfrak{A}$) and an edge $\{d, e\}$ whenever $(d, e) \in R^{\mathfrak{A}}$ for some role R. We say that $\mathfrak{A}$ is *tree-shaped* if $G_{\mathfrak{A}}$ is a tree without reflexive loops and $R^{\mathfrak{A}} \cap S^{\mathfrak{A}}$ if $R \neq S$. We say that $\mathfrak{A}$ has *finite outdegree* if $G_{\mathfrak{A}}$ has finite outdegree. We call a model $\mathfrak{A}$ of an $\mathcal{ALCI}$ KB $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ a *forest model* of $\mathcal{K}$ if $\mathfrak{A}$ is the disjoint union of tree-shaped models $\mathfrak{B}_{a^{\mathfrak{A}}}$, for $a \in \text{cons}(\mathcal{D})$, extended by all $R(a^{\mathfrak{A}}, b^{\mathfrak{A}})$ with $R(a, b) \in \mathcal{D}$. The following result is well known.

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

Now let $\mathfrak{A}$ be an arbitrary structure. The *Gaifman graph* $G_{\mathfrak{A}}$ of $\mathfrak{A}$ has the set of vertices dom($\mathfrak{A}$) and an edge $\{d, e\}$ whenever there exists $\vec{a} \in R^{\mathfrak{A}}$ containing d, e for some relation R. A *path* of length n from a to b in $\mathfrak{A}$ is a sequence $R_1(\vec{b}_1), \ldots, R_n(\vec{b}_n)$ with

- $\mathfrak{A} \models R_i(\vec{b}_i)$ and $|[\vec{b}_i]| \geq 2$ for all $i \leq n$;
- $a \in [\vec{b}_1], b \in [\vec{b}_n]$;
- $[\vec{b}_i] \cap [\vec{b}_{i+1}] \neq \emptyset$, for all $i < n$.

We call a path p *strict* if all $[\vec{b}_i] \cap [\vec{b}_{i+1}]$ are singletons containing distinct points c_i and there are sets $A_1, \ldots, A_n \subseteq$ dom($\mathfrak{A}$) covering dom($\mathfrak{A}$) such that $[\vec{b}_i] \subseteq A_i$, $A_i \cap A_{i+1} = \{c_i\}$ 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 *distance* $\text{dist}_{\mathfrak{A}}(a, b)$ between $a, b \in \text{dom}(\mathfrak{A})$ is defined as the length of a shortest path from a to b, if such a path exists. Otherwise $\text{dist}_{\mathfrak{A}}(a, b) = \infty$. The *maximal connected component (mcc)* $\mathfrak{A}_{\text{con}(\vec{a})}$ of $\vec{a}$ in $\mathfrak{A}$ is the substructure of $\mathfrak{A}$ induced by the set of all b such that there exists $a \in [\vec{a}]$ with $\text{dist}_{\mathfrak{A}}(a, b) < \infty$.

B Proofs for Section 5.1: Data Complexity

Our aim is to prove Theorem 3. We start with the $\mathcal{ALC}$-part.

Theorem 18 *Projective $(\mathcal{ALC}, \mathcal{ALC})$-separability is* PSPACE*-complete in data complexity.*

Proof. For the lower bound, by Corollary 2 and Theorem 1 it suffices to show that rooted UCQ evaluation on $\mathcal{ALC}$-KBs $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ with the Gaifman graph of $\mathcal{D}$ connected is PSPACE-hard. The reduction is from QBF validity. We first define the ontology $\mathcal{O}$ as:

$$\begin{array}{rcl} E & \sqsubseteq & \exists R.(U \sqcap T) \sqcap \exists R.(U \sqcap F) \\ U & \sqsubseteq & \exists R.(E \sqcap (T \sqcup F)) \\ T & \equiv & \neg F \\ \top & \sqsubseteq & \exists S.\top \end{array}$$

and also fix a database $\mathcal{D} = \{U(a_0), M(a_0)\}$. Set $\mathcal{K} = (\mathcal{O}, \mathcal{D})$. We assume w.l.o.g. that the input QBF is of the form

$$\varphi = \forall x_1 \exists x_2 \forall x_3 \cdots \exists x_n \, \psi$$

with $\psi = \psi_1 \wedge \cdots \wedge \psi_m$ in KNF. We show how to construct in polynomial time a rooted UCQ q_φ such that φ is valid iff $\mathcal{K} \not\models q_\varphi(a_0)$. The UCQ q_φ consists of the following CQs where x_0 is the (only) answer variable and we use $R^i(x,y)$, $i \geq 1$, as shorthand for $R(z_1,z_2),\dots,R(z_{i-1},z_i)$ with $z_1 = x$, $z_i = y$, and $z_2,\dots,z_{i-1}$ fresh variables:

- memorize chosen truth values; for $1 \leq i,j \leq n$ with $i+j=n$:

  $$M(x_0), R^i(x_0,x_1), T(x_1), R^j(x_1,x_2), S^i(x_2,x_3), F(x_3)$$

- make sure ψ is satisfied; for $1 \leq i \leq m$ with $\psi_i = \ell_1 \vee \cdots \vee \ell_k$ and the variable in ℓ_j being p_{i_j} for $1 \leq j \leq n$:

  $$M(x_0), R^n(x_0,x_1),$$
  $$S^{i_1}(x_1,y_1), \overline{V}_1(y_1), \dots, S^{i_k}(x_1,y_k), \overline{V}_k(y_k)$$

  where $\overline{V}_j = F$ if the literal ℓ_j is positive and $\overline{V}_j = T$ otherwise.

It can be verified that q_φ is as required.

For the upper bound, it suffices to show that rooted UCQ evaluation in $\mathcal{ALC}$ is in PSPACE when the ontology is fixed. We do not rely on the assumption that the Gaifman graph of the database is connected. An *$\mathcal{EL}$-concept* is an $\mathcal{ALC}$-concept that uses only the constructors $\top$, $\sqcap$, and $\exists r.C$. An *augmented database* is a database that may contain 'atoms' $\neg C(a)$, C an $\mathcal{EL}$-concept and an *augmented $\mathcal{ALC}$-KB* 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 (**?**) that given an $\mathcal{ALC}$-KB $(\mathcal{O},\mathcal{D})$ and a Boolean CQ q, one can compute a sequence of augmented $\mathcal{ALC}$-KBs $\mathcal{K}_1,\dots,\mathcal{K}_n$, $\mathcal{K}_i = (\mathcal{O}_i,\mathcal{D})$ such that $\mathcal{K} \not\models q$ iff at least one $\mathcal{K}_i$ is satisfiable. The proof straightforwardly extends to non-Boolean UCQs q 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 KBs $\mathcal{K}_1,\dots,\mathcal{K}_n$, 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 KB $(\mathcal{O},\mathcal{D})$ is satisfiable.

We only sketch the procedure. Let $\mathsf{cl}(\mathcal{O})$ denote the set of subconcepts of (concepts that occur in) $\mathcal{O}$ closed under single negation. A *type* t is a subset of $\mathsf{cl}(\mathcal{O})$. We can precompute in constant time the set S of types that are satisfiable w.r.t. $\mathcal{O}$. To check satisfiability of $(\mathcal{O},\mathcal{D})$, we first guess an assignment $\delta : \mathsf{dom}(\mathcal{D}) \to S$ of satisfiable types to constants in $\mathcal{D}$ such that whenever $r(a,b) \in \mathcal{D}$ and $\exists r.C \in \mathsf{cl}(\mathcal{O})$, then $C \in \delta(b)$ implies $\exists r.C \in \delta(a)$. It remains to check whether for every $a \in \mathsf{dom}(\mathcal{D})$, the concept $\bigsqcap_{C(a)\in\mathcal{D}} C$ is satisfiable w.r.t. $\mathcal{O}$. This can be done using a minor variation of the standard $\mathcal{ALC}$-worlds style procedure that decides in PSPACE the satisfiability of an $\mathcal{ALC}$-concept C without ontologies by verifying the existence tree-shaped model of depth $||C||$ using a depth-first traversal (Baader et al. 2017). In our case, we have to take into account the ontology $\mathcal{O}$. But since it is fixed, we we can still use the same algorithm searching for a tree model of depth $||\bigsqcap_{C(a)\in\mathcal{D}} C||$: as long as we make sure that all occurring types are from S, it is guaranteed that we can extend the identified initial piece of a tree model to an infinite tree model that satisfies not only $\bigsqcap_{C(a)\in\mathcal{A}} C$, but also $\mathcal{O}$. ❑

Theorem 19 *Projective $(\mathcal{ALCI},\mathcal{ALCI})$-separability is* NEXPTIME*-complete in data complexity.*

The upper bound is immediate since rooted UCQ evaluation on $\mathcal{ALCI}$-KBs is in CONEXPTIME (**?**). For the lower bound, it suffices to show that the following problem is CONEXPTIME-hard for every fixed $\mathcal{ALCI}$-ontology $\mathcal{O}$: given a database $\mathcal{D}$ whose Gaifman graph is connected, a unary rooted UCQ q, and an $a \in \mathsf{dom}(\mathcal{D})$, decide whether $(\mathcal{O},\mathcal{D}) \models q(a)$. To achieve this, we adapt a CONEXPTIME-hardness proof from (Lutz 2008).

A *tiling system* $\mathfrak{T}$ is a triple (T,H,V), where $T = \{0,1,\dots,k-1\}$, $k \geq 0$, is a finite set of *tile types* and $H,V \subseteq T \times T$ represent the *horizontal and vertical matching conditions*. Let $\mathfrak{T}$ be a tiling system and $c = c_0 \cdots c_{n-1}$ an *initial condition*, i.e. an n-tuple of tile types. A mapping $\tau : \{0,\dots,2^n-1\} \times \{0,\dots,2^n-1\} \to T$ is a *solution* for $\mathfrak{T}$ and c if for all $x,y < 2^n$, the following holds where $\oplus_i$ denotes addition modulo i:

1. if $\tau(x,y) = t$ and $\tau(x \oplus_{2^n} 1, y) = t'$, then $(t,t') \in H$;
2. if $\tau(x,y) = t$ and $\tau(x, y \oplus_{2^n} 1) = t'$, then $(t,t') \in V$;
3. $\tau(i,0) = c_i$ for $i < n$.

It is well-known that there is a tiling system $\mathfrak{T}$ such that it is NEXPTIME-hard to decide, given an initial condition c, whether there is a solution for $\mathfrak{T}$ and c. For what follows, fix such a system $\mathfrak{T}$.

We first define the fixed ontology $\mathcal{O}$. We use S to denote the role composition $R_0; R_0^-$. It is convenient to think of S as a symmetric role. To represent tiles, we introduce a concept name D_i for each $i \in T$. We write $\exists R.C$ as shorthand for

$$\exists S.(B_1 \sqcap \exists S.(B_2 \sqcap \exists S.(B_3 \sqcap C))).$$

Now $\mathcal{O}$ contains the following:

$$\begin{array}{rcl}
A & \sqsubseteq & \exists R.(A \sqcap T) \sqcap \exists R.(A \sqcap F) \\
A & \sqsubseteq & \bigsqcap_{1\leq i\leq 3} \exists R.(H' \sqcap \exists R.(G' \sqcap G'_i)) \\
\neg M \sqcap H' & \sqsubseteq & H \\
\neg M \sqcap G' & \sqsubseteq & G \\
\neg M \sqcap G'_i & \sqsubseteq & G_i \text{ for } 1 \leq i \leq 3 \\
G & \sqsubseteq & \bigsqcup_{i\in T}(D_i \sqcap \bigsqcap_{j\in T\setminus\{i\}} \neg D_j) \\
H & \sqsubseteq & \bigsqcup_{i\in T}(\neg D_i \sqcap \bigsqcap_{j\in T\setminus\{i\}} D_j) \\
T & \equiv & \neg F \\
\top & \sqsubseteq & \exists R_1.\top
\end{array}$$

Let $c = c_0 \cdots c_{n-1}$ be an initial condition for $\mathfrak{T}$. We aim to construct a rooted UCQ q_c such that there is a solution for $\mathfrak{T}$ and c iff $(\mathcal{O},\mathcal{D}) \not\models q_c(a_0)$ where $\mathcal{D} = \{A(a_0)\}$. When defining CQs, we write $S(x,y)$ as shorthand for

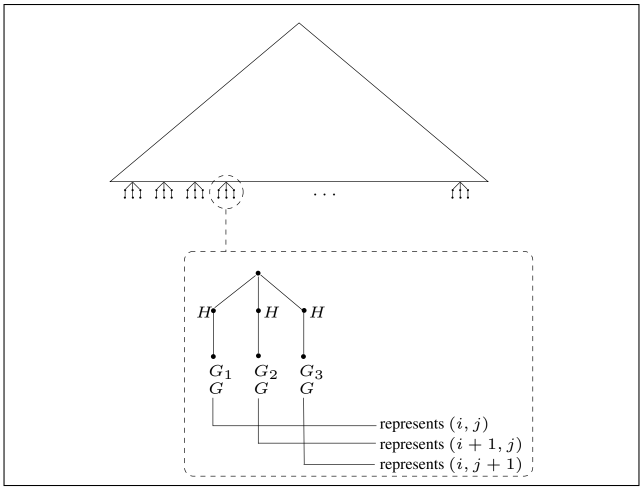


Figure 1: The structure encoding the $2^n \times 2^n$-grid.

$r(x, z), r(y, z)$ with z a fresh variable and $R(x, y)$ as shorthand for $S(x, z_1), S(z_1, z_2), S(z_2, y)$ with z_1, z_2 fresh variables. Note that this corresponds to the $\exists R.C$ abbreviation used in the construction of $\mathcal{O}$, but without the concept names B_1, B_2, B_3 (we only need those to make sure that there is really 'progress' whenever we introduce new successors in Line 1 of $\mathcal{O}$). We further use $R^i(x, y)$, $i \geq 1$, as shorthand for $R(z_1, z_2), \ldots, R(z_{i-1}, z_i)$ where $z_1 = x$, $z_i = y$, and $z_2, \ldots, z_{i-1}$ are fresh variables. We start with several CQs in the UCQ q_c that are comparably simple to construct. In each of them, x_0 is the (only) answer variable.

Our first aim is to generate an R-tree of depth $2n$ whose leaves are the roots of additional depth two gadgets as shown in Figure 1 where all edges are 'R-edges' (in the sense of the abbreviation defined above) and H, G, G_1, G_2, G_3 are concept names. As shown in the figure, every G-node represents a position in the $2^n \times 2^n$-grid. This representation is in binary, that is, we encode the numbers $0, \ldots, 2^{2n} - 1$ in binary by assuming bit i to be one at a domain element d if $d \in (\exists R_1^{i+1}.T)^{\mathcal{I}}$ and zero if $d \in (\exists R_1^{i+1}.F)^{\mathcal{I}}$ where bit 0 is the least significant bit. In principle, the tree and gadgets are already generated by $\mathcal{O}$. However, $\mathcal{O}$ uses concept names H', G', and G'_i in place of F, G, and G_i. We still need to 'activate' the non-primed versions at the right depth via the concepts $\neg M$. This is the purpose of our first CQs. Note that, for this and all following CQs, we are interested in models in which they are *false* in the sense that there is no homomorphism h from the CQ to models of the form shown in Figure 1 such that h maps x_0 to the root of the tree. The first CQs work together with Lines 3-5 of $\mathcal{O}$. They are

$$R^{2n+1}(x_0, x_1), M(x_1)$$

and

$$R^{2n+2}(x_0, x_1), M(x_1).$$

We next achieve the correct labeling with bits at the G_1-nodes. The correct counting is given by the T and F concept names used on the path of the tree that leads to the G_1-node. We still need to 'push it down' to R_1-paths below G-nodes to achieve the desired representation. For $1 \leq i, j \leq n$ with $i + j = 2n$, include the CQ

$$R^i(x_0, x_1), T(x_1), R^{j+2}(x_1, x_2), G_1(x_2), R_1^i(x_2, x_3), F(x_3).$$

To make the counter value unique, we should also ensure that $\exists r^i.T$ and $\exists r^i.F$ are not both true. We need this both for G-nodes and for H-nodes. From now on, we use $(x \text{ bit } i = V)$, $0 \leq i < 2n$ and $V \in \{T, F\}$, to abbreviate $R_1^{i+1}(x, y), V(y)$ for a fresh variable y. For $0 \leq i < 2n$, add the CQs

$$R^{2n+1}(x_0, x_1), H(x_1), (x_1 \text{ bit } i = T), (x_1 \text{ bit } i = F)$$

and

$$R^{2n+2}(x_0, x_1), G(x_1), (x_1 \text{ bit } i = T), (x_1 \text{ bit } i = F).$$

We further want that, relative to its G_1-sibling, each G_2 node represents the horizontal neighbor position in the grid and each G_3-node represents the vertical neighbor position. This can be achieved by a couple of additional CQs that are slightly tedious. We only give CQs which express that if a G_1-node represents (i, j), then its G_2-sibling represents $(i \oplus_{2^n} 1, \ell)$ for some ℓ. For $0 \leq i < n$, put

$$R^{2n}(x_0, x_1), R^2(x_1, y_1), G_1(y_1), \bigwedge_{0 \leq j \leq i} (y_1 \text{ bit } j = T)$$
$$R^2(x_1, y_2), G_2(y_2), (y_2 \text{ bit } i = F)$$

and

$$R^{2n}(x_0, x_1), R^2(x_1, y_1), G_1(y_1), \bigwedge_{0 \leq j < i} (y_1 \text{ bit } j = T),$$
$$(y_1 \text{ bit } i = F) R^2(x_1, y_2), G_2(y_2), (y_2 \text{ bit } i = T)$$

and for $0 \leq j < i < n$, put

$$R^{2n}(x_0, x_1), R^2(x_1, y_1), G_1(y_1), (y_1 \text{ bit } j = F),$$
$$(y_1 \text{ bit } i = F), R^2(x_1, y_2), G_2(y_2), (y_2 \text{ bit } i = T)$$

and

$$R^{2n}(x_0, x_1), R^2(x_1, y_1), G_1(y_1), (y_1 \text{ bit } j = F),$$
$$(y_1 \text{ bit } i = T), R^2(x_1, y_2), G_2(y_2), (y_2 \text{ bit } i = F).$$

Due to Line 6 of $\mathcal{O}$, every G-node is labeled with D_i for a unique $i \in \mathfrak{T}$. The initial condition is now easily guaranteed. For $0 \leq i < n$, and each $j \in \mathfrak{T} \setminus \{c_i\}$, add the CQ

$$R^{2n+2}(x_0, x_1), G(x_1), D_j(x_1),$$
$$(y_1 \text{ bit } 0 = V_0), \ldots, (y_1 \text{ bit } n - 1 = V_{n-1})$$

where V_i is T if the i-th bit in the binary representation of i is 1 and F otherwise. To enforce the matching conditions, we proceed in two steps. First we ensure that they are satisfied locally, i.e., among the three G-nodes in each gadget. For all $i, j \notin H$, put

$$R^{2n}(x_0, x_1), R^2(x_1, y_1), G_1(y_1), T_i(y_1),$$
$$R^2(x_1, y_2), G_2(y_2), T_j(y_2)$$

and likewise for all $i, j \notin V$ and G_3 in place of G_2.

Second, we enforce the following condition, which together with local satisfaction of the matching conditions ensures their global satisfaction:

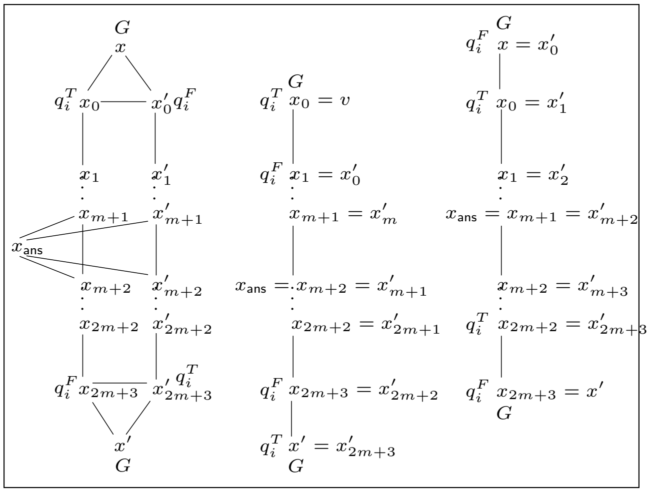


Figure 2: The query q^i (left) and two collapsings.

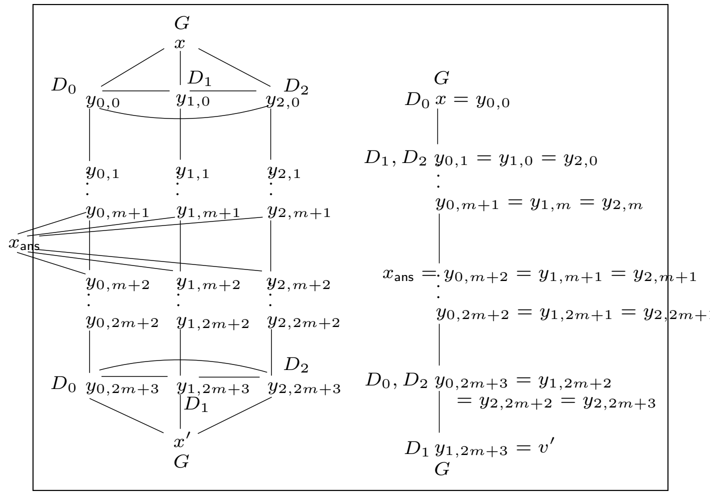


Figure 3: The query q_{tile} (left) and one collapsing.

($*$) if two G-nodes represent the same grid position, then their tile types coincide.

In ($*$), a G-node can be any of a G_1-, G_2-, or G_3-node. To enforce ($*$), we use a CQ that is less straightforward to construct and was first used in (Lutz 2008). To prepare, we first need more CQs to enforce two technical conditions that will be explained later: if d is an H-node and e its G-node successor, then

T1 d satisfies $\exists s^i.T$ iff e satisfies $\exists s^i.F$ for $1 \leq i \leq 2n$;

T2 if d satisfies D_j, then e satisfies $\neg D_j \sqcap \displaystyle\bigsqcap_{\ell \in T \setminus \{j\}} D_j$.

We can enforce T1 using for $0 \leq i < 2n$ the CQs

$$R^{m+1}(x_0, x_1), H(x_1), (x_1 \text{ bit } i = T),$$
$$R(x_1, x_2), G(x_2), (x_2 \text{ bit } i = T)$$

and

$$R^{m+1}(x_0, x_1), H(x_1), (x_1 \text{ bit } i = F),$$
$$R(x_1, x_2), G(x_2), (x_2 \text{ bit } i = F).$$

We now construct a CQ q that does not match the grid representation iff ($*$) is satisfied. In other words, q matches the grid representation iff there are two G-nodes that agree on the grid position but are labelled with different tile types. Such a CQ has first been constructed in (Lutz 2008) and has been reused several times. We only adapt it in a minor way. We confine ourselves to a graphical presentation. q consists of $2n+1$ components $q_0, \dots, q_{2n-1}, q_{\mathsf{tile}}$ that all share three variables x_{ans}, x, and x' with x_{ans} being the (only) answer variable. The CQ q_i, for $0 \leq i < 2n$, is shown in Figure 2. There, every edge is an R-edge and $q_i^V(y)$, $V \in \{T, F\}$ is shorthand for the CQ (y bit $i = V$) whose edges we do not show. The most important property of q_i is that it can collapse (by identifying variables) in the two ways shown in the middle and right of Figure 2. Informally, the middle collapsing represents the fact that bit i is one and the right collapsing represents that bit i is zero. The homomorphisms h from $q_0 \wedge \dots \wedge q_{2n-1}$ to the grid representation with $h(x_{\mathsf{ans}})$ the root of the tree connect via $h(x)$ and $h(x')$ exactly those G-nodes that represent the same grid position. We refer to (Lutz 2008) for more detailed arguments.

The additional component q_{tile} of q is shown in Figure 3 for the case where $\mathfrak{T} = \{0, 1, 2\}$. It should be obvious how to generalize this to different tile sets. There are six (in general $|\mathcal{O}| \cdot (|\mathcal{O}| - 1)$) relevant collapsings of q_{tile}, very similar to those of the CQs q_i. While collapsings of q_i that have a match in the grid representation have either q_i^T or q_i^F on both ends of the resulting path, the q_{tile} collapses into paths that have D_i and D_j at the ends for any distinct $i, j \in \mathfrak{T}$. It should now be obvious that the desired property ($*$) holds iff the root of the grid presentation is not an answer to q.

C Proofs for Section 5.1

Let Σ be a signature and $\mathfrak{A}$ and $\mathfrak{B}$ structures. A relation $S \subseteq \text{dom}(\mathfrak{A}) \times \text{dom}(\mathfrak{B})$ is a Σ-*bisimulation* between $\mathfrak{A}$ and $\mathfrak{B}$ if the following conditions hold for all concept names $A \in \Sigma$ and roles R over Σ:

(atom) for all $(d, e) \in S$, $d \in A^{\mathfrak{A}}$ iff $e \in A^{\mathfrak{B}}$;

(forth) if $(d, e) \in S$ and $(d, d') \in R^{\mathfrak{A}}$, then there exists e' with $(e, e') \in R^{\mathfrak{B}}$ and $(d', e') \in S$;

(back) if $(d, e) \in S$ and $(e, e') \in R^{\mathfrak{B}}$, then there exists d' with $(d, d') \in R^{\mathfrak{A}}$ and $(d', e') \in S$.

We say that pointed structures $\mathfrak{A}, a$ and $\mathfrak{B}, b$ are Σ-*bisimilar* and write $\mathfrak{A}, a \sim_{\mathcal{ALCI},\Sigma} \mathfrak{B}, b$ if there exists a Σ-bisimulation between $\mathfrak{A}$ and $\mathfrak{B}$ containing (a, b).

The following bisimulation based characterization of $(\mathcal{ALCI}, \mathcal{ALCI})$-separability can be proved using the fact that bisimulations characterize the expressive power of $\mathcal{ALCI}$ (Lutz, Piro, and Wolter 2011; Goranko and Otto 2007).

Theorem 20 *Let $(\mathcal{K}, P, \{b\})$ be a labeled $\mathcal{ALCI}$-KB and $\Sigma = sig(\mathcal{K})$. Then the following conditions are equivalent:*

1. *$(\mathcal{K}, P, \{b\})$ is $\mathcal{ALCI}$-separable;*

2. *there exists a forest model $\mathfrak{A}$ of $\mathcal{K}$ of finite outdegree such that for all $a \in P$: there exists no model $\mathfrak{B}$ of $\mathcal{K}$ with $\mathfrak{B}, a^{\mathfrak{B}} \sim_{\mathcal{ALCI},\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$.*

We now refine this characterization further. Let Σ be a signature. A relation S between $\mathsf{cons}(\mathcal{D})$ and $\mathrm{dom}(\mathfrak{A})$ is an $\mathcal{ALCI}(\Sigma)$*-embedding* if the following conditions hold for all concept names A and roles names R:

(atom) if $(a, b) \in S$ and $A(a) \in \mathcal{D}$, then $b \in A^{\mathfrak{A}}$;

(bisim) if $(a, b), (a, b') \in S$, then $\mathfrak{A}, b \sim_{\mathcal{ALCI},\Sigma} \mathfrak{A}, b'$;

(forth) if $R(a, a') \in \mathcal{D}$ and $(a, b) \in S$, then there exists b' with $(b, b') \in R^{\mathfrak{A}}$ and $(a', b') \in S$.

We write $\mathcal{D}, a \preceq_\Sigma \mathfrak{A}, b^{\mathfrak{A}}$ iff there exists an $\mathcal{ALCI}(\Sigma)$ embedding S with $(a, b^{\mathfrak{A}}) \in S$. We obtain the following characterization using Theorem 20.

Theorem 21 *Let $(\mathcal{K}, P, \{b\})$ be a labeled $\mathcal{ALCI}$-KB and $\Sigma = \mathsf{sig}(\mathcal{K})$. Then the following conditions are equivalent:*

1. *$(\mathcal{K}, P, \{b\})$ is $\mathcal{ALCI}$-separable;*
2. *there exists a forest model $\mathfrak{A}$ of $\mathcal{K}$ of finite outdegree such that for every $a \in P$: $\mathcal{D}_{con(a)}, a \not\preceq_\Sigma \mathfrak{A}, b^{\mathfrak{A}}$.*

Let $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ be an $\mathcal{ALCI}$-KB and $\Sigma = \mathsf{sig}(\mathcal{K})$. We give a syntactic characterization of when a $\mathcal{K}$-type t is $\mathcal{ALCI}$-complete. Let R be a role. We say that $\mathcal{K}$-types t_1 and t_2 are *R-coherent* if there exists a model $\mathfrak{A}$ of $\mathcal{K}$ and nodes d_1 and d_2 realizing t_1 and t_2, respectively, such that $(d_1, d_2) \in R^{\mathfrak{A}}$. We write $t_1 \rightsquigarrow_R t_2$ in this case. A sequence

$$\sigma = t_0 R_0 \ldots R_n t_{n+1} \tag{1}$$

of $\mathcal{K}$-types $t_0, \ldots, t_{n+1}$ and Σ-roles $R_0, \ldots, R_n$ *witnesses $\mathcal{ALCI}$-incompleteness of a $\mathcal{K}$-type t* if $t = t_0$, $n \geq 1$, and

- $t_i \rightsquigarrow_{R_{i+1}} t_{i+1}$ for $i \leq n$;
- there exists a model $\mathfrak{A}$ of $\mathcal{K}$ and nodes $d_{n-1}, d_n \in \mathrm{dom}(\mathfrak{A})$ 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 $(d_n, d_{n+1}) \in R^{\mathfrak{A}}_n$.

Lemma 2 *The following conditions are equivalent, for any $\mathcal{K}$-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 of length not exceeding $2^{||\mathcal{O}||} + 2$ witnessing $\mathcal{ALCI}$-incompleteness of t.*

It is decidable in ExpTime *whether a $\mathcal{K}$-type t is $\mathcal{ALCI}$-complete.*

Proof. "$1 \Rightarrow 2$". Let $\Sigma = \mathsf{sig}(\mathcal{K})$. Consider the tree-shaped model $\mathfrak{A}_t$ of $\mathcal{O}$ whose root c realizes t such that if a node $e \in \mathrm{dom}(\mathfrak{A}_t)$ realizes any $\mathcal{K}$-type t_1 and is of depth $k \geq 0$, then for every $\mathcal{K}$-type t_2 with $t_1 \rightsquigarrow_R t_2$ for some Σ-role R there exists e' realizing t_2 of depth $k+1$ with $(e, e') \in R^{\mathfrak{A}_t}$. 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 $\mathfrak{A}_t, c \not\sim_{\mathcal{ALCI},\Sigma} \mathfrak{A}_{t'}, c'$. But then there is a sequence σ of the form (1), possibly with $n = 0$, witnessing $\mathcal{ALCI}$-incompleteness of t that is realized in $\mathfrak{A}_t$ starting from c. To obtain a sequence σ with $n \geq 1$ assume that there exist a role R, a $\mathcal{K}$-type t' and a node $d \in \mathrm{dom}(\mathfrak{A}_t)$ 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 $n \geq 1$.) Now observe that $\exists R.\top \in t$. Thus there exists d' realizing a $\mathcal{K}$-type t'' in $\mathfrak{A}'$ such that $(d, d') \in R^{\mathfrak{A}'}$. Then

$$tRt''R^- tRt'$$

is as required.

"$2 \Rightarrow 3$". A straightforward pumping argument.

"$3 \Rightarrow 1$". Straightforward from the definition.

To show that it is in ExpTime to decide whether a $\mathcal{K}$-type t is $\mathcal{ALCI}$-complete, observe that one can construct a structure $\mathfrak{A}$ whose domain consists of all $\mathcal{K}$-types t and such that $t \in A^{\mathfrak{A}}$ if $A \in t$ and $(t_1, t_2) \in R^{\mathfrak{A}}$ if $t_1 \rightsquigarrow_R t_2$. 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 exponential time. ❑

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

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

Proof. Let $\Sigma = \mathsf{sig}(\mathcal{K})$.

"$\Rightarrow$". Assume $(\mathcal{K}, P, \{b\})$ is $\mathcal{ALCI}$-separable. By Theorem 21, there exists a forest model $\mathfrak{A}$ of $\mathcal{K}$ of finite outdegree such that for all $a \in P$: $\mathcal{D}_{\mathrm{con}(a)}, a \not\preceq_\Sigma \mathfrak{A}, b^{\mathfrak{A}}$. To show that Condition 1 holds, assume that there exists $a \in P$ and a homomorphism h from $\mathcal{D}_{\mathrm{con}(a)}$ to $\mathfrak{A}$ mapping a to $b^{\mathfrak{A}}$. As h is clearly an $\mathcal{ALCI}(\Sigma)$-embedding, we have derived a contradiction. To show that Condition 2 holds, assume that $\mathrm{tp}_{\mathcal{K}}(\mathfrak{A}, b^{\mathfrak{A}})$ is $\mathcal{ALCI}$-complete and that $\mathrm{tp}_{\mathcal{K}}(\mathfrak{A}, b^{\mathfrak{A}})$ is realized at $a^{\mathfrak{B}}$ in a model $\mathfrak{B}$ of $\mathcal{K}$. By definition, $\mathfrak{B}, a^{\mathfrak{B}} \sim_{\mathcal{ALCI},\Sigma} \mathfrak{A}, b^{\mathfrak{A}}$. But then the restriction of the bisimulation witnessing this to $\mathcal{D}_{\mathrm{con}(a)}$ is an $\mathcal{ALCI}(\Sigma)$-embedding between $\mathcal{D}_{\mathrm{con}(a)}, a$ and $\mathfrak{A}, b^{\mathfrak{A}}$ and we have derived a contradiction.

"$\Leftarrow$". 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 $\mathrm{tp}_{\mathcal{K}}(\mathfrak{A}, b^{\mathfrak{A}})$ is connected and $\mathcal{ALCI}$-complete, then by Condition 2 $\neg(\bigsqcap_{C \in \mathrm{tp}_{\mathcal{K}}(\mathfrak{A}, b^{\mathfrak{A}})} C)$ separates $(\mathcal{K}, P, \{b\})$ and we are done. If $\mathrm{tp}_{\mathcal{K}}(\mathfrak{A}, b^{\mathfrak{A}})$ is not connected, then it follows from $\mathcal{D}_{\mathrm{con}(a)}, a \not\to \mathfrak{A}, b^{\mathfrak{A}}$ 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 $\mathrm{tp}_{\mathcal{K}}(\mathfrak{A}, b^{\mathfrak{A}})$ is not realizable in $\mathcal{K}, a$. Thus $\neg(\bigsqcap_{C \in \mathrm{tp}_{\mathcal{K}}(\mathfrak{A}, b^{\mathfrak{A}})} C)$ separates $(\mathcal{K}, P, \{b\})$ and we are done.

Assume now that $\mathrm{tp}_{\mathcal{K}}(\mathfrak{A}, b^{\mathfrak{A}})$ is connected and not $\mathcal{ALCI}$-complete. For a model $\mathfrak{C}$ of $\mathcal{K}$ and $\ell \geq 0$ we denote by $\mathfrak{C}^{\leq \ell}_{\mathcal{D},b}$ the substructure of $\mathfrak{C}$ induced by all nodes reachable from

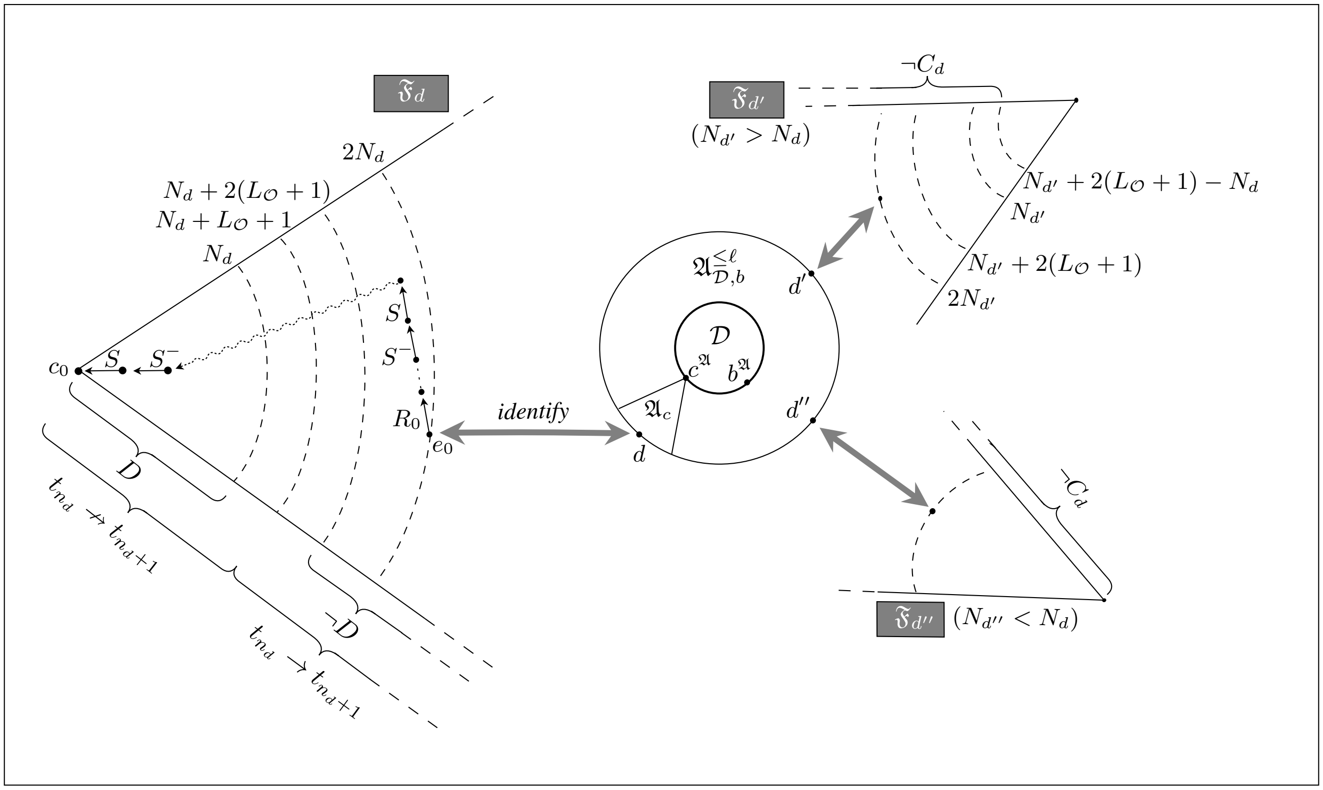


Figure 4: Construction of $\mathfrak{C}$.

some $c^\mathfrak{C}$, $c \in \text{dom}(\mathcal{D}_{\text{con}(b)})$, in at most ℓ steps. We construct for any $\ell \geq 0$ a model $\mathfrak{C}$ of $\mathcal{K}$ such that

(a) $\mathfrak{C}^{\leq \ell}_{\mathcal{D},b}, b^\mathfrak{C} \to \mathfrak{A}, b^\mathfrak{A}$;

(b) for any two distinct $d_1, d_2 \in \text{dom}(\mathfrak{C}^{\leq \ell}_{\mathcal{D},b})$: $\mathfrak{C}, d_1 \not\sim_{\mathcal{ALCI},\Sigma} \mathfrak{C}, d_2$.

We first show that the theorem is proved if such a $\mathfrak{C}$ can be constructed.

Claim. If (a) and (b) hold for $\ell \geq |\mathcal{D}|$ and $\mathcal{D}_{\text{con}(a)}, a \not\to \mathfrak{A}, b^\mathfrak{A}$ for $a \in P$, then $\mathcal{D}_{\text{con}(a)}, a \not\preceq_\Sigma \mathfrak{C}, b^\mathfrak{A}$.

By the claim and Theorem 21, $(\mathcal{K}, P, \{b\})$ is $\mathcal{ALCI}$-separable, as required. To prove the claim, let $\ell \geq |\mathcal{D}|$. Assume that there exists an $\mathcal{ALCI}(\Sigma)$ embedding S between $\mathcal{D}_{\text{con}(a)}, a$ and $\mathfrak{C}, b^\mathfrak{C}$ for some $a \in P$. As there is no homomorphism from $\mathcal{D}_{\text{con}(a)}$ to $\mathfrak{A}$ mapping a to $b^\mathfrak{A}$, by Condition (a) there is no homomorphism from $\mathcal{D}_{\text{con}(a)}$ to $\mathfrak{C}^{\leq \ell}_{\mathcal{D},b}$ 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 $\text{dist}(b^\mathfrak{C}, d), \text{dist}(b^\mathfrak{C}, d') \leq |\mathcal{D}_{\text{con}(a)}|$. Then $\mathfrak{C}, d \sim_{\mathcal{ALCI},\Sigma} \mathfrak{C}, d'$ and we have derived a contradiction to Condition (b) for $\mathfrak{C}$.

We come to the construction of $\mathfrak{C}$. It is illustrated in Figure 4. Take a sequence $\sigma = t^\sigma_0 R^\sigma_0 \ldots R^\sigma_{m_\sigma} t^\sigma_{m_\sigma+1}$ that witnesses $\mathcal{ALCI}$-incompleteness of $t^\sigma_0 := \text{tp}_\mathcal{K}(\mathfrak{A}, b^\mathfrak{A})$, where $1 \leq m_\sigma \leq L_\mathcal{O} := 2^{||\mathcal{O}||} + 1$. Note that there exists $d \in \text{dom}(\mathfrak{A})$ such that $(b^\mathfrak{A}, d) \in (R^\sigma_0)^\mathfrak{A}$, since $\exists R^\sigma_0.\top \in t^\sigma_0$.

By unfolding $\mathfrak{A}$ at all $c^\mathfrak{A}$, $c \in \text{dom}(\mathcal{D}_{\text{con}(b)})$, 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 $R \in \text{sig}(\mathcal{K})$.

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 $\mathfrak{A}_c$, $c \in \text{dom}(\mathcal{D}_{\text{con}(b)})$. We obtain $\mathfrak{C}$ by keeping only $\mathfrak{A}^{\leq \ell}_{\mathcal{D},b}$ and then attaching to every $d \in L$ a tree-shaped model $\mathfrak{F}_d$ such that in the resulting model no node in L is Σ-bisimilar to any other node in $\mathfrak{A}^{\leq \ell}_{\mathcal{D},b}$. It then directly follows that $\mathfrak{C}$ satisfies Conditions (a) and (b).

We set

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

and let $S = R^\sigma_{m_\sigma - 1}, T = R^\sigma_{m_\sigma}$. Take for any $d \in L$ a number $N_d > |\mathcal{D}| + 2\ell + 2(L_\mathcal{O} + 1)$ such that $|N_d - N_{d'}| > 2(L_\mathcal{O} + 1)$ for $d \neq d'$. Now fix $d \in L$ and let $t_0 = \text{tp}_\mathcal{K}(\mathfrak{A}, d)$. 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 $t^\sigma_{m_\sigma - 1} S t^\sigma_{m_\sigma} T t^\sigma_{m_\sigma+1}$. By Lemma 2 we may assume that $n_d \leq L_\mathcal{O}$. Let

$$D = \exists \Sigma^{L_\mathcal{O}}.(C_1 \sqcap \neg \exists T.C_2),$$

where $\exists \Sigma^k.C$ stands for the disjunction of all $\exists \rho.C$ with ρ a path $R_1 \cdots R_m$ of Σ roles $R_1, \ldots, R_m$ and $m \leq k$.

To construct $\mathfrak{A}_d$ consider the tree-shaped model $\mathfrak{A}_{c_0}$ of $\mathcal{O}$ whose root c_0 realizes t_{n_d} such that if a node $e \in \text{dom}(\mathfrak{A}_{c_0})$ realizes any $\mathcal{K}$-type t and is of depth $k \geq 0$, then for every $\mathcal{K}$-type t' with $t \rightsquigarrow_R t'$ for some Σ-role R there exists e' realizing t' of depth $k+1$ with $(e, e') \in R^{\mathfrak{A}_{c_0}}$, except if $k \leq N_d + L_{\mathcal{O}} + 1$, $t = t_{n_d}$, $R = T$, and $t' = t_{n_d+1}$. Observe that $\mathfrak{A}_{c_0}$ satisfies

- $e \in D^{\mathfrak{A}_{c_0}}$ for all e with $\text{dist}_{\mathfrak{A}_{c_0}}(c_0, e) \leq N_d$;
- $e \notin D^{\mathfrak{A}_{c_0}}$ for all e with $\text{dist}_{\mathfrak{A}_{c_0}}(c_0, e) > N_d + 2(L_{\mathcal{O}} + 1)$. 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;
- $(e_i, e_{i+1}) \in R_i^{\mathfrak{A}_{c_0}}$ for $i < n_d$;
- $(e_{n_d+2k+1}, e_{n_d+2k}), (e_{n_d+2k+1}, e_{n_d+2k+2}) \in S^{\mathfrak{A}_{c_0}}$ for $0 \leq k < N_d$;
- $e_{n_d+2k} \in C_1^{\mathfrak{A}_{c_0}}$, for all $k \leq N_d$;
- $e_{n_d+2k+1} \in C_0^{\mathfrak{A}_{c_0}}$, for all $k < N_d$.

Then $\mathfrak{F}_d$ is obtained from $\mathfrak{A}_{c_0}$ by renaming e_0 to d. Finally $\mathfrak{C}$ is obtained from $\mathfrak{A}^{\leq \ell}_{\mathcal{D},b}$ by hooking $\mathfrak{F}_d$ at d to $\mathfrak{A}^{\leq \ell}_{\mathcal{D},b}$ for all $d \in L$, see Figure 4. $\mathfrak{C}$ is a model of $\mathcal{K}$ since t_0 is realized in e_0. Moreover, clearly $\mathfrak{C}$ satisfies Condition (a). For Condition (b) assume $d \in L$ is as above. Let $C_d = \forall \Sigma^{N_d}.D$, where $\forall \Sigma^k.D$ stands for $\neg \exists \Sigma^k.\neg D$. Then $e_{n_d+2N_d} \in C_d^{\mathfrak{C}}$ and by construction $C_d^{\mathfrak{C}} \subseteq \text{dom}(\mathfrak{F}_d)$. Condition (b) 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 $\mathfrak{A}^{\leq \ell}_{\mathcal{D},b}$ to a node satisfying C_d. ❑

D Proofs for Section 5.2

D.1 Preliminaries for Guarded Bisimulations

We define guarded bisimulations, a standard tool for proving that two structures satisfy the same guarded formulas (Grädel and Otto 2014; Hernich et al. 2020).

Let $\mathfrak{A}$ be structure. It will be convenient to use the notation $[\vec{a}] = \{a_1, \ldots, a_n\}$ to denote the set of components of the tuple $\vec{a} = (a_1, \ldots, a_n) \in \text{dom}(\mathfrak{A})^n$. A set $G \subseteq \text{dom}(\mathfrak{A})$ is *guarded* in $\mathfrak{A}$ if G is a singleton or there exists R with $\mathfrak{A} \models R(\vec{a})$ such that $G = [\vec{a}]$. By $S(\mathfrak{A})$, we denote the set of all guarded sets in $\mathfrak{A}$. A tuple $\vec{a} \in \text{dom}(\mathfrak{A})^n$ is *guarded* in $\mathfrak{A}$ if $[\vec{a}]$ is a subset of some guarded set in $\mathfrak{A}$.

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

A set I of partial Σ-isomorphisms $p : \vec{a} \mapsto \vec{b}$ from guarded tuples $\vec{a}$ in $\mathfrak{A}$ to guarded tuples $\vec{b}$ in $\mathfrak{B}$ is called a *connected guarded* Σ*-bisimulation* if the following hold for all $p : \vec{a} \mapsto \vec{b} \in I$:

(i) for every guarded tuple $\vec{a}'$ in $\mathfrak{A}$ with $[\vec{a}] \cap [\vec{a}'] \neq \emptyset$ there exists a guarded tuple $\vec{b}'$ in $\mathfrak{B}$ and $p' : \vec{a}' \mapsto \vec{b}' \in I$ such that p' and p coincide on $[\vec{a}] \cap [\vec{a}']$.

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

Assume that $\vec{a}$ and $\vec{b}$ are (possibly not guarded) tuples in $\mathfrak{A}$ and $\mathfrak{B}$. Then we say that $\mathfrak{A}, \vec{a}$ and $\mathfrak{B}, \vec{b}$ are *connected guarded* Σ*-bisimilar*, in symbols $\mathfrak{A}, \vec{a} \sim_{\text{openGF},\Sigma} \mathfrak{B}, \vec{b}$, if there exists a partial Σ-isomorphism $p : \vec{a} \mapsto \vec{b}$ and a connected guarded Σ-bisimulation I such that the condition (i) and (ii) hold for p (Hernich et al. 2020).

Connected guarded Σ-bisimulations differ from the standard guarded Σ bismulations (Grädel and Otto 2014) in requiring $[\vec{a}] \cap [\vec{a}'] \neq \emptyset$ in Condition (i) and $[\vec{b}] \cap [\vec{b}'] \neq \emptyset$ in Condition (ii). If we drop these conditions then we talk about *guarded* Σ*-bisimulations* and *guarded* Σ*-bisimilarity*, in symbols $\mathfrak{A}, \vec{a} \sim_{\text{GF},\Sigma} \mathfrak{B}, \vec{b}$.

In the finitary versions of (connected) guarded bisimulations the Conditions (i) and (ii) are required to hold a finite number $\ell \geq 0$ of rounds only. Thus, one considers sets $I_\ell, \ldots, I_0$ of partial Σ-isomorphisms such that I_ℓ contains the partial Σ-isomorphism $p : \vec{a} \mapsto \vec{b}$ and for any $p \in I_i$ there exist $p' \in I_{i-1}$ such that (i) and, respectively, (ii) hold, for $0 < i \leq \ell$. If such sets exist then we say that $\mathfrak{A}, \vec{a}$ and $\mathfrak{B}, \vec{b}$ are *(connected) guarded* Σ ℓ*-bisimilar* and write $\mathfrak{A}, \vec{a} \sim^\ell_{\text{openGF},\Sigma} \mathfrak{B}, \vec{b}$ and $\mathfrak{A}, \vec{a} \sim^\ell_{\text{GF},\Sigma} \mathfrak{B}, \vec{b}$, respectively.

We say that $\mathfrak{A}, \vec{a}$ and $\mathfrak{B}, \vec{b}$ are *GF*(Σ)*-equivalent*, in symbols $\mathfrak{A}, \vec{a} \equiv_{\text{GF},\Sigma} \mathfrak{B}, \vec{b}$, if $\mathfrak{A} \models \varphi(\vec{a})$ iff $\mathfrak{A} \models \varphi(\vec{a})$ for all formulas $\varphi(\vec{x})$ in $\text{GF}(\Sigma)$. The *guarded quantifier rank* $\text{gr}(\varphi)$ of a formula φ in GF is the number of nestings of guarded quantifiers in it. We say that $\mathfrak{A}, \vec{a}$ and $\mathfrak{B}, \vec{b}$ are *GF*$^\ell(\Sigma)$*-equivalent*, in symbols $\mathfrak{A}, \vec{a} \equiv^\ell_{\text{GF},\Sigma} \mathfrak{B}, \vec{b}$, if $\mathfrak{A} \models \varphi(\vec{a})$ iff $\mathfrak{A} \models \varphi(\vec{a})$ for all formulas $\varphi(\vec{x})$ in $\text{GF}(\Sigma)$ of guarded quantifier rank at most ℓ. We say that $\mathfrak{A}, \vec{a}$ and $\mathfrak{B}, \vec{b}$ are *openGF*(Σ)*-equivalent*, in symbols $\mathfrak{A}, \vec{a} \equiv_{\text{openGF},\Sigma} \mathfrak{B}, \vec{b}$, if $\mathfrak{A} \models \varphi(\vec{a})$ iff $\mathfrak{A} \models \varphi(\vec{a})$ for all formulas $\varphi(\vec{x})$ in $\text{openGF}(\Sigma)$. We say that $\mathfrak{A}, \vec{a}$ and $\mathfrak{B}, \vec{b}$ are *openGF*$^\ell(\Sigma)$*-equivalent*, in symbols $\mathfrak{A}, \vec{a} \equiv^\ell_{\text{openGF},\Sigma} \mathfrak{B}, \vec{b}$, if $\mathfrak{A} \models \varphi(\vec{a})$ iff $\mathfrak{A} \models \varphi(\vec{a})$ for all formulas $\varphi(\vec{x})$ in $\text{openGF}(\Sigma)$ of guarded quantifier rank at most ℓ. The following is shown in (Grädel and Otto 2014; Hernich et al. 2020).

Lemma 3 *Let $\mathfrak{A}, \vec{a}$ and $\mathfrak{B}, \vec{b}$ be pointed structures and Σ a signature. Then for $\mathcal{L} \in \{GF, openGF\}$ and all $\ell \geq 0$:*

$$\mathfrak{A}, \vec{a} \equiv^\ell_{\mathcal{L},\Sigma} \mathfrak{B}, \vec{b} \quad \textit{iff} \quad \mathfrak{A}, \vec{a} \sim^\ell_{\mathcal{L},\Sigma} \mathfrak{B}, \vec{b}.$$

Moreover,

$$\mathfrak{A}, \vec{a} \sim_{\mathcal{L},\Sigma} \mathfrak{B}, \vec{b} \quad \textit{implies} \quad \mathfrak{A}, \vec{a} \equiv_{\mathcal{L},\Sigma} \mathfrak{B}, \vec{b}$$

and, conversely, if $\mathfrak{A}$ and $\mathfrak{B}$ are ω-saturated, then

$$\mathfrak{A}, \vec{a} \equiv_{\mathcal{L},\Sigma} \mathfrak{B}, \vec{b} \quad \textit{implies} \quad \mathfrak{A}, \vec{a} \sim_{\mathcal{L},\Sigma} \mathfrak{B}, \vec{b}$$

D.2 Preliminaries for Guarded Tree Decompositions

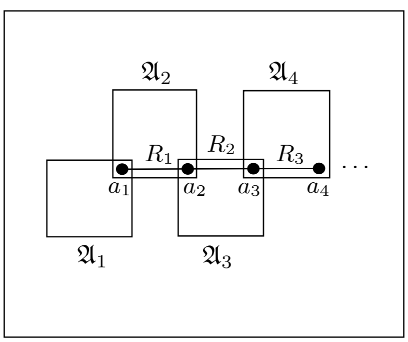
We introduce guarded tree decompositions as also used for example in (Grädel and Otto 2014; Hernich et al. 2020). A *guarded tree decomposition* of a structure $\mathfrak{A}$ is a triple (T, E, bag) with (T, E) an undirected tree and bag a function that assigns to every $t \in T$ a guarded set $\text{bag}(t)$ in $\mathfrak{A}$ such that

1. $\mathfrak{A} = \bigcup_{t\in T} \mathfrak{A}_{|\text{bag}(t)}$;
2. $\{t \in T \mid a \in \text{bag}(t)\}$ is connected in (T, E), for every $a \in \text{dom}(\mathfrak{A})$.

When convenient, we assume that (T, E) has a designated root r which allows us to view (T, E) as a directed tree. Also, it will be useful to sometimes allow $\text{bag}(r)$ not to be guarded. The difference between a classical tree decomposition and a guarded one is that in the latter, the elements in each bag must be a guarded set. While there is a classical tree decomposition of every structure, albeit of potentially high width (that is, maximum bag size), this is not the case for guarded tree decompositions. We say that $\mathfrak{A}$ is *guarded tree decomposable* if there exists a guarded tree decomposition of $\mathfrak{A}$. Observe that for every GF-ontology $\mathcal{O}$ and GF-formula $\varphi(\vec{x})$ such that $\mathcal{O} \not\models \varphi$ there exists a guarded tree-decomposable model $\mathfrak{A}$ of $\mathcal{O}$ such that $\mathfrak{A} \models \neg\varphi(\vec{a})$ for a tuple $\vec{a}$ with $[\vec{a}] \subseteq \text{bag}(r)$.

D.3 Partial Unfoldings

We introduce a new construction that allows us to transform paths into strict paths. The *partial unfolding* $\mathfrak{A}_{\vec{a}}$ of a structure $\mathfrak{A}$ along a tuple $\vec{a} = (a_1, \dots, a_n)$ in $\text{dom}(\mathfrak{A})$ such that $\text{dist}_{\mathfrak{A}}(a_i, a_{i+1}) = 1$ for all $i < n$ is defined as the following union of $n + 1$ copies of $\mathfrak{A}$. Denote the copies by $\mathfrak{A}_1, \mathfrak{A}_2, \dots, \mathfrak{A}_{n+1}$. The copies are mutually disjoint except that $\mathfrak{A}_i$ and $\mathfrak{A}_{i+1}$ share a copy of a_i. Formally, the domain of $\mathfrak{A}_i$ is $\mathfrak{A} \times \{i\}$ except that (a_{i-1}, i) is replaced by $(a_{i-1}, i - 1)$, for all $i > 1$. The constants are interpreted in $\mathfrak{A}_1$ as before and we often denote the elements $(a, 1)$ of $\mathfrak{A}_1$ simply by a. We following figure illustrates the construction.



We use the following properties of $\mathfrak{A}_{\vec{a}}$:

Lemma 4 *1. If $i < j$, then any path in $\mathfrak{A}_{\vec{a}}$ from an element of dom$(\mathfrak{A}_i)$ to an element of dom$(\mathfrak{A}_j)$ contains (a_k, k) for all $k \in \{i, \dots, j-1\}$;*

2. Let I contain for all i with $1 \leq i \leq n+1$ 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 $\mathfrak{A}_{\vec{a}}$.

3. If $\mathfrak{A}$ is a model of $\mathcal{K}$, then $\mathfrak{A}_{\vec{a}}$ is a model of $\mathcal{K}$.

4. The mapping h from $\mathfrak{A}_{\vec{a}}$ to $\mathfrak{A}$ defined by setting $h(b, i) = b$ is a homomorphism from $\mathfrak{A}_{\vec{a}}$ to $\mathfrak{A}$.

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

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

D.4 Proof for Example 6

Proposition 2 *Let $(\mathcal{K}, P, \{b\})$ be as in Example 6. Then any openGF-formula separating $(\mathcal{K}, P, N)$ has guarded quantifier rank at least n.*

Proof. Let $\Sigma = \{A, E, R\} = \text{sig}(\mathcal{K})$. To prove that no openGF-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 $\mathfrak{A}, b_0^{\mathfrak{A}} \sim^m_{\text{openGF},\Sigma} \mathfrak{B}, a_0^{\mathfrak{B}}$. 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 structure (Hernich et al. 2020) and $\mathfrak{A}$, modified by

- interpreting $a_i^{\mathfrak{B}}$, $0 \leq i \leq n$, 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 $E^{\mathfrak{B}}$;
- setting $b_i^{\mathfrak{B}} := b_i^{\mathfrak{A}}$, for $0 \leq i \leq n$.

It is straightforward to check that $\mathfrak{B}$ is a model of $\mathcal{K}$ and that $\mathfrak{B}, a_0^{\mathfrak{B}} \sim^m_{\text{openGF},\Sigma} \mathfrak{A}, b_0^{\mathfrak{A}}$, for all $m < n$. ❑

D.5 Guarded Embeddings

We introduce guarded embeddings as a tool to prove Theorems 6 and 7. Let $\mathcal{D}, \vec{a}$ be a pointed database, $\mathfrak{A}, \vec{b}$ a pointed structure, $\ell \geq 0$, and $\Sigma \supseteq \text{sig}(\mathcal{D})$ a signature. A *partial embedding* is an injective partial homomorphism. A pair (e, H) is a *guarded Σ ℓ-embedding between $\mathcal{D}, \vec{a}$ and $\mathfrak{A}, \vec{b}$* 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 $h_0 : e(\vec{a}) \mapsto \vec{b}$ and a partial embedding h from any guarded set in $\mathcal{D}'$ to $\mathfrak{A}$ such that the following condition hold:

- if $h_i : \vec{a}_i \mapsto \vec{b}_i \in H$ for $i = 1, 2$, then there exists a partial isomorphism $p : h_1([\vec{a}_1] \cap [\vec{a}_2]) \mapsto h_2([\vec{a}_1] \cap [\vec{a}_2])$ such that $p \circ h_1$ and h_2 coincide on $[\vec{a}_1] \cap [\vec{a}_2]$ and for any $\vec{c}$ with $[\vec{c}] = h_1([\vec{a}_1] \cap [\vec{a}_2]), \mathfrak{A}, \vec{c} \sim^\ell_{\text{openGF},\Sigma} \mathfrak{A}, p(\vec{c})$.

We write $\mathcal{D}, \vec{a} \preceq^\ell_{\text{openGF},\Sigma} \mathfrak{A}, \vec{b}^{\mathfrak{A}}$ if there exists a guarded Σ ℓ-embedding H between $\mathcal{D}, \vec{a}$ and $\mathfrak{A}, \vec{b}$.

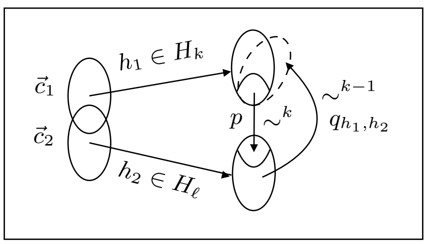
The following lemma shows that guarded Σ ℓ-embeddings determine a sequence $H_\ell, \dots, H_0$ of partial embeddings satisfying the (forth) condition of guarded Σ ℓ-bisimulations.

Lemma 5 *Let $(\mathcal{D}, \vec{a})$ be a pointed database and $(\mathfrak{A}, \vec{b}^{\mathfrak{A}})$ be a pointed model such that $(\mathcal{D}, \vec{a}) \preceq^{\ell}_{openGF,\Sigma} (\mathfrak{A}, \vec{b}^{\mathfrak{A}})$. 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 $k \leq \ell$, all $h \in H_k$ and all guarded sets $\vec{c}$ in $\mathcal{D}'$ such that $[\vec{c}] \cap dom(h) \neq \emptyset$, there exists $h' \in H_{k-1}$ with domain $[\vec{c}]$ such that h' coincides with h on $[\vec{c}] \cap dom(h)$.*
2. *for all $k_1, k_2 \leq \ell$, all $h_1 \in H_{k_1}, h_2 \in H_{k_2}$, and all tuples $\vec{c}_1, \vec{c}_2$ in $\mathcal{D}'$ such that $[\vec{c}_i] = dom(h_i)$, we have $h_1(\vec{c}) \sim^{\min(k_1,k_2)}_{openGF,\Sigma} h_2(\vec{c})$ for all $\vec{c}$ such that $[\vec{c}] = [\vec{c}_1] \cap [\vec{c}_2]$.*

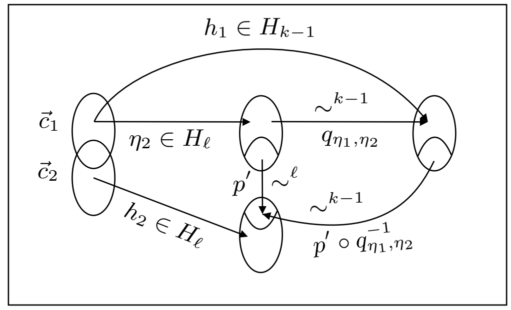
Proof. Let H be the set of partial embeddings witnessing $(\mathcal{D}, \vec{a}) \preceq^{\ell}_{\text{openGF},\Sigma} (\mathfrak{A}, \vec{b}^{\mathfrak{A}})$. Define $H_\ell := H$. We define H_k for $k < \ell$ by induction. Suppose H_k has been defined. We define H_{k-1}. We assume that for all $h_1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

(*) for any tuple $\vec{c}$ in $\mathcal{D}'$ such that $[\vec{c}] = [\vec{c}_1] \cap [\vec{c}_2]$, there is a partial isomorphism $p : h_1(\vec{c}) \mapsto h_2(\vec{c})$ witnessing $h_1(\vec{c}) \sim^{k}_{\text{openGF},\Sigma} h_2(\vec{c})$



Now assume that h_1, h_2 satisfying the conditions above are given. As $[h_2(\vec{c}_2)]$ is guarded (by $\vec{c}_2$ being guarded and h a partial homomorphism) and intersects The $h_2[[\vec{c}_1] \cap [\vec{c}_2]]$, and as p witnesses a openGF Σ k-bisimulation, there exists a partial isomorphism q_{h_1,h_2} with domain $[h_2(\vec{c}_2)]$ witnessing $h_2(\vec{c}_2) \sim^{k-1}_{\text{openGF},\Sigma} q_{h_1,h_2}(h_2(\vec{c}_2))$ and that coincides with p^{-1} on $h_2[[\vec{c}_1] \cap [\vec{c}_2]]$. We then include $q_{h_1,h_2} \circ \circ h_2$ in H_{k-1}. This is well-defined, as the assumption (*) holds for all $k \leq \ell$:

- If $k = \ell$, then (*) is stated in the definition of Σ 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 $[\vec{c}_1], [\vec{c}_2]$ and $\vec{c}_2$ 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 Σ ℓ-guarded embeddings, as η_2 and h_2 are both in H_ℓ and have intersecting domains $[\vec{c}_1]$ and $[\vec{c}_2]$, there exists a partial isomorphism p' witnessing $\eta_2(\vec{c}) \sim^{\ell}_{\text{openGF},\Sigma} h_2(\vec{c})$ for any $\vec{c}$ such that $[\vec{c}] = [\vec{c}_1] \cap [\vec{c}_2]$. Then, by composition of bisimulations, $p := p'_{|\eta_2[\vec{c}]} \circ (q^{-1}_{\eta_1,\eta_2})_{|h_1[\vec{c}]}$ is a partial isomorphism witnessing $h_1(\vec{c}) \sim^{k-1}_{\text{openGF},\Sigma} h_2(\vec{c})$ i.e. (*) 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 $0 \leq k \leq \ell$ and $h_1 \in H_k$ with domain $[\vec{c}_1]$. Let $\vec{c}_2$ be guarded in $\mathcal{D}'$ such that $[\vec{c}_1] \cap [\vec{c}_2] \neq \emptyset$. 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 $[\vec{c}_2]$. 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 $[\vec{c}_1] \cap [\vec{c}_2]$, and q_{h_1,h_2} coincides with p^{-1} on $h_2[[\vec{c}_1] \cap [\vec{c}_2]]$, it follows that h' coincides with h_1 on $[\vec{c}_1] \cap [\vec{c}_2]$.
2. Let $h_1 \in H_{k_1}, h_2 \in H_{k_2}$ with intersecting domains $[\vec{c}_1], [\vec{c}_2]$. By definition of ℓ-guarded embeddings, there exists h'_2 in $H = H_\ell$ with domain $[\vec{c}_2]$. By (*), for every $\vec{c}$ in $\mathcal{D}'$ such that $[\vec{c}] = [\vec{c}_1] \cap [\vec{c}_2]$ we have $h_1(\vec{c}) \sim^{k_1}_{\text{openGF},\Sigma} h'_2(\vec{c})$ and $h_2(\vec{c}) \sim^{k_2}_{\text{openGF},\Sigma} h'_2(\vec{c})$, thus $h_1(\vec{c}) \sim^{\min(k_1,k_2)}_{\text{openGF},\Sigma} h_2(\vec{c})$ by composition of bisimulations.

❏

Observe that if $H_\ell, \dots, H_0$ satisfying the conditions of Lemma 5 exist, then $H_\ell \subseteq \cdots \subseteq H_0$: let $k \leq \ell$ and $\vec{c} \mapsto \vec{d} \in H_k$. By condition (1), since $[\vec{c}] \cap [\vec{c}] \neq \emptyset$ there exists $\vec{c} \mapsto \vec{d}' \in H_{k-1}$ that coincides with $\vec{c} \mapsto \vec{d}$ on $[\vec{c}]$, i.e. $\vec{c} \mapsto \vec{d} \in H_{k-1}$.

Theorem 22 *Let $(\mathcal{K}, P, \{\vec{b}\})$ be a labelled GF-KB and $\Sigma = sig(\mathcal{K})$. Then the following conditions are equivalent:*

1. *$(\mathcal{K}, P, \{\vec{b}\})$ is openGF-separable.*
2. *$(\mathcal{K}, P, \{\vec{b}\})$ is GF-separable.*
3. *there exists a (finite) model $\mathfrak{A}$ of $\mathcal{K}$ and $\ell \geq 0$ such that for all models $\mathfrak{B}$ of $\mathcal{K}$ and $\vec{a} \in P$: $\mathfrak{B}, \vec{a}^{\mathfrak{B}} \not\sim^{\ell}_{GF,\Sigma} \mathfrak{A}, \vec{b}^{\mathfrak{A}}$.*
4. *there exists a (finite) model $\mathfrak{A}$ of $\mathcal{K}$ and $\ell \geq 0$ such that for all $\vec{a} \in P$: $\mathcal{D}_{con(\vec{a})}, \vec{a} \not\preceq^{\ell}_{openGF,\Sigma} \mathfrak{A}, \vec{b}^{\mathfrak{A}}$.*
5. *there exists a (finite) model $\mathfrak{A}$ of $\mathcal{K}$ and $\ell \geq 0$ such that for all models $\mathfrak{B}$ of $\mathcal{K}$ and all $\vec{a} \in P$: $\mathfrak{B}, \vec{a}^{\mathfrak{B}} \not\sim^{\ell}_{openGF,\Sigma} \mathfrak{A}, \vec{b}^{\mathfrak{A}}$.*

Proof. The implications “1 ⇒ 2”, “2 ⇒ 3”, and “5 ⇒ 1” are straightforward. We prove “3 ⇒ 4” and “4 ⇒ 5”.

“3 ⇒ 4”. Take a model $\mathfrak{A}$ of $\mathcal{K}$ and $\ell \geq 0$ 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 $\vec{a}_0 \in P$ such that there exists a guarded Σ ℓ-embedding (e, H) from $\mathcal{D}_{\text{con}(\vec{a}_0)}, \vec{a}_0$ to $\mathfrak{A}, \vec{b}^{\mathfrak{A}}$. Assume $e : \mathcal{D}_{\text{con}(\vec{a}_0)} \mapsto \mathcal{D}'$ and that $e(\vec{a}_0) = \vec{a}'_0$. We construct a model $\mathfrak{B}$ as follows: first take a copy $\mathfrak{B}'$ of $\mathfrak{A}$. For the constants $c \in \text{dom}(\mathcal{D}) \setminus \text{dom}(\mathcal{D}_{\text{cons}(\vec{a}_0)})$, we define $c^{\mathfrak{B}'}$ as the copy of $c^{\mathfrak{A}}$ in $\mathfrak{B}'$. The interpretation of the constants in $\mathcal{D}_{\text{cons}(\vec{a}_0)}$ will be defined later. 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 $\vec{a}'_0 \mapsto \vec{b}^{\mathfrak{A}}$ removed if $\vec{a}_0$ is not guarded. Now let

$$\text{dom}(\mathfrak{B}'') = (H' \times \text{dom}(\mathfrak{A}))/_\sim,$$

where $\sim$ identifies all $(h, d), (h', d')$ such that $(h, d) = (h', d')$ or there exists $c \in \text{dom}(h) \cap \text{dom}(h')$ such that $h(c) = d$ and $h'(c) = d'$. Denote the equivalence class of (h, d) w.r.t. $\sim$ by $[h, d]$. For any constant c in $\mathcal{D}_{\text{con}(\vec{a}_0)}$, we set $c^{\mathfrak{B}''} = [h, h(e(c))]$, where $h \in H'$ is such that $e(c) \in \text{dom}(h)$. Observe that this is well defined as $(h', h'(e(c))) \sim (h, h(e(c)))$ for any $h' \in H'$ with $e(c) \in \text{dom}(h')$. We define the interpretation $R^{\mathfrak{B}''}$ of the relation symbol R by setting for $e_1, \ldots, e_n \in \text{dom}(\mathfrak{B}'')$, $\mathfrak{B}'' \models R(e_1, \ldots, e_n)$ if there exists $h \in H'$ and $c_1, \ldots, c_n \in \text{dom}(\mathfrak{A})$ such that $e_i = [h, c_i]$ and $\mathfrak{A} \models R(c_1, \ldots c_n)$. Then, the map

$$\begin{aligned} f_h : \ \text{dom}(\mathfrak{A}) &\to (H' \times \text{dom}(\mathfrak{A}))/_\sim \\ c &\mapsto [h, c] \end{aligned}$$

is an embedding from $\mathfrak{A}$ to $\mathfrak{B}''$, by definition.

We show that $\mathfrak{B}, \vec{a}_0^{\mathfrak{B}} \sim^{\ell}_{\text{GF},\Sigma} \mathfrak{A}, \vec{b}^{\mathfrak{A}}$. 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 Σ ℓ-bisimulation $\hat{H}_\ell, \ldots, \hat{H}_0$, let S_i be the set of $p : \vec{c} \mapsto \vec{d}$ witnessing that $\mathfrak{A}, \vec{c} \sim^{i}_{\text{openGF},\Sigma} \mathfrak{A}, \vec{d}$, where $\vec{c}$ is guarded. Then include in $\hat{H}_i$

- all $\vec{c}' \mapsto \vec{c}$, where $\vec{c}'$ is the copy in $\mathfrak{B}'$ of the guarded tuple $\vec{c}$ in $\mathfrak{A}$;
- all compositions $p \circ (f_h^{-1})_{|[\vec{d}]}$ for any guarded tuple $\vec{d}$ in the range of f_h and $p \in S_i$;

In addition, include in $\vec{a}_0^{\mathfrak{B}} \mapsto \vec{b}^{\mathfrak{A}}$ in all $\hat{H}_i$, $0 \leq i \leq \ell$. We show that $\hat{H}_\ell, \ldots, \hat{H}_0$ is a guarded Σ ℓ-bisimulation.

For any $i \leq \ell$ any $g \in \hat{H}_i$ is clearly a partial Σ-isomorphism, either trivially if $\text{dom}(g) \subseteq \mathfrak{B}'$ or by composition of partial Σ-isomorphisms if $\text{dom}(g) \subseteq \mathfrak{B}''$. By definition, $\hat{H}_\ell$ contains $\vec{a}_0^{\mathfrak{B}} \mapsto \vec{b}^{\mathfrak{A}}$. We thus only need to check the "Forth" and "Back" conditions for guarded ℓ-bisimulations. Let $g \in \hat{H}_k$ for some k with $1 \leq k \leq \ell$. By definition of $\hat{H}_k$, we have either $\text{dom}(g) \subseteq \mathfrak{B}'$ or $\text{dom}(g) \subseteq \mathfrak{B}''$. In each case, we show that for any guarded $\vec{c}$ in $\mathfrak{B}$ and guarded $\vec{d}$ in $\mathfrak{A}$, there exists $g'_0 \in \hat{H}_{k-1}$ with domain $[\vec{c}]$ that coincides with g on $[\vec{c}] \cap \text{dom}(g)$ (Forth) and there exists $g'_1 \in \hat{H}_{k-1}$ such that $\text{dom}((g'_1)^{-1}) = [\vec{d}]$ and $(g'_1)^{-1}$ coincides with g^{-1} on $[\vec{d}] \cap \text{im}(g)$ (Back).

First assume that $[\vec{c}] \cap \text{dom}(g) = \emptyset$. Then, as $\vec{c}$ is guarded in $\mathfrak{B}$, it is either included in $\mathfrak{B}'$ or included in $\mathfrak{B}''$. If $\vec{c}$ is included in $\mathfrak{B}'$, then the partial isomorphism mapping $\vec{c}$ to its copy in $\mathfrak{A}$ is in $\hat{H}_{k-1}$, as required. If $\vec{c}$ is in $\mathfrak{B}''$, then $\vec{c}$ can be written $([h, c_1], \ldots, [h, c_n])$ for some $h \in H'$ and $c_1, \ldots, c_n \in \text{dom}(\mathfrak{A})$ as it is guarded. But then $(f_h)^{-1}_{|[\vec{c}]} \in \hat{H}_{k-1}$ is as required.

The case $[\vec{d}] \cap \text{im}(g) = \emptyset$ is similar. Assume $\vec{d} = (d_1, \ldots, d_n)$ and let $[h, \vec{d}] := ([h, d_1], \ldots, [h, d_n]) \in \mathfrak{B}''$ for any $h \in H'$. Then $(f_h)^{-1}_{|[h,\vec{d}]} \in \hat{H}_{k-1}$ is as required, for any $h \in H'$. We now focus on proving (Forth) and (Back) assuming intersections are not empty.

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

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

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

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

(Forth) Write $\text{dom}(g)$ as $([h_1, c_1], \ldots, [h_n, c_n])$ with $h_1, \ldots, h_n \in H'$ and $(c_1, \ldots, c_n) =: \vec{c}$ a tuple in $\mathfrak{A}$. We want to prove that for any $([h'_1, c'_1], \ldots, [h'_m, c'_m])$ guarded in $\mathfrak{B}$ that intersects $\text{dom}(g)$ there exists $g' \in \hat{H}_{k-1}$ with domain $([h', c'_1], \ldots, [h', c'_m])$ that coincides with g on $\text{dom}(g) \cap \text{dom}(g')$. As $([h'_1, c'_1], \ldots, [h'_m, c'_m])$ is guarded in $\mathfrak{B}$ and intersects $\text{dom}(g)$ 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 $([h', c'_1], \ldots, [h', c'_m])$ for some $h' \in H'$, with $(c'_1, \ldots, c'_m)$ being guarded in $\mathfrak{A}$, again by construction of $\mathfrak{B}$. As for $([h_1, c_1], \ldots, [h_n, c_n])$, we can write it as $([h, c_1], \ldots, [h, c_n])$ for some $h \in H'$, either because it is guarded or because it is equal to $\vec{a}^{\mathfrak{B}}$, i.e. $([h, h(a_1)], \ldots, [h, h(a_n)])$ for some $h \in H'$. By definition of $\hat{H}_k$, we can write g as $p \circ (f_h^{-1})_{|\text{dom}(g)}$ for some $p \in S_k$, and we know g' has to be in the form $p' \circ (f_{h'}^{-1})_{|\text{dom}(g')}$ for some $p' \in S_{k-1}$. For notation purposes, we write $[h, \vec{c}] = ([h, c_1], \ldots, [h, c_n])$ and $[h', \vec{c}'] = ([h', c'_1], \ldots, [h', c'_m])$.

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

Case 2. $h \neq h'$. Figure 5 illustrates the following construction. For all $[h, c_i] \in [[h, \vec{c}]] \cap [[h', \vec{c}']]$ we have that $c_i = h(d_i)$ and $c'_i = h'(d_i)$ for some $d_i \in \text{dom}(h) \cap \text{dom}(h')$. For any tuple $\vec{d}$ in $\mathcal{D}'$ such that $[\vec{d}] = \text{dom}(h) \cap \text{dom}(h')$, we have $\mathfrak{A}, h'(\vec{d}) \sim^{\ell}_{\text{GF},\Sigma} \mathfrak{A}, h(\vec{d})$, witnessed by some partial isomorphism $q : [h'(\vec{d})] \to [h(\vec{d})]$. Also, via p, we have $\mathfrak{A}, \vec{d}' \sim^{k}_{\text{GF},\Sigma} \mathfrak{A}, p(\vec{d}')$ for any $\vec{d}'$ such that $[\vec{d}'] = [h(\vec{d})] \cap [\vec{c}]$. By composition, for any $\vec{d}''$ such that $[\vec{d}''] = [h'(\vec{d})] \cap [\vec{c}']$ we have $\mathfrak{A}, \vec{d}'' \sim^{k}_{\text{GF},\Sigma} \mathfrak{A}, p(q(\vec{d}''))$. Because $p \circ q$ is in S_k (by definition of S_k) and because $[\vec{c}']$ trivially intersects $[h'(\vec{d})] \cap [\vec{c}']$, there exists, by definition of guarded k-bisimulations, a partial isomorphism $p' \in S_{k-1}$ of domain $[\vec{c}']$ that coincides with $p \circ q$ on $[h'(\vec{d})] \cap [\vec{c}']$. Then, $g' := p' \circ (f^{-1}_{h'})_{|[h',\vec{c}']}$ is the desired partial isomorphism in $\hat{H}_{k-1}$.

(Back) The construction is illustrated in Figure 6. Let $\vec{d}$ be guarded in $\mathfrak{A}$ such that $[\vec{d}] \cap \text{im}(g)$. We show there exists $g' \in \hat{H}_{k-1}$ with image $[\vec{d}]$ such that $(g')^{-1}$ coincides with g^{-1} on $[\vec{d}] \cap \text{im}(g)$. By definition of $\hat{H}_k$ we can write $g = p \circ (f^{-1}_h)_{|\text{dom}(g)}$ for some $h \in H'$ and $p \in S_k$, and we know g' has to be of the form $p' \circ (f^{-1}_{h'})_{|[\vec{d}]}$ for some $h' \in H'$. By definition of guarded k-bisimulations there exists $p' \in S_{k-1}$ such that $\text{im}(p') = [\vec{d}]$ and p'^{-1} coincides with p^{-1} on $\text{im}(p) \cap \text{im}(p')$. Given that $\text{im}(g) = \text{im}(p)$, if we write $\vec{d} = (d_1, \ldots, d_n)$ and $[h, p'^{-1}(\vec{d})] := ([h, p'^{-1}(d_1)], \ldots, [h, p'^{-1}(d_n)]) \in \mathfrak{B}''$, then $g' = p' \circ (f_h)^{-1}_{|[h,p'^{-1}(\vec{d})]} \in \hat{H}_{k-1}$ is as required.

"4 ⇒ 5" For an indirect proof, suppose $I_{2\ell}, \ldots, I_0$ is a guarded Σ 2ℓ-bisimulation between $\mathfrak{B}, \vec{a}^{\mathfrak{B}}$ and $\mathfrak{A}, \vec{b}^{\mathfrak{A}}$ for a model $\mathfrak{B}$ of $\mathcal{K}$, where $\ell \geq |\mathcal{D}|$. We may assume that $I_{i+1} \subseteq I_i$ for $i < 2\ell$. Let $\mathcal{D}'$ be the restriction of $\mathfrak{B}$ to $\{c^{\mathfrak{B}} \mid c \in \text{cons}(\mathcal{D}_{\text{con}(\vec{a})})\}$, where we regard the elements $c^{\mathfrak{B}}$ as constants. Define $e : \mathcal{D}_{\text{con}(\vec{a})} \to \mathcal{D}'$ by setting $e(c) = c^{\mathfrak{B}}$. Let H contain $h_0 : \vec{a}^{\mathfrak{B}} \mapsto \vec{b}^{\mathfrak{A}}$ and, for every guarded tuple $\vec{d}$ in $\mathcal{D}'$ any $h : \vec{d} \mapsto \vec{c} \in I_\ell$. It is easy to show that (e, H) is a guarded Σ ℓ-embedding: assume that $h_i : \vec{c}_i \mapsto \vec{d}_i \in H$ for $i = 1, 2$. Let X_1, X_2 be the images of $[\vec{c}_1] \cap [\vec{c}_2]$ under h_i and $\vec{d}$ such that $[\vec{d}] = X_1$. Then we have $h_i : \vec{c}_i \mapsto \vec{d}_i \in I_\ell$, 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

$$\mathfrak{A}, \vec{d} \sim^{\ell}_{\text{openGF},\Sigma} \mathfrak{B}, h_1^{-1}(\vec{d}) \sim^{\ell}_{\text{openGF},\Sigma} \mathfrak{A}, h_2(h^{-1}(\vec{d})).$$

❏

D.6 Proof of Theorem 7

Let $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ be a GF-KB and $\Sigma = \text{sig}(\mathcal{K})$. We give a syntactic description of when a $\mathcal{K}$-type $\Phi(x)$ is openGF-complete. A *guarded $\mathcal{K}$-type* $\Phi(\vec{x})$ is a $\mathcal{K}$-type that contains an atom $R(\vec{x})$. Call $\mathcal{K}$-types $\Phi_1(\vec{x}_1)$ and $\Phi_2(\vec{x}_2)$ *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 $[\vec{x}_1] \cup [\vec{x}_2]$. For a $\mathcal{K}$-type $\Phi(\vec{x})$ and a subsequence $\vec{x}_I$ of $\vec{x}$ we denote by $\Phi_{|\vec{x}_I}$ the subset of Φ containing all formulas in Φ with free variables from $\vec{x}_I$. $\Phi_{|\vec{x}_I}$ is called the *the restriction of* Φ *of* $\vec{x}_I$. Observe that $\mathcal{K}$-types $\Phi_1(\vec{x}_1)$ and $\Phi_2(\vec{x}_2)$ are coherent iff their restrictions to $[\vec{x}_1] \cap [\vec{x}_2]$ are logically equivalent. Assume a $\mathcal{K}$-type $\Phi(x)$ is given. A sequence

$$\sigma = \Phi_0(\vec{x}_0), \ldots, \Phi_n(\vec{x}_n), \Phi_{n+1}(\vec{x}_{n+1})$$

witnesses openGF-incompleteness of Φ if Φ is the restriction of Φ_0 to x, $n \geq 0$, and all Φ_i, $0 \leq i \leq n+1$, are guarded $\mathcal{K}$-types each containing at $\neg(x = y)$ for some variables x, y (we say that the Φ_i are *non-unary*) such that $[\vec{x}_i] \cap [\vec{x}_{i+1}] \neq \emptyset$, all Φ_i, Φ_{i+1} are coherent, and there exists a model $\mathfrak{A}$ of $\mathcal{K}$ and a tuple $\vec{a}_n$ in $\mathfrak{A}$ such that $\mathfrak{A} \models (\Phi_n \wedge \neg \exists \vec{x}'_{n+1} \Phi_{n+1})(\vec{a}_n)$, where $\vec{x}'_{n+1}$ is the sequence $\vec{x}_{n+1}$ without $[\vec{x}_n] \cap [\vec{x}_{n+1}]$.

Lemma 6 *The following conditions are equivalent, for any $\mathcal{K}$-type $\Phi(x)$:*

1. *$\Phi(x)$ is not openGF-complete;*
2. *there is a sequence witnessing openGF-incompleteness of $\Phi(x)$;*
3. *there is a sequence of length not exceeding $2^{2^{||\mathcal{O}||}} + 2$ witnessing openGF incompleteness of $\Phi(x)$.*

It is decidable in 2ExpTime *whether a $\mathcal{K}$-type $\Phi(x)$ is openGF complete.*

Proof. Let $\Sigma = \text{sig}(\mathcal{K})$. It is straightforward to construct a guarded tree decomposable model $\mathfrak{A}$ of $\mathcal{O}$ with tree decompositon (T, E, bag) and root r such that $\Phi(x)$ is realized in $\text{bag}(r)$ by a and for every $\mathcal{K}$-type $\Psi_1(\vec{x})$ realized in some $\text{bag}(t)$ by $\vec{a}$ and every $\mathcal{K}$-type $\Psi_2(\vec{y})$ coherent with $\Psi_1(\vec{x})$ there exists a successor t' of t in T such that $\Psi_1(\vec{x}) \cup \Phi_2(\vec{y})$ is realized in $\text{bag}(t) \cup \text{bag}(t')$ in $\mathfrak{A}$ under an assignment μ of the variables $[\vec{x}] \cup [\vec{y}]$ such that $\mu(\vec{x}) = \vec{a}$. Thus, $\mathfrak{A}$ satisfies $\forall \vec{x}(\Psi_1 \to \exists \vec{y}' \Psi_2)$ for any coherent pair $\Psi_1(\vec{x}), \Psi_2(\vec{y})$, where $\vec{y}'$ is $\vec{y}$ without $[\vec{x}] \cap [\vec{y}]$.

"1 ⇒ 2". If $\Phi(x)$ is not openGF-complete, then there exists a guarded tree decomposable model $\mathfrak{A}'$ of $\mathcal{K}$ with root r which realizes $\Phi(x)$ in $\text{bag}(r)$ at a' such that $\mathfrak{A}, a \not\sim_{\text{openGF},\Sigma} \mathfrak{A}', a'$. But then $\mathfrak{A}, a$ realizes a sequence σ that witnesses openGF-incompleteness of $\Phi(x)$, except that possibly there exists already a guarded non-unary $\mathcal{K}$-type $\Phi_0(\vec{x}_0)$ which is realized in some $\vec{a}_0$ in $\mathfrak{A}$ with $a \in [\vec{a}_0]$ but there is no $\vec{a}'_0$ in $\mathfrak{A}'$ containing a' and realizing $\Phi_0(\vec{x}_0)$. Let $R_0(\vec{x}_0) \in \Phi_0$. Then, because we included the formulas $\exists \vec{x}_i(R(\vec{x}) \wedge x_i \neq x_j)$ in $\text{cl}(\mathcal{K})$, there exists a non-unary guarded $\mathcal{K}$-type $\Phi'(\vec{x}'_0)$ containing $R_0(\vec{x}'_0)$ such that there exists a tuple $\vec{a}'_0$ in $\mathfrak{A}'$ containing a' realizing Φ'. We obtain a sequence σ of any length by first taking $\Phi'(\vec{x}_0)$ an arbitrary number of times and then appending Φ_0.

"2 ⇒ 3". This can be proved by a straightforward pumping argument. This is particularly straightforward if one works with a sequence σ realized by a strict path. Consider a sequence

$$\sigma = \Phi_0(\vec{x}_0), \ldots, \Phi_n(\vec{x}_n), \Phi_{n+1}(\vec{x}_{n+1})$$

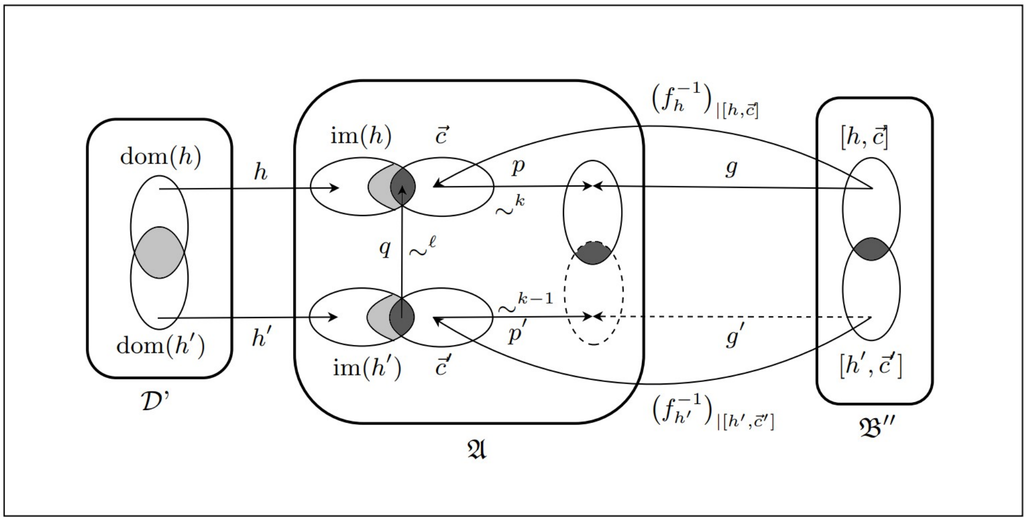


Figure 5: Illustration of proof of (Forth) condition for $\hat{H}_\ell, \ldots, \hat{H}_0$.

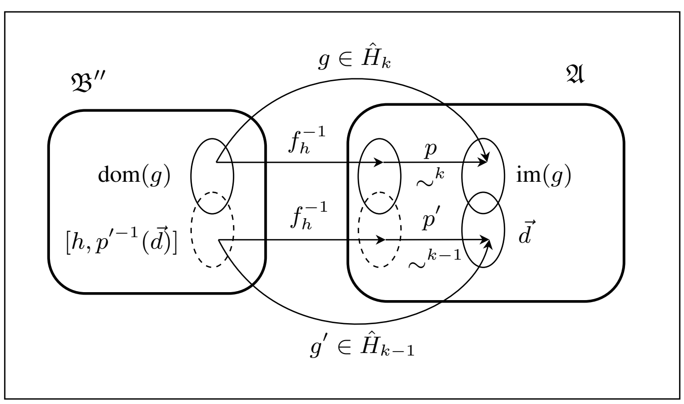


Figure 6: Illustration of proof of (Back) condition for $\hat{H}_\ell, \ldots, \hat{H}_0$.

that witnesses openGF-incompleteness of $\Phi(x)$ and a model $\mathfrak{A}$ of $\mathcal{K}$ satisfying $\mathfrak{A} \models (\Phi_n \wedge \neg\exists \vec{x}'_{n+1}\Phi_{n+1})(\vec{a}_n)$. 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 $R_0(\vec{a}_0), \ldots, R_n(\vec{a}_n)$ such that $\vec{a}_i$ realizes Φ_i and $\mathfrak{A} \models (\Phi_n \wedge \neg\exists \vec{x}'_{n+1}\Phi_{n+1})(\vec{a}_n)$. We now modify $\mathfrak{A}$ in such a way that we obtain a sequence witnessing openGF-incompletensss of $\Phi(x)$ which is realized by a strict path. Choose a sequence $\vec{a} = (a_1, \ldots, a_m)$ such that $a_1 = a$ for the node a in $\vec{a}_0$ realizing $\Phi(x)$, $a_i \neq a_{i+1}$ and $a_i, a_{i+1} \in [\vec{a}_j]$ for some $j \leq n$, for all $i < m$, and $a_m \in \vec{a}_n$. Clearly one can find such a sequence for some $m \leq 2n$. Then take the partial unfolding $\mathfrak{A}_{\vec{a}}$ of $\mathfrak{A}$ along $\vec{a}$. In $\mathfrak{A}_{\vec{a}}$ we find the required strict path Lemma4. Pumping on this path is straightforward.

"3 $\Rightarrow$ 1". Straightforward.

The 2ExpTime upper bound for deciding whether a $\mathcal{K}$-type is openGF-complete can now be proved similarly to the ExpTime upper bound for deciding whether a type defined by an $\mathcal{ALCI}$-KB is $\mathcal{ALCI}$-complete. ❑

Lemma 7 *A $\mathcal{K}$-type $\Phi(\vec{x})$ is openGF-complete iff all restrictions $\Phi(x)$ of Φ to some variable x in $\vec{x}$ are openGF-complete.*

Proof. Assume first that $\Phi(\vec{x})$ is not openGF-complete. One can show similarly to the proof of Lemma 6 that (i) or (ii) holds:

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

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

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

If (ii), then we are done by taking any variable x in x_I and the restriction $\Phi_{|x}$ of Φ to x. Then $\Phi_{|x}$ is not openGF-complete. Now assume that (i) holds. We are again done if I contains at most one element (we can simply take the type of $\mathrm{tp}_{\mathcal{K}}(\mathfrak{A}, a_I)$ then). Consider a relation R_0 with $R_0(\vec{x}_0) \in \Phi_0$. By the closure condition on $\mathcal{K}$-types, we have $\mathfrak{A}' \models \exists \vec{x}'_0 R_0(\vec{x}_0)(\vec{a}_I)$. Take an extension $\vec{a}_1$ of $\vec{a}_I$ such that $\mathfrak{A}' \models R_0(\vec{a}_1)$. Take any $a \in \vec{a}_I$, the unary $\mathcal{K}$-type $\Phi(x) = \mathrm{tp}_{\mathcal{K}}(\mathfrak{A}', a)$, and the $\mathcal{K}$-type $\Phi_1(\vec{x}_1) := \mathrm{tp}_{\mathcal{K}}(\mathfrak{A}', \vec{a}_1)$. Then the sequence Φ_1, Φ_0 shows that $\Phi(x)$ is not openGF-complete. ❑

Lemma 8 *Let $\mathcal{D}, \vec{a}$ be a pointed database, $\mathfrak{A}, \vec{b}$ be a pointed structure, and $\ell \geq |\mathcal{D}|$. If $\mathcal{D}_{con(\vec{a})}, \vec{a} \preceq^{\ell}_{openGF,\Sigma} \mathfrak{A}, \vec{b}$ and $\mathcal{D}_{con(\vec{a})}, \vec{a} \not\to \mathfrak{A}, \vec{b}$, then there exist d, d' with $d \neq d'$ in $\mathfrak{A}^{\leq |\mathcal{D}|}_{\vec{b}}$ such that $\mathfrak{A}, d \sim^{\ell - |\mathcal{D}|}_{openGF,\Sigma} \mathfrak{A}, d'$.*

Proof. Let $e, H_\ell, \ldots, H_0$ witness $\mathcal{D}_{con(\vec{a})}, \vec{a} \preceq^{\ell}_{\mathrm{openGF},\Sigma} \mathfrak{A}, \vec{b}$ as in Lemma 5. We define a sequence of mappings $S_0, \ldots, S_\ell$ with $S_k \subseteq H_{\ell-k}$ for $k \leq \ell$ as follows. Define $S_0 = \{e(\vec{a}) \mapsto \vec{b}\}$ 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 $\vec{c}$ intersecting $\mathrm{dom}(h)$ an $h' \in H_{\ell-k-1}$ with domain $[\vec{c}]$ that coincides with h on $[\vec{c}] \cap \mathrm{dom}(h)$ (this is possible by Condition 1 of Lemma 5) and add it to S_{k+1}. Define $\overline{h} = \bigcup(\bigcup_{k \leq |\mathcal{D}|} S_k)$. We can see $\overline{h}$ as a set of pairs from

$$\mathrm{dom}(\mathcal{D}'_{\mathrm{con}(e(\vec{a}))}) \times \mathrm{dom}(\mathfrak{A}),$$

which may or may not be functional. We know h is not a homomorphism from $\mathcal{D}'_{\mathrm{con}(e(\vec{a}))}, e(\vec{a})$ to $\mathfrak{A}, \vec{b}^{\mathfrak{A}}$ because otherwise $h \circ e$ would witness $\mathcal{D}_{\mathrm{con}(\vec{a})}, \vec{a} \to \mathfrak{A}, \vec{b}^{\mathfrak{A}}$. However,

* $\mathcal{D}' \models R(c_1, \ldots, c_n)$ implies $\mathfrak{A} \models R(d_1, \ldots, d_n)$ for every $(c_1, d_1), \ldots, (c_n, d_n) \in \overline{h}$ and every n-ary $R \in \Sigma$, since $\overline{h}$ is a union of partial homomorphisms
* for every $c \in \mathcal{D}'_{\mathrm{con}(e(\vec{a}))}$ there exists $h \in \bigcup_{k \leq \mathrm{dist}_{\mathcal{D}'}(c, e(\vec{a}))} S_k$ such that $c \in \mathrm{dom}(h)$, so $\overline{h}$ is defined on the entire underlying set of $\mathcal{D}'_{\mathrm{con}(e(\vec{a}))}$, as $\mathrm{dist}_{\mathcal{D}'}(c, e(\vec{a})) \leq |\mathcal{D}'|$ trivially and $|\mathcal{D}'| \leq |\mathcal{D}|$ as e is surjective
* $e(\vec{a}) \mapsto \vec{b}$ is included in $\overline{h}$

Therefore the only possible reason as to why $\overline{h}$ is not a homomorphism witnessing $\mathcal{D}_{\mathrm{con}(\vec{a})}, \vec{a} \to \mathfrak{A}, \vec{b}$ is that $\overline{h}$ is not functional, i.e. there exist $c \in \mathrm{dom}(\mathcal{D}'_{\mathrm{con}(e(\vec{a}))})$ and $d, d' \in \mathrm{dom}(\mathfrak{A})$ such that $d \neq d'$ and $(c, d), (c, d') \in \overline{h}$. As every h included in $\overline{h}$ is functional, that implies there exist $h, h' \in \bigcup_{k \leq |\mathcal{D}|} S_k$ such that $h(c) = d$ and $h'(c) = d'$. There exist $k_1, k_2 \geq \ell - |\mathcal{D}|$ such that $h \in H_{k_1}$ and $h' \in H_{k_2}$. By condition (2) of Lemma 5 we get $\mathfrak{A}, d \sim^{\min(k_1,k_2)}_{\mathrm{openGF},\Sigma} \mathfrak{A}, d'$, hence $\mathfrak{A}, d \sim^{\ell-|\mathcal{D}|}_{\mathrm{openGF},\Sigma} \mathfrak{A}, d'$. Finally, $d, d' \in \mathfrak{A}^{\leq |\mathcal{D}|}_{\vec{b}}$ follows from the fact that $\mathrm{dist}_{\mathfrak{A}}(h(c), \vec{b}) \leq k$ for any $c \in \mathrm{dom}(h)$ such that $h \in S_k$. This can be proved by induction on k. ❑

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

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

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

Proof. "$\Rightarrow$". Assume $(\mathcal{K}, P, \{\vec{b}\})$ is non-projectively GF-separable. Let $\Sigma = \mathrm{sig}(\mathcal{K})$. By Theorem 22, there exists a finite model $\mathfrak{A}$ of $\mathcal{K}$ and $\ell_0 \geq 0$ such that for all $\vec{a} \in P$: $\mathcal{D}_{\mathrm{con}(\vec{a})}, \vec{a} \not\preceq^{\ell_0}_{\mathrm{openGF},\Sigma} \mathfrak{A}, \vec{b}^{\mathfrak{A}}$. Assume $\vec{a} \in P$ is given. As $\mathcal{D}_{\mathrm{con}(\vec{a})}, \vec{a} \to \mathfrak{A}, \vec{b}^{\mathfrak{A}}$ implies $\mathcal{D}_{\mathrm{con}(\vec{a})}, \vec{a} \preceq^{\ell}_{\mathrm{openGF},\Sigma} \mathfrak{A}, \vec{b}^{\mathfrak{A}}$ for all $\ell \geq 0$, we obtain that Condition 1 holds. To show that Condition 2 holds for $\mathfrak{A}$ and $\vec{a}$, assume that I as defined in the theorem is not empty and that $\mathrm{tp}_{\mathcal{K}}(\mathfrak{A}, \vec{b}^{\mathfrak{A}})$ is realizable in $\mathcal{K}, \vec{a}$. Take a model $\mathfrak{B}$ witnessing this. Consider the maximal sets $I_1, \ldots, I_k \subseteq \{1, \ldots, n\}$ such that $\vec{b}^{\mathfrak{B}}_{I_j}$ is in a connected component $\mathfrak{B}_j$ of $\mathfrak{B}$. Then there exists at least one j such that $\mathrm{tp}_{\mathcal{K}}(\mathfrak{A}, \vec{b}^{\mathfrak{A}}_{I_j})$ is not openGF-complete or not connected: otherwise $\mathfrak{B}, \vec{a}^{\mathfrak{B}}_{I_j} \sim_{\mathrm{openGF},\Sigma} \mathfrak{A}, \vec{b}^{\mathfrak{A}}_{I_j}$ for all j and so $\mathcal{D}_{\mathrm{con}(\vec{a}_{I_j})}, \vec{a}_{I_j} \preceq^{\ell}_{\mathrm{openGF},\Sigma} \mathfrak{A}, \vec{b}^{\mathfrak{A}}_{I_j}$ for all $\ell \geq 0$, thus $\mathcal{D}_{\mathrm{con}(\vec{a})}, \vec{a} \preceq^{\ell}_{\mathrm{openGF},\Sigma} \mathfrak{A}, \vec{b}^{\mathfrak{A}}$, for all $\ell \geq 0$, a contradiction.

For any j such that $\mathrm{tp}_{\mathcal{K}}(\mathfrak{A}, \vec{b}^{\mathfrak{A}}_{I_j})$ is not openGF-complete we have by Lemma 7 that $\mathrm{tp}_{\mathcal{K}}(\mathfrak{A}, \vec{b}^{\mathfrak{A}}_{i})$ is not openGF-complete for any $i \in I_j$. Therefore $J \neq \emptyset$. Assume now for a proof by contradiction that $\mathcal{D}_{\mathrm{con}(\vec{a}_J)}, \vec{a}_J \to \mathfrak{A}, \vec{b}^{\mathfrak{A}}_J$. Then $\mathcal{D}_{\mathrm{con}(\vec{a}_J)}, \vec{a}_J \preceq^{\ell}_{\mathrm{openGF},\Sigma} \mathfrak{A}, \vec{b}^{\mathfrak{A}}_J$, for any $\ell \geq 0$. By Lemma 7, $\mathrm{tp}_{\mathcal{K}}(\mathfrak{A}, \vec{b}^{\mathfrak{A}}_I)$ is openGF-complete and so $\mathfrak{B}, \vec{a}^{\mathfrak{B}}_I \sim_{\mathrm{openGF},\Sigma} \mathfrak{A}, \vec{b}^{\mathfrak{A}}_I$, and therefore $\mathcal{D}_{\mathrm{con}(\vec{a}_I)}, \vec{a}_I \preceq^{\ell}_{\mathrm{openGF},\Sigma} \mathfrak{A}, \vec{b}^{\mathfrak{A}}_I$, for every $\ell \geq 0$. As $\mathcal{D}_{\mathrm{con}(\vec{a}_I)}$ and $\mathcal{D}_{\mathrm{con}(\vec{a}_J)}$ are disjoint, it follows that $\mathcal{D}_{\mathrm{con}(\vec{a})}, \vec{a} \preceq^{\ell}_{\mathrm{openGF},\Sigma} \mathfrak{A}, \vec{b}^{\mathfrak{A}}$ for any $\ell \geq 0$, and we have derived a contradiction.

"$\Leftarrow$". Assume Conditions 1 and 2 hold for some model $\mathfrak{A}$ of $\mathcal{K}$ for all $\vec{a} \in P$. 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 $\Phi_i(x) = \mathrm{tp}_{\mathcal{K}}(\mathfrak{A}, b^{\mathfrak{A}}_i)$ is not connected. If $X = \{1, \ldots, n\}$, then $\neg \bigwedge_{i \in X} \Phi_i(x_i)$ separates $(\mathcal{K}, P, \{\vec{b}\})$ and we are done. Otherwise, let $\mathfrak{A}_i$, $i \in X$, be the maximal connected components of $\mathfrak{A}$ containing the singleton $b^{\mathfrak{A}}_i$. Our aim is to show that $\mathcal{D}_{\mathrm{con}(\vec{a})}, \vec{a} \not\preceq^{\ell}_{\mathrm{openGF},\Sigma} \mathfrak{C}, \vec{b}^{\mathfrak{A}}$ for a variant $\mathfrak{C}$ of $\mathfrak{A}$ and for sufficiently large ℓ, where $\Sigma = \mathrm{sig}(\mathcal{K})$.

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 $\mathcal{K}$-types $\Phi(x)$ with one free variable x such that $(\Phi(x), \Psi(x)) \in E$ iff there exists a model $\mathfrak{A}$ of $\mathcal{K}$ and nodes a, b in $\mathrm{dom}(\mathfrak{A})$ 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 $\{b^{\mathfrak{A}}_i \mid i \in X\}$ such that:

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

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

$(*)$ if $\mathcal{D}_{\mathrm{con}(\vec{a}_{I_0})}, \vec{a}_{I_0} \not\to \mathfrak{A}, \vec{b}^{\mathfrak{A}}_{I_0}$, then there exists $\ell \geq 0$ such that $\mathcal{D}_{\mathrm{con}(\vec{a}_{I_0})}, \vec{a}_{I_0} \not\preceq^{\ell}_{\mathrm{openGF},\Sigma} \mathfrak{C}, \vec{b}^{\mathfrak{A}}_{I_0}$.

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

(a) $\mathfrak{C}^{\leq \ell}_{\mathcal{D}'}, \vec{b}^{\mathfrak{A}}_{I_0} \to \mathfrak{A}, \vec{b}^{\mathfrak{A}}_{I_0}$;

(b) for any two distinct $d_1, d_2 \in \mathrm{dom}(\mathfrak{C}^{\leq \ell}_{\mathcal{D}'})$, $\mathfrak{C}, d_1 \not\sim^{\ell'}_{\mathrm{openGF},\Sigma} \mathfrak{C}, d_2$.

We first show that $(*)$ follows. Assume ℓ' is such that (b) holds. Let $\ell'' = \ell' + |\mathcal{D}|$ and $\ell \geq |\mathcal{D}|$. Assume that $\mathcal{D}_{\mathrm{con}(\vec{a}_{I_0})}, \vec{a}_{I_0} \preceq^{\ell''}_{\mathrm{openGF},\Sigma} \mathfrak{C}, \vec{b}^{\mathfrak{A}}_{I_0}$ but $\mathcal{D}_{\mathrm{con}(\vec{a}_{I_0})}, \vec{a}_{I_0} \not\to \mathfrak{A}, \vec{b}^{\mathfrak{A}}_{I_0}$. By Condition (a),

$$\mathcal{D}_{\mathrm{con}(\vec{a}_{I_0})}, \vec{a}_{I_0} \not\to \mathfrak{C}^{\leq \ell''}_{\mathcal{D}'}, \vec{b}^{\mathfrak{C}}_{I_0}$$

By Lemma 8, there exist d, d' with $d \neq d'$ in $\mathfrak{C}^{\leq |\mathcal{D}|}_{\mathcal{D}'}$ such that $\mathfrak{C}, d \sim^{\ell'}_{\mathrm{openGF},\Sigma} \mathfrak{C}, d'$ and we have derived a contradiction to Condition (b).

Assume $\ell \geq 0$ is given. To construct $\mathfrak{C}$, tet $T_{\mathfrak{C}}$ be the set of $\mathcal{K}$-types $\Phi(x)$ that are E-equivalent to some $\mathcal{K}$-type realized in $\mathfrak{C}$. Observe that $T_{\mathfrak{C}}$ is an equivalence class for the relation E, by construction of $\mathfrak{C}$. No $\mathcal{K}$-type in $T_{\mathfrak{C}}$ is openGF-complete and, in fact, we find a sequence

$$\sigma = \Phi^{\sigma}_0, \Phi^{\sigma}_1, \Phi^{\sigma}_2$$

such that for any $\mathcal{K}$-type $\Phi(x) \in T_{\mathfrak{C}}$ we find a sequence witnessing openGF-incompleteness of $\Phi(x)$ that ends with σ. 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 $\mathfrak{C}$ so that

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

For the construction of $\mathfrak{B}$, let $\mathfrak{B}_0 = \mathfrak{A}$ and include all $d \in \mathrm{dom}(\mathfrak{A}^{\leq \ell}_{\mathcal{D}'})$ 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 $k = \mathrm{dist}_{\mathfrak{B}_i}(\mathcal{D}', d)$. Then let $\mathfrak{B}_{i+1}$ be the partial unfolding $(\mathfrak{B}_i)_{\vec{d}}$ of $\mathfrak{B}_i$ for the tuple $\vec{d} = (d, d', d, d', \ldots)$ of length $\ell - k$, and obtain F_{i+1} by removing d from F_i and adding all new nodes in $\mathrm{dom}((\mathfrak{B}_{i+1})^{\leq \ell}_{\mathcal{D}'})$. Clearly this construction terminates after finitely many steps and (path) holds, see Lemma 4.

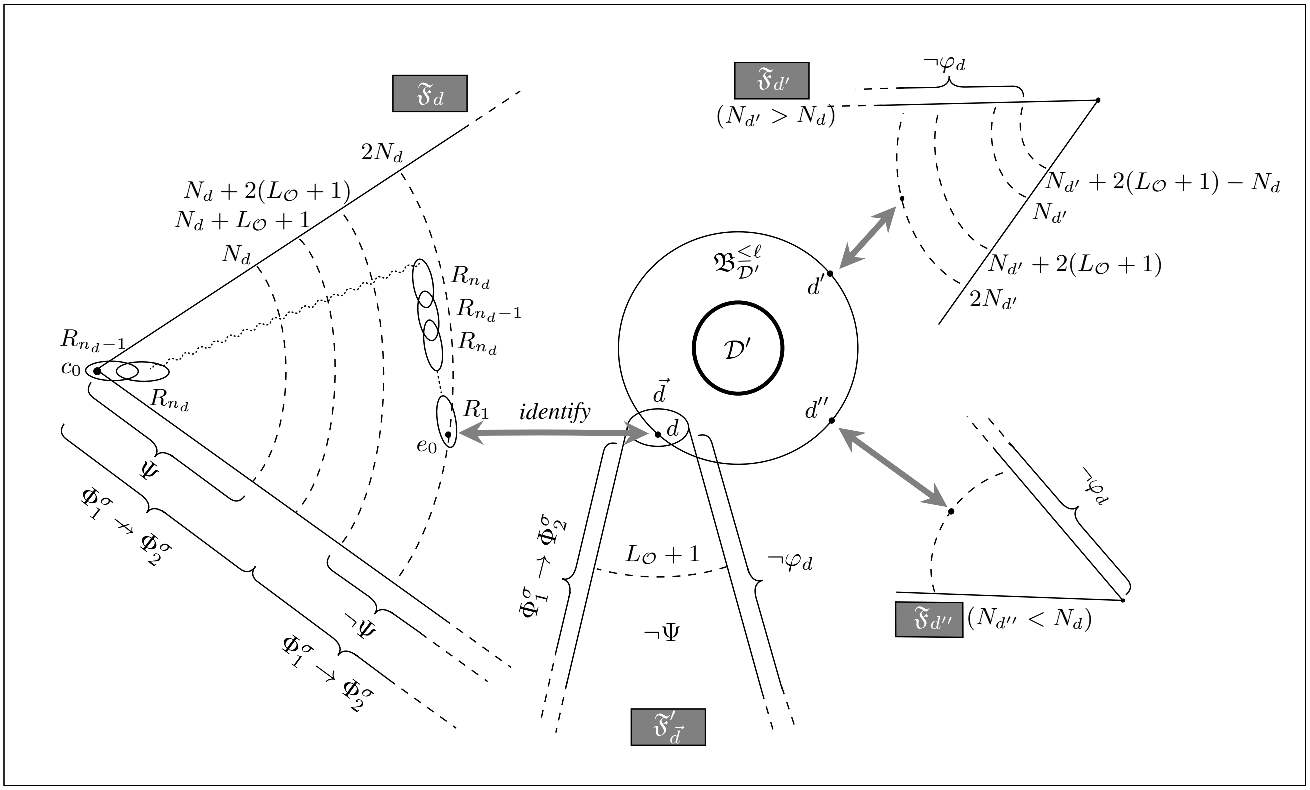


Figure 7: Construction of $\mathfrak{C}$.

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

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

$$\mathfrak{F}'_{\vec{d}} \models \forall \vec{x}_1 (\Phi_1^\sigma \to \exists \vec{x}' \Phi_2^\sigma)$$

where here and in what follows $\vec{x}_1$ are the variables in Φ_1^σ and $\vec{x}'$ are the variables in Φ_2 that are not in Φ_1.

The construction of $\mathfrak{F}_d$ is more involved. Let $L_\mathcal{O} = 2^{2^{||\mathcal{O}||}} + 1$ and take for any $d \in L$ a number

$$N_d > |\mathfrak{B}^{\leq \ell+1}_{\mathcal{D}'}| + 2(L_\mathcal{O} + 1)$$

such that $|N_d - N_{d'}| > 2(L_\mathcal{O} + 1)$ for $d \neq d'$. Fix $d \in L$ and let $\Phi_0(x) = \text{tp}_\mathcal{K}(\mathfrak{A}, d)$. Then $\Phi_0(x) \in T_\mathfrak{C}$ and we find a sequence $\Phi_0(\vec{x}_0), \ldots, \Phi_{n_d}(\vec{x}_{n_d}), \Phi_{n_d+1}(\vec{x}_{n_d+1})$ that witnesses openGF incompleteness of $\Phi_0(x_0)$ and ends with $\Phi_0^\sigma \Phi_1^\sigma \Phi_2^\sigma$. By Lemma 2 we may assume that $1 \leq n_d \leq L_\mathcal{O} + 1$. Let

$$\Psi(x) = \exists \Sigma^{L_\mathcal{O}+1}.(\Phi_1^\sigma \wedge \neg \exists \vec{x}' \Phi_2^\sigma),$$

where $\exists \Sigma^k.\chi$ stands for the disjunction of all openGF formulas stating that the 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 decomposible 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 $\mathcal{K}$-type $\Psi_1(\vec{x}_1)$ realized by some $\vec{c}$ with $[\vec{c}] = \text{bag}(t)$ and $\mathcal{K}$ type $\Psi_2(\vec{x}_2)$ coherent with $\Psi_1(\vec{x}_1)$ there exists a successor t' of t in T such that $\Psi_1(\vec{x}_1) \cup \Psi_2(\vec{x}_2)$ is realized in $\text{bag}(t) \cup \text{bag}(t')$ under an assignment μ of the variables $[\vec{x}_1] \cup [\vec{x}_2]$ such that $\mu(\vec{x}_1) = \vec{c}_1$, except if $\Psi_1 \wedge \neg \exists \vec{x}' \Psi_2$ (with $\vec{x}'$ the sequence of variables in $\vec{x}_2$ which are not in $\vec{x}_1$) is equivalent to $\Phi_1^\sigma \wedge \neg \exists \vec{x}' \Phi_2^\sigma$ and $\text{dist}_{\mathfrak{A}_r}(\text{bag}(t), \text{bag}(r)) \leq N_d + L_\mathcal{O} + 1$. Observe that

- $\mathfrak{A}_r \models \Psi(e)$ for all e with $\text{dist}_{\mathfrak{A}_r}(c_0, e) \leq N_d$;
- $\mathfrak{A}_r \models \neg\Psi(e)$ for all e with $\text{dist}_{\mathfrak{A}_r}(c_0, e) > N_d + 2(L_\mathcal{O} + 1)$.

Moreover, $\mathfrak{A}_r$ contains a strict path

$$R_1(\vec{e}_1), \ldots, R_{n_d}(\vec{e}_{n_d}), \ldots, R_{n_d}(\vec{e}_{n_d+2N_d})$$

from $e_0 \in [\vec{e}_1]$ to $c_0 \in [\vec{e}_{n_d+2N_d}]$ 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 $\mathfrak{B}^{\leq \ell}_{\mathcal{D}'}$ 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 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 $\mathfrak{B}^{\leq \ell}_{\mathcal{D}'}$ to a node satisfying φ_d.

We have proved $(*)$. We now aim to extend $(*)$ and show that $\mathcal{D}_{\text{con}(\vec{a})}, \vec{a} \not\preceq^{\ell}_{\text{openGF},\Sigma} \mathfrak{C}, \vec{b}^{\mathfrak{A}}$, for appropriately defined $\mathfrak{C}$, all $\vec{a} \in P$, and for sufficiently large ℓ.

Let $\vec{a} \in P$ be fixed. If $\mathcal{D}_{\text{con}(\vec{a}_{I_0})}, \vec{a}_{I_0} \not\to \mathfrak{A}, \vec{b}^{\mathfrak{A}}_{I_0}$ for some I_0 associated to some $\mathfrak{C} \in K$, then, by $(*)$, $\mathcal{D}_{\text{con}(\vec{a}_{I_0})}, \vec{a}_{I_0} \not\preceq^{\ell}_{\text{openGF},\Sigma} \mathfrak{C}, \vec{b}^{\mathfrak{C}}_{I_0}$, for some ℓ, and therefore $\mathcal{D}_{\text{con}(\vec{a})}, \vec{a} \not\preceq^{\ell}_{\text{openGF},\Sigma} \mathfrak{C}, \vec{b}^{\mathfrak{A}}$, for some ℓ, and we are done. Now assume that $\mathcal{D}_{\text{con}(\vec{a}_{I_0})}, \vec{a}_{I_0} \to \mathfrak{A}, \vec{b}^{\mathfrak{A}}_{I_0}$ for all I_0 associated with any $\mathfrak{C} \in K$. We know that $\mathcal{D}_{\text{con}(\vec{a})}, \vec{a} \not\to \mathfrak{A}, \vec{b}^{\mathfrak{A}}$. But then either (i) $\mathcal{D}_{\text{con}(a_i)}, a_i \not\preceq^{\ell}_{\text{openGF},\Sigma} \mathfrak{C}, b_i^{\mathfrak{C}}$ for some $i \in X$ (and we are done) or (ii) some a_i, a_j with $i \neq j$ and $i, j \in X$ are connected in $\mathcal{D}$, or (iii) some a_i, $i \in X$ and $a \in [\vec{a}_{I_0}]$ with I_0 linked to some $\mathfrak{C} \in K$ are connected in $\mathcal{D}$ or (iv) some $a \in [\vec{a}_{I_0}]$ and $a' \in [\vec{a}_{I'_0}]$ with I_0 and I'_0 linked to distinct $\mathfrak{C} \in K$ are connected in $\mathcal{D}$. In all these cases it follows that $\mathcal{D}_{\text{con}(\vec{a})}, \vec{a} \not\preceq^{\ell}_{\text{openGF},\Sigma} \mathfrak{C}, \vec{b}^{\mathfrak{A}}$, for sufficiently large ℓ. ❑

E Proofs for Section 5.3

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

Proof. We start with $\mathcal{L} = \text{FO}$ and show later how to generalize to FO^2. We reduce the infinite tiling problem which is, given a triple (T, V, H) with $V, H \subseteq T \times T$, determine whether there is a function $\tau : \mathbb{N} \times \mathbb{N} \to T$ such that, for all $i, j \geq 0$:

(i) $(\tau(i,j), \tau(i+1,j)) \in H$, and

(ii) $(\tau(i,j), \tau(i,j+1)) \in V$.

In this case, we say that (T, H, V) *admits a solution.* Given such a triple (T, H, V), we construct an FO^2-KB $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ as follows:

$$\begin{aligned}\mathcal{O} = \forall x\, (B(x) \to (\exists y R_v(x,y) \wedge B(y)) \wedge \\ (\exists y R_h(x,y) \wedge B(y))) \wedge \qquad (2)\end{aligned}$$

$$\forall xy\, (B(x) \wedge B(y) \to U(x,y)) \qquad (3)$$

$$\forall xy\, (\neg R_v(x,y) \to \overline{R}_v(x,y)) \wedge \qquad (4)$$

$$\forall x \bigvee_{t \in T} (A_t(x) \wedge \bigwedge_{t' \in T \setminus \{t\}} \neg A_{t'}(x)) \wedge \qquad (5)$$

$$\forall xy\, (R_v(x,y) \to \bigvee_{(t,t') \in V} A_t(x) \wedge A_{t'}(y)) \wedge \qquad (6)$$

$$\forall xy\, (R_h(x,y) \to \bigvee_{(t,t') \in H} A_t(x) \wedge A_{t'}(y)) \wedge \qquad (7)$$

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

Claim 1. $\varphi_{\mathcal{D}_{\text{con}(a)},a}$ separates $(\mathcal{K}, \{a\}, \{b\})$ iff (T, H, V) admits a solution.

Proof of Claim 1. By Theorem 1, it suffices to verify that $\mathcal{K} \not\models \varphi_{\mathcal{D}_{\text{con}(a)},a}(b)$ iff (T, H, V) admits a solution.

For $(\Rightarrow)$, let $\mathfrak{A}$ be a structure witnessing $\mathcal{K} \not\models \varphi_{\mathcal{D}_{\text{con}(a)},a}(b)$. Since $\mathfrak{A}$ is a model of formulas (2)–(4) and $\mathfrak{A} \not\models \varphi_{\mathcal{D}_{\text{con}(a)},a}(b)$, it contains an infinite grid formed by relations R_v and R_h. Since $\mathfrak{A}$ is a model of formula (5) 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 (6) and (7), the relations H and V are respected along R_h and R_v, respectively.

For $(\Leftarrow)$, we can easily read off a structure $\mathfrak{A}$ from a solution τ for (T, H, V). The domain of $\mathfrak{A}$ is $\mathbb{N} \times \mathbb{N}$. The binary relation symbols R_v and R_h are interpreted as the vertical and horizontal successor relations, respectively. The relation $\overline{R}_v$ is the complement of R_v, and U is the universal relation. Finally, every element $(i,j) \in \Delta^{\mathcal{I}}$ is labeled with B and (precisely) with $A_{\tau(i,j)}$. By construction of $\mathfrak{A}$ and since τ is a solution, we have $\mathfrak{A} \models \mathcal{K}$. However, $\mathfrak{A} \not\models \varphi_{\mathcal{D}_{\text{con}(a)},a}$ because $\mathfrak{A}$ was constructed from a grid. This finishes the proof of the Claim, and establishes undecidability of projective and non-projective (FO^2, FO)-separability.

For $(\text{FO}^2, \text{FO}^2)$-separability, we make a very similar reduction, but reduce from tiling problems that have a finite solution iff they have an infinite one (which are still undecidable). Moreover, we assume that T contains three fixed tiles t_r, t_u, t_{ru} which are *border tiles* corresponding to the right border, upper border, and upper right corner with the appropriate entries in H, V. The formulas in $\mathcal{O}$ are slightly changed so as to also allow finite models: points labeled with border tiles do not have the respective successors.

Claim 2. The following are equivalent:

1. $(\mathcal{K}, \{a\}, \{b\})$ is projectively or non-projectively FO-separable;
2. $(\mathcal{K}, \{a\}, \{b\})$ is projectively or non-projectively FO^2-separable;
3. (T, H, V) admits a solution.

Proof of Claim 2. (2) $\Rightarrow$ (1) is trivial.

(1) $\Leftrightarrow$ (3) is proven in the Claim above.

(3) $\Rightarrow$ (2) We show that $(\mathcal{K}, \{a\}, \{b\})$ is non-projectively FO^2-separable under the assumption that $\mathcal{K}$ mentions a binary relation symbol S. This is without loss of generality, as we can include $\forall xy\, S(x,y) \to S(x,y)$. If (T, H, V) admits a solution then it admits a finite one, say a solution $\tau : [n] \times [m] \to T$. Let π be a bijection from $[n] \times [m]$ to $[nm]$ and let C_{ij} be the $\mathcal{ALCI}$-concept (corresponding to an FO^2-formula) expressing that there is an S-path of length $\pi(i,j)$. We construct the following FO^2-formula $\varphi_{mn}(x)$, written as an $\mathcal{ALCI}$-concept:

$$\bigsqcup_{i,j} \exists U.(\forall R_v.\forall R_h.C_{ij} \to \exists R_h.\overline{R}_v.C_{ij}).$$

It should be clear that $\mathcal{K} \models \varphi_{mn}(a)$ since already $\mathcal{D} \models \varphi_{mn}(a)$. To see that $\mathcal{K} \not\models \varphi_{mn}(b)$, note that the finite solution τ viewed as a structure with points d_{ij} labeled with $A_{\tau(i,j)}$ and having outgoing S-paths of length $\pi(i,j)$ is a model of $\mathcal{K}$ and $\neg\varphi_{mn}(b)$. ❑

The fact that positive and negative examples in Claims 1 and 2 in the proof of Theorem 9 are singletons immediately yields undecidability of entity comparison by FO/FO^2-formulas over FO^2 KBs. Moreover, one can verify that Claims 1 and 2 remain valid if the set $\{b\}$ of negative examples is replaced with the set $N = \text{cons}(\mathcal{D}) \setminus \{a\}$. This shows that the undecidability applies to the problem GRE as well.

The construction from the proof of Theorem 9 can be used to prove that evaluating rooted CQs with a single answer variable is not finitely controllable for FO^2. One simply uses an infinite tiling problem that has a solution but that does not have a solution that can be 'realized' in a finite model of the ontology $\mathcal{O}$. It thus follows from the following proof that projective (FO^2, FO)-separability does not coincide with projective (FO^2, FO^2)-separability even for labeled KBs with a single positive example consisting of a single constant.

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

Proof. Assume that evaluating UCQs on $\mathcal{L}$-KBs is not finitely controllable, that is, there is an $\mathcal{L}$-KB $\mathcal{K} = (\mathcal{O}, \mathcal{D})$, a rooted UCQ $q(\vec{x}) = \bigvee_{i \in I} q_i(\vec{x})$, $\vec{x} = (x_1, \ldots, x_n)$, and a tuple $\vec{a}$ in $\mathcal{D}$ such that $\mathcal{K} \not\models q(\vec{a})$, but $\mathfrak{B} \models q(\vec{a})$ for all finite models $\mathfrak{B}$ of $(\mathcal{O}, \mathcal{D})$. Consider the relativization $\mathcal{O}_{|A}$ of the axioms of $\mathcal{O}$ to A and $\mathcal{D}^{+A} = \mathcal{D} \cup \{A(a) \mid a \in \text{dom}(\mathcal{D})\}$, for a fresh unary relation A.

Regard each query $q_i(x_1, \ldots, x_n)$ as a pointed database $\mathcal{D}_i, ([x_1], \ldots, [x_n])$ as follows: define an equivalence relation $\sim$ on the set of variables in q_i by setting $x \sim y$ if (x, y) is in the smallest equivalence relation containing all conjuncts $(x = y)$ of q_i. Then regard the equivalence classes $[x]$ as constants and set $R([y_1], \ldots, [y_m]) \in \mathcal{D}_i$ iff there are $y'_1 \in [y_1], \ldots, y'_m \in [y_m]$ such that $R(y'_1, \ldots, y'_m)$ is a conjunct of q_i. We assume the pointed databases $\mathcal{D}_i, ([x_1], \ldots, [x_n])$, $i \in I$, are mutually disjoint with the copy of $([x_1], \ldots, [x_n])$ in $\mathcal{D}_i$ denoted $([x_1]^i, \ldots, [x_n]^i)$. Let $\mathcal{D}' = \mathcal{D}^{+A} \cup \bigcup_{i \in I} \mathcal{D}_i$ and set

$$P = \{([x_1]^i, \ldots, [x_n]^i) \mid i \in I\}, \quad N = \{\vec{a}\}.$$

Consider the labeled knowledge base $(\mathcal{K}', P, N)$ for $\mathcal{K}' = (\mathcal{O}_{|A}, \mathcal{D}')$. Then the UCQ $q(\vec{x})$ separates $(\mathcal{K}', P, N)$: on the one hand, $\mathcal{K}' \models q_i([x_1]^i, \ldots, [x_n]^i)$ for all $i \in I$ since $\mathcal{D}_i \subseteq \mathcal{D}'$. Thus $\mathcal{K}' \models q([x_1]^i, \ldots, [x_n]^i)$ for all $([x_1]^i, \ldots, [x_n]^i) \in P$. On the other hand, $\mathfrak{A} \not\models q(\vec{a})$ for some model $\mathfrak{A}$ of $\mathcal{K}$ and, by relativization, such an $\mathfrak{A}$ can be expanded to a model of $\mathcal{K}'$ in which $q(\vec{a})$ is still not satisfied.

Suppose there is an $\mathcal{L}$-formula $\varphi(\vec{x})$ that separates $(\mathcal{K}', P, N)$. Since $\mathcal{L}$ has the FMP, there exists a finite model $\mathfrak{A}_f$ of $\mathcal{K}'$ such that $\mathfrak{A}_f \models \neg\varphi(\vec{a})$. As $\mathfrak{B} \models q(\vec{a})$ for all finite models $\mathfrak{B}$ of $(\mathcal{O}, \mathcal{D})$, there exists $i \in I$ with $\mathfrak{A}_f \models q_i(\vec{a})$. Then there is a homomorphism h from $\mathcal{D}_i, ([x_1]^i, \ldots, [x_n]^i)$ to $\mathfrak{A}_f, \vec{a}$ witnessing this. We modify $\mathfrak{A}_f$ to obtain a new structure $\mathfrak{A}'_f$ which coincides with $\mathfrak{A}_f$ except that the constants c in $\mathcal{D}_i$ are interpreted as $h(c)$. Then $\mathfrak{A}'_f$ is a model of $\mathcal{K}'$ with $\mathfrak{A}'_f \models \neg\varphi([x_1]^i, \ldots, [x_n]^i)$ which contradicts the assumption that $\varphi(\vec{x})$ separates $(\mathcal{K}', P, N)$. ❑

F Proofs for Section 7.1

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

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

Proof. Only "2 $\Rightarrow$ 3" is not trivial. Let $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ and assume that Point 3 does not hold, that is, there exist models $\mathfrak{A}$ and $\mathfrak{B}$ of $\mathcal{K}$ and $a \in P$, $b \in N$ such that $\text{tp}_{\mathcal{K}}(\mathfrak{A}, a^{\mathfrak{A}}) = \text{tp}_{\mathcal{K}}(\mathfrak{B}, b^{\mathfrak{B}})$. We prove that $(\mathcal{O}, \mathcal{D}_{a=b})$ is satisfiable. This implies that $(\mathcal{K}, P, N)$ is not strongly FO separable by Theorem 11.

By reinterpreting constants, we can achieve that $\mathfrak{B}$ is a model of the database $\mathcal{D}'$ from the definition of $\mathcal{D}_{a=b}$. Define the structure $\mathfrak{C}$ as $\mathfrak{A} \uplus \mathfrak{B}$ in which $a^{\mathfrak{A}}$ and $b'^{\mathfrak{B}}$ are identified. There is an obvious surjection $f : \text{dom}(\mathfrak{A} \uplus \mathfrak{B}) \to \text{dom}(\mathfrak{C})$. Using the fact that $\text{tp}_{\mathcal{K}}(\mathfrak{A}, a^{\mathfrak{A}}) = \text{tp}_{\mathcal{K}}(\mathfrak{B}, b'^{\mathfrak{B}})$ and a simple induction on the structure of concepts C, we can show that for all $C \in \text{cl}(\mathcal{K})$ and $d \in \text{dom}(\mathfrak{A} \uplus \mathfrak{B})$, $d \in C^{\mathfrak{A} \uplus \mathfrak{B}}$ iff $f(d) \in C^{\mathfrak{C}}$. Since $\mathfrak{A}$ and $\mathfrak{B}$ are models of $\mathcal{O}$, it follows that $\mathfrak{C}$ is a model of $\mathcal{O}$. By construction, it is also a model of $\mathcal{D}_{a=b}$. ❑

G Proofs for Section 7.2

Theorem 15 is an immediated consequence of the following theorem.

Theorem 23 *Let $(\mathcal{K}, P, N)$ be a labeled GF-KB and $\Sigma = sig(\mathcal{K})$. Then the following conditions are equivalent:*

1. $(\mathcal{K}, P, N)$ *is strongly openGF-separable;*
2. $(\mathcal{K}, P, N)$ *is strongly GF-separable;*
3. *For all models* $\mathfrak{A}$ *and* $\mathfrak{B}$ *of* $\mathcal{K}$ *and* $\vec{a} \in P$ *and* $\vec{b} \in N$ *we have* $\mathfrak{A}, \vec{a}^{\mathfrak{A}} \not\sim_{GF,\Sigma} \mathfrak{B}, \vec{b}^{\mathfrak{B}}$*;*
4. *For all models* $\mathfrak{A}$ *and* $\mathfrak{B}$ *of* $\mathcal{K}$ *and* $\vec{a} \in P$ *and* $\vec{b} \in N$ *we have* $\mathfrak{A}, \vec{a}^{\mathfrak{A}} \not\sim_{openGF,\Sigma} \mathfrak{B}, \vec{b}^{\mathfrak{B}}$.

Proof. The implications 1. $\Rightarrow$ 2. and 4. $\Rightarrow$ 3. are trivial. Moreover, 2. $\Rightarrow$ 3. and 1. $\Rightarrow$ 4. are immediate from Lemma 3. We prove below 3. $\Rightarrow$ 2., 4. $\Rightarrow$ 1, and 3. $\Rightarrow$ 4. We start with the first implication; the proof of the second is analogous.

3. $\Rightarrow$ 2. Suppose $(\mathcal{K}, P, N)$ is not strongly GF-separable. Let

$$\begin{aligned}\Gamma_P &= \{\varphi(\vec{x}) \in \text{GF}(\Sigma) \mid \forall \vec{a} \in P : \mathcal{K} \models \varphi(\vec{a})\}\\ \Gamma_N &= \{\varphi(\vec{x}) \in \text{GF}(\Sigma) \mid \forall \vec{a} \in N : \mathcal{K} \models \varphi(\vec{a})\}\end{aligned}$$

In what follows we use the fact that Γ_P and Γ_N are closed under conjunction. We say that a set Γ of GF formulas is satisfiable in $\vec{a}$ w.r.t. a KB $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ if the extended (possibly infinite) KB

$$\mathcal{K}' = (\mathcal{O}, \mathcal{D} \cup \{\varphi(\vec{a}) \mid \varphi(\vec{x}) \in \Gamma\})$$

is satisfiable.

Claim 1. (1) There exists $\vec{a} \in P$ such that $\Gamma_P \cup \Gamma_N$ is satisfiable in $\vec{a}$ w.r.t. $\mathcal{K}$. (2) There exists $\vec{a} \in N$ such that $\Gamma_P \cup \Gamma_N$ is satisfiable in a w.r.t. $\mathcal{K}$.

We prove (1), the proof of (2) is dual. Assume $\Gamma_P \cup \Gamma_N$ is not satisfiable in any $\vec{a} \in P$ w.r.t. $\mathcal{K}$. Then Γ_N is not satisfiable in any $\vec{a} \in P$ w.r.t. $\mathcal{K}$. By compactness, there exist $\varphi_{\vec{a}}(\vec{x}) \in \Gamma_N$ such that $\mathcal{K} \models \neg\varphi_{\vec{a}}(\vec{a})$, for all $\vec{a} \in P$. Thus, $\mathcal{K} \models \neg(\bigwedge_{\vec{b} \in P} \varphi_{\vec{b}})(\vec{a})$ for all $\vec{a} \in P$ and $\mathcal{K} \models (\bigwedge_{\vec{b} \in P} \varphi_{\vec{b}})(\vec{a})$ for all $\vec{a} \in N$. However, this is in contradiction to the assumption that $(\mathcal{K}, P, N)$ is not strongly separable.

Now, let $\Gamma_0 = \Gamma_P \cup \Gamma_N$ and consider an enumeration $\varphi_1, \varphi_2, \ldots$ of the remaining GF(Σ) formulas. Then we set inductively, $\Gamma_{i+1} = \Gamma_i \cup \{\varphi_{i+1}\}$ if there exist $\vec{a} \in P$ and $\vec{b} \in N$ such that $\Gamma_i \cup \{\varphi_{i+1}\}$ is satisfiable in both $\vec{a}$ and $\vec{b}$ 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 $\vec{a} \in P$ and $\vec{b} \in N$ such that $\Gamma_i \cup \{\varphi_{i+1}\}$ is satisfiable in both $\vec{a}$ and $\vec{b}$ w.r.t. $\mathcal{K}$ or there are $\vec{a} \in P$ and $\vec{b} \in N$ such that $\Gamma_i \cup \{\neg\varphi_{i+1}\}$ is satisfiable in both $\vec{a}$ and $\vec{b}$ w.r.t. $\mathcal{K}$.

Assume Claim 2 has been proved for $i - 1$. Let w.l.o.g., $\Gamma_i = \Gamma_P \cup \Gamma_N \cup \{\varphi_1, \ldots, \varphi_i\}$. Assume Claim 2 does not hold for i. Then, again w.l.o.g., there is no $\vec{a} \in P$ such that $\Gamma_i \cup \{\varphi_{i+1}\}$ is satisfiable in $\vec{a}$ w.r.t. $\mathcal{K}$ and there is no $\vec{b} \in N$ such that $\Gamma_i \cup \{\neg\varphi_{i+1}\}$ is satisfiable in $\vec{b}$ w.r.t. $\mathcal{K}$. By compactness, there exists $\varphi \in \Gamma_N$ such that $\mathcal{K} \models \varphi'(\vec{a})$ for all $\vec{a} \in P$ and

$$\varphi' = ((\varphi \sqcap \varphi_1 \sqcap \cdots \sqcap \varphi_i) \rightarrow \neg\varphi_{i+1}).$$

Then, by definition, we have $\varphi' \in \Gamma_P$. Then $\varphi' \in \Gamma_i$ and so there is no $b \in N$ such that Γ_i is satisfiable in b w.r.t. $\mathcal{K}$. We have derived a contradiction.

Let $\Gamma = \bigcup_{i \geq 0} \Gamma_i$. Then there exist models $\mathfrak{A}$ and $\mathfrak{B}$ of $\mathcal{K}$ and $\vec{a} \in P$ and $\vec{b} \in P$ such that $\mathfrak{A} \models \varphi(\vec{a})$ for all $\varphi \in \Gamma$ and $\mathfrak{B} \models \varphi(\vec{b})$ for all $\varphi \in \Gamma$. Thus, $\mathfrak{A}, \vec{a} \equiv_{\text{GF}(\Sigma)} \mathfrak{B}, \vec{b}$. We may assume that $\mathfrak{A}$ and $\mathfrak{B}$ are ω-saturated in the sense of classical model theory. By Lemma 3, we obtain $\mathfrak{A}, \vec{a} \sim_{\text{GF},\Sigma} \mathfrak{B}, \vec{b}$, as required.

3. $\Rightarrow$ 4. Suppose there are models $\mathfrak{A}$ and $\mathfrak{B}$ of $\mathcal{K}$ and $\vec{a} \in P$ and $\vec{b} \in N$ such that $\mathfrak{A}, \vec{a}^{\mathfrak{A}} \sim_{\text{openGF},\Sigma} \mathfrak{B}, \vec{b}^{\mathfrak{B}}$. Obtain $\mathfrak{B}'$ from $\mathfrak{A}$ and $\mathfrak{B}$ by removing the parts from $\mathfrak{B}$ that are not connected to $\vec{b}^{\mathfrak{B}}$, and adding a disjoint copy of $\mathfrak{A}$ to the remaining connected component of $\mathfrak{B}$.

Obviously, we have $\mathfrak{A}, \vec{a}^{\mathfrak{A}} \sim_{\text{openGF},\Sigma} \mathfrak{B}', \vec{b}^{\mathfrak{B}'}$ and the connected guarded bisimulation can be extended to a guarded bisimulation by adding all partial isomorphisms between $\mathfrak{A}$ and its copy in $\mathfrak{B}'$. It remains to note that $\mathfrak{B}'$ is a model of $\mathcal{K}$ since it is still a model of $\mathcal{D}$ and one can verify the following.

Claim 3. $\mathfrak{A}$ and $\mathfrak{B}'$ satisfy the same GF(Σ) sentences.

Proof of Claim 3. It suffices to consider sentences of the form $\psi = \exists\vec{y}(\alpha(\vec{y}) \wedge \varphi(\vec{y}))$. Moreover, we can 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 substructure of $\mathfrak{B}'$ and by $(*)$, ψ is also satisfied in $\mathfrak{B}'$. Conversely, assume that ψ is satisfied in $\mathfrak{B}'$ and let $\vec{c}$ be such that $\mathfrak{B}', [\vec{y}/\vec{c}] \models \alpha(\vec{y}) \wedge \varphi(\vec{y})$. If $\vec{c}$ is in the copy of $\mathfrak{A}$ in $\mathfrak{B}'$, then ψ is also satisfied in $\mathfrak{A}$, due to $(*)$. If $\vec{c}$ is connected to $\vec{b}^{\mathfrak{B}}$, then $\mathfrak{A}, \vec{a}^{\mathfrak{A}} \sim_{\text{openGF}(\Sigma)} \mathfrak{B}', \vec{b}^{\mathfrak{B}'}$ implies that also $\mathfrak{A} \models \psi$. This finishes the proof of the Claim. ❑

Moreover, we observe that GF and openGF differ in terms of the size of the strongly separating formula.

Example 8 *Let* $\mathcal{O}$ *be the GF-ontology containing*

$$A_1 \sqsubseteq \forall S.A_1, \quad A_2 \sqsubseteq \forall R.A_2,$$

to propagate A_1 *and* A_2 *along roles* S *and* R*, respectively, and*

$$E_2 \sqcap A_1 \sqsubseteq \exists u.B, \quad E_1 \sqcap A_2 \sqsubseteq \neg\exists u.B.$$

to trigger either that B *holds somewhere or holds nowhere (where* u *is the universal role). Let* $\mathcal{D}$ *contain*

- *an* R*-chain from* a_0 *to* c_n *of length* n *and* $A_1(a_0)$ *and* $E_1(c_n)$*:*
$$R(a_0, c_1), \ldots, R(c_{n-1}, c_n)$$
- *an* S*-chain from* b_0 *to* c'_n *of length* n *with* $A_2(b_0)$ *and* $E_2(c'_n)$*:*
$$S(b_0, c'_1), \ldots, S(c'_{n-1}, c'_n)$$

Let $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ *and let* $P = \{a_0\}$ *and* $N = \{b_0\}$*. In GF (in fact in* $\mathcal{ALC}$ *with the universal role) the following formula strongly separates* $(\mathcal{K}, P, N)$*:*

$$(A_1 \sqcap A_2 \sqcap \neg\exists u.B) \sqcup (A_1 \sqcap \neg A_2).$$

In contrast, any strongly separating formula in openGF has guarded quantifier rank at least n*.*

Proposition 3 *Any openGF-formula strongly separating* $(\mathcal{K}, P, N)$ *has guarded quantifier rank* $\geq n$*.*

Proof. Consider the model $\mathfrak{A}$ with

- $\text{dom}(\mathfrak{A}) = \{a_0, c_1, \ldots, c_n, d_1, \ldots, d_n, b_0, c'_1, \ldots, c'_n, f\}$;
- $A_1^{\mathfrak{A}} = \{a_0, c_1, \ldots, c_n, d_1, \ldots, d_n\}$;
- $A_2^{\mathfrak{A}} = \{a_0, c_1, \ldots, c_n, d_1, \ldots, d_n\} \cup \{b_0\}$;
- $B^{\mathfrak{A}} = \emptyset$;
- $E_1^{\mathfrak{A}} = \{c_n\}$, $E_2^{\mathfrak{A}} = \{c'_n\}$;
- $R^{\mathfrak{A}} = \{(a_0, c_1), \ldots, (c_{n-1}, c_n)\}$;
- $$S^{\mathfrak{A}} = \{(a_0, d_1), \ldots, (d_{n-1}, d_n)\} \cup \{(b_0, c'_1), \ldots, (c'_{n-1}, c'_n)\}$$
- $a_0^{\mathfrak{A}} = a_0$, $b_0^{\mathfrak{A}} = b_0$, $c_i^{\mathfrak{A}} = c_i$, $(c'_i)^{\mathfrak{A}} = c'_i$;

and the model $\mathfrak{B}$ with

- $\text{dom}(\mathfrak{B}) = \{b_0, c'_1, \ldots, c'_n, d_1, \ldots, d_n, a_0, c_1, \ldots, c_n, f\}$;
- $A_1^{\mathfrak{B}} = \{b_0, c'_1, \ldots, c'_n, d_1, \ldots, d_n\} \cup \{a_0\}$;
- $A_2^{\mathfrak{B}} = \{b_0, c'_1, \ldots, c'_n, d_1, \ldots, d_n\}$;
- $B^{\mathfrak{B}} = \{f\}$;
- $E_1^{\mathfrak{B}} = \{c_n\}$, $E_2^{\mathfrak{B}} = \{c'_n\}$;
- $S^{\mathfrak{B}} = \{(b_0, c'_1), \ldots, (c_{n-1}, c'_n)\}$;
- $$R^{\mathfrak{B}} = \{(b_0, d_1), \ldots, (d_{n-1}, d_n)\} \cup \{(a_0, c_1), \ldots, (c_{n-1}, c_n)\}$$
- $a_0^{\mathfrak{B}} = a_0$, $b_0^{\mathfrak{B}} = b_0$, $c_i^{\mathfrak{B}} = c_i$, $(c'_i)^{\mathfrak{B}} = c'_i$.

Then $\mathfrak{A}$ and $\mathfrak{B}$ are both models of $\mathcal{K}$ and $\mathfrak{A}, a_0^{\mathfrak{A}} \sim^{n-1}_{\text{openGF},\text{sig}(\mathcal{K})} \mathfrak{B}, b_0^{\mathfrak{B}}$. ❑

We next provide the announced analogue of Theorem 13; here, the $\mathcal{K}$-type refers to the $\mathcal{K}$-type for GF. Its proof is essentially as the proof of Theorem 13, so we omit it.

Theorem 24 *For every labeled GF-KB $(\mathcal{K}, P, N)$, the following conditions are equivalent:*

1. *$(\mathcal{K}, P, N)$ is strongly GF-separable;*
2. *$(\mathcal{K}, P, N)$ is strongly FO-separable;*
3. *For all $\vec{a} \in P$ and $\vec{b} \in N$, there do not exist models $\mathfrak{A}$ and $\mathfrak{B}$ of $\mathcal{K}$ such that $\vec{a}^{\mathfrak{A}}$ and $\vec{b}^{\mathfrak{B}}$ realize the same $\mathcal{K}$-type;*
4. *The GF-formula $\Phi_1(\vec{x}) \vee \cdots \vee \Phi_n(\vec{x})$ strongly separates $(\mathcal{K}, P, N)$, $\Phi_1(\vec{x}), \ldots, \Phi_n(\vec{x})$ the $\mathcal{K}$-types realizable in $\mathcal{K}, \vec{a}$.*

H Proofs for Section 7.3

We start by introducing appropriate types for FO^2-KBs. Assume that $\mathcal{K} = (\mathcal{O}, \mathcal{D})$ is a FO^2-KB. Let $\mathsf{cl}(\mathcal{K})$ denote the union of the closure under single negation and swapping the variables x, y of the set of subformulas of $\mathcal{O}$ and $\{R(x,x) \mid R \in \mathsf{sig}(\mathcal{K})\}$. The *1-type for* $\mathcal{K}$ of a pointed structure $\mathfrak{A}, a$, denoted $\text{tp}_{\mathcal{K}}(\mathfrak{A}, a)$, is the set of all formulas $\psi(x) \in \mathsf{cl}(\mathcal{K})$ such that $\mathfrak{A} \models \psi(a)$. We denote by $T_x(\mathcal{K})$ the set of all 1-types for $\mathcal{K}$. We say that $t(x) \in T_x(\mathcal{K})$ is *realized* in $\mathfrak{A}, a$ if $t(x) = \text{tp}_{\mathcal{K}}(\mathfrak{A}, a)$. Denote by $t(x)[y/x]$ the set of formulas obtained from $t(x)$ by swapping y and x.

The 2-*type for* $\mathcal{K}$ of a pointed structure $\mathfrak{A}, a, b$, denoted $\text{tp}_{\mathcal{K}}(\mathfrak{A}, a, b)$, is the set of all $R(x,y)$ with $\mathfrak{A} \models R(a,b)$, $R(y,x)$ with $\mathfrak{A} \models R(b,a)$, $\neg R(x,y)$ with $\mathfrak{A} \not\models R(a,b)$, and $\neg R(y,x)$ with $\mathfrak{A} \not\models R(b,a)$, where R is a binary relation in $\mathcal{K}$. In addition, $x = y \in \text{tp}_{\mathcal{K}}(\mathfrak{A}, a, b)$ if $a = b$ and $\neg(x = y) \in \text{tp}_{\mathcal{K}}(\mathfrak{A}, a, b)$ if $a \neq b$. We denote by $T_{x,y}(\mathcal{K})$ the set of all 2-types for $\mathcal{K}$. We say that $t(x,y) \in T_{x,y}(\mathcal{K})$ is *realized* in $\mathfrak{A}, a, b$ if $t(x,y) = \text{tp}_{\mathcal{K}}(\mathfrak{A}, a, b)$.

Types as defined above are not yet sufficiently powerful to ensure that models of $\mathcal{K}$ can be merged. To achieve this we introduce extended types. For $t(x) \in T_x(\mathcal{K})$, we set $t(x)^{=1} = \forall y(\bigwedge t(y) \rightarrow (x = y))$. The *extended 2-type for* $\mathcal{K}$ of a pointed structure $\mathfrak{A}, a, b$, denoted $\text{tp}^*_{\mathcal{K}}(\mathfrak{A}, a, b)$, is the conjunction of

1. $\text{tp}_{\mathcal{K}}(\mathfrak{A}, a, b) \wedge \text{tp}_{\mathcal{K}}(\mathfrak{A}, a) \wedge \text{tp}_{\mathcal{K}}(\mathfrak{A}, b)[y/x]$;
2. $\exists y(\text{tp}_{\mathcal{K}}(\mathfrak{A}, a, c) \wedge \text{tp}_{\mathcal{K}}(\mathfrak{A}, c)[y/x])$ for any $c \in \text{dom}(\mathfrak{A}) \setminus \{a, b\}$ such that $\text{tp}_{\mathcal{K}}(\mathfrak{A}, c)$ is realized only once in $\mathfrak{A}$;
3. $\exists x(\text{tp}_{\mathcal{K}}(\mathfrak{A}, c, b) \wedge \text{tp}_{\mathcal{K}}(\mathfrak{A}, c)[x/y])$ for any $c \in \text{dom}(\mathfrak{A}) \setminus \{a, b\}$ such that $\text{tp}_{\mathcal{K}}(\mathfrak{A}, c)$ 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. $\exists xy \bigwedge \text{tp}_{\mathcal{K}}(\mathfrak{A}, c, d) \wedge \text{tp}_{\mathcal{K}}(\mathfrak{A}, c) \wedge \text{tp}_{\mathcal{K}}(\mathfrak{A}, d)[y/x]$ for any $c \neq d$ such that $\text{tp}_{\mathcal{K}}(\mathfrak{A}, c)$ and $\text{tp}_{\mathcal{K}}(\mathfrak{A}, d)$ are realized only once in $\mathfrak{A}$.

We denote by $T^*_{x,y}(\mathcal{K})$ the set of all extended 2-types for $\mathcal{K}$. We say that $t(x,y) \in T^*_{x,y}(\mathcal{K})$ is *realized* in $\mathfrak{A}, a, b$ if $t(x,y) = \text{tp}_{\mathcal{K}}(\mathfrak{A}, a, b)$.

The *extended 1-type for* $\mathcal{K}$ of a pointed structure $\mathfrak{A}, a$ is defined in the same way with $\text{tp}_{\mathcal{K}}(\mathfrak{A}, a, b), \text{tp}_{\mathcal{K}}(\mathfrak{A}, b)[y/x]$ removed in Point 1 and with Point 3 completely removed. We also define the realization of such types by pointed structures as expected.

Theorem 25 *For every labeled FO^2-KB such that the tuples in $P \cup N$ have length $i \in \{1, 2\}$, the following are equivalent:*

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

Proof. Assume w.l.o.g. that the tuples in P and N have length two. 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 $\vec{a} = (a_1, a_2) \in P$ and $\vec{b} = (b_1, b_2) \in N$ and models $\mathfrak{A}$ and $\mathfrak{B}$ of $\mathcal{K}$ such that the extended 2-types of $\mathfrak{A}, \vec{a}$ and $\mathfrak{B}, \vec{b}$ coincide. We show that there exists a model $\mathfrak{C}$ of $(\mathcal{O}, \mathcal{D}_{\vec{a}=\vec{b}})$. Then, by Theorem 11, $(\mathcal{K}, P, N)$ is not FO-separable. We construct $\mathfrak{C}$ from $\mathfrak{A}$ and

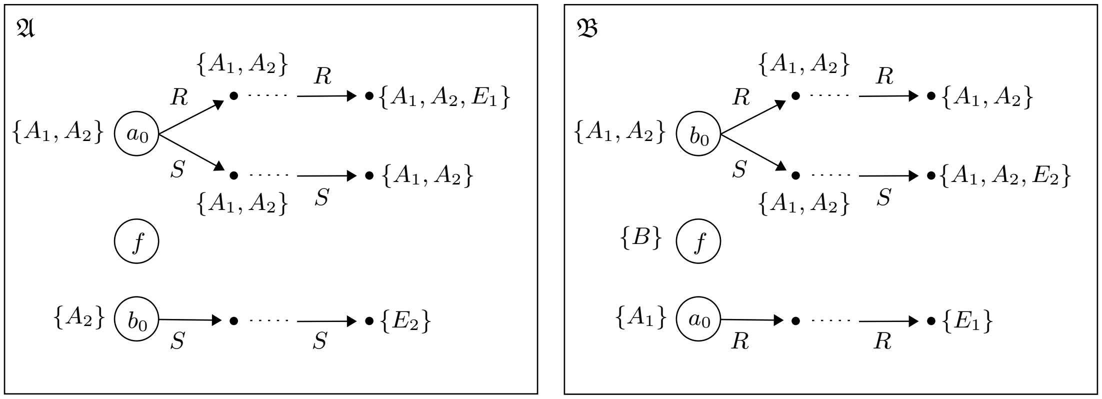
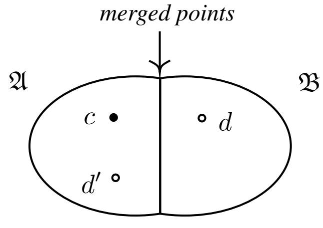


Figure 8: Construction in the proof of Proposition 3.

$\mathfrak{B}$ as follows: 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 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 1-types once. We may thus assume that

- $a_i^{\mathfrak{A}} = b_i^{\mathfrak{B}}$, for $i = 1, 2$;
- it $t(x) \in T_x(\mathcal{K})$ is realized only once, then $a = b$ for the nodes $a \in \mathrm{dom}(\mathfrak{A})$ and $b \in \mathrm{dom}(\mathfrak{B})$ with $t(x) = \mathrm{tp}_{\mathcal{K}}(\mathfrak{A}, a) = \mathrm{tp}_{\mathcal{K}}(\mathfrak{B}, b)$;
- no other nodes are shared by $\mathrm{dom}(\mathfrak{A})$ and $\mathrm{dom}(\mathfrak{B})$.

Now let $\mathrm{dom}(\mathfrak{C}) = \mathrm{dom}(\mathfrak{A}) \cup \mathrm{dom}(\mathfrak{B})$ and define the interpretation of the relation symbols in $\mathfrak{C}$ such that the relativisation of $\mathfrak{C}$ to $\mathrm{dom}(\mathfrak{A})$ and $\mathrm{dom}(\mathfrak{B})$ coincides with $\mathfrak{A}$ and with $\mathfrak{B}$, respectively. This 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 $\mathcal{D}$ and $(c')^{\mathfrak{C}} = (c')^{\mathfrak{B}}$ for all constants $c' \in \mathcal{D}'$ (from the definition of $\mathcal{D}_{\vec{a}=\vec{b}}$). It remains to define the 2-type realized by (c, d) in $\mathfrak{C}$ for $c \in \mathrm{dom}(\mathfrak{C}) \setminus \mathrm{dom}(\mathfrak{B})$ and $d \in \mathrm{dom}(\mathfrak{C}) \setminus \mathrm{dom}(\mathfrak{A})$. Assume such a (c, d) is given. Then the type $\mathrm{tp}_{\mathcal{K}}(\mathfrak{B}, d)$ is realized in $\mathfrak{A}$, by the formulas in Point 5 and 6 of the definition of extended types. Take $d' \in \mathrm{dom}(\mathfrak{A})$ such that $\mathrm{tp}_{\mathcal{K}}(\mathfrak{A}, d') = \mathrm{tp}_{\mathcal{K}}(\mathfrak{B}, d)$.



We may assume that $d' \neq c$ as $\mathrm{tp}_{\mathcal{K}}(\mathfrak{B}, d)$ is realized at least twice in both $\mathfrak{A}$ and in $\mathfrak{B}$. Now interpret the relations $R \in \mathrm{sig}(\mathcal{K})$ in $\mathfrak{C}$ in such a way that $\mathrm{tp}_{\mathcal{K}}(\mathfrak{C}, d, c) = \mathrm{tp}_{\mathcal{K}}(\mathfrak{A}, d', c)$. One can show that $\mathfrak{C}$ is a model of $(\mathcal{O}, \mathcal{D}_{\vec{a}=\vec{b}})$.

❏