Query and Predicate Emptiness in Ontology-Based Data Access

Franz Baader, Meghyn Bienvenu, Carsten Lutz, Frank Wolter

Query and Predicate Emptiness in Ontology-Based Data Access

Franz Baader
TU Dresden, Germany
baader@inf.tu-dresden.de

Meghyn Bienvenu
Laboratoire de Recherche en Informatique, CNRS & Université Paris-Sud, France
meghyn@lri.fr

Carsten Lutz
University of Bremen, Germany
clu@uni-bremen.de

Frank Wolter
Department of Computer Science, University of Liverpool, UK
wolter@liverpool.ac.uk

Abstract

In ontology-based data access (OBDA), database querying is enriched with an ontology that provides domain knowledge and additional vocabulary for query formulation. We identify query emptiness and predicate emptiness as two central reasoning services in this context. Query emptiness asks whether a given query has an empty answer over all databases formulated in a given vocabulary. Predicate emptiness is defined analogously, but quantifies universally over all queries that contain a given predicate. In this paper, we determine the computational complexity of query emptiness and predicate emptiness in the $\mathcal{EL}$, DL-Lite, and $\mathcal{ALC}$-families of description logics, investigate the connection to ontology modules, and perform a practical case study to evaluate the new reasoning services.

1. Introduction

In recent years, the paradigm of ontology-based data access (OBDA) has gained increased popularity. The general idea is to add an ontology to database querying to provide domain knowledge and to enrich the vocabulary that is available for the formulation of queries. This is particularly useful when the data to be queried is highly incomplete and when multiple data sources with diverging vocabularies are integrated (Poggi, Lembo, Calvanese, De Giacomo, Lenzerini, & Rosati, 2008). OBDA has been taken up with particular verve in the area of description logic (DL) where it has been studied intensively both for lightweight DLs such as the members of the DL-Lite and $\mathcal{EL}$ families, which are tractable regarding the data complexity of querying, and for more expressive DLs such as those of the $\mathcal{ALC}$ and $\mathcal{SHIQ}$ families where querying is intractable in data complexity. For the use in OBDA of the former, see for example (Calvanese, De Giacomo, Lembo, Lenzerini, Poggi, Rodriguez-Muro, & Rosati, 2009; Lutz, Toman, & Wolter, 2009; Pérez-Urbina, Motik, & Horrocks, 2009; Chortaras, Trivela, & Stamou, 2011; Eiter, Ortiz, Simkus, Tran, & Xiao, 2012) and the surveys (Krötzsch, 2012; Kontchakov, Rodriguez-Muro, & Zakharyaschev, 2013); for the latter, please see (Glimm, Lutz, Horrocks, & Sattler, 2008; Ortiz, Calvanese, & Eiter, 2008; Bienvenu, ten Cate, Lutz, & Wolter, 2014) and the references in (Ortiz & Simkus, 2012).

The OBDA approach is fueled by the availability of ontologies that aim at providing a 'standard vocabulary' for the targeted application domain. In particular, there are now many such ontologies in the bio-medical domain such as SNOMED CT[1], NCI (Golbeck, Fragoso, Hartel, Hendler, Oberthaler, & Parsia, 2003), and GO[2], which are all formulated in a DL and allow a comparably inexpensive adoption of OBDA in bio-medical applications such as querying electronic medical records (Patel, Cimino, Dolby, Fokoue, Kalyanpur, Kershenbaum, Ma, Schonberg, & Srinivas, 2007). Ontologies of this kind typically have a very broad coverage and their vocabulary often contain tens or even hundreds of thousands of predicates that embrace various subject areas such as anatomy, diseases, medication, and even social context and geographic location. In any given application, only a small fragment of the ontology's vocabulary will actually occur in the data. Still, the remaining predicates are potentially very useful for formulating queries as they are linked to the data vocabulary by the ontology—this is precisely how OBDA enriches the vocabulary available for query formulation.

Due to the size and complexity of the involved ontologies and vocabularies, however, it can be difficult to understand which of the additional predicates are useful for query formulation and how to write meaningful queries in the extended vocabulary. Static analysis tools for analysing these queries would thus be very useful. In this paper, we consider the fundamental static analysis problem of *query emptiness* as well as a natural variation of it called *predicate emptiness*. In the former, the problem is to decide whether a given query q provides an empty answer over all databases formulated in a given data vocabulary Σ. Query emptiness thus helps to identify queries which are useless due to wrong use of the ontology vocabulary. It is a standard static analysis problem in many subareas of database theory such as XML, see e.g. (Benedikt, Fan, & Geerts, 2008) and references therein.

As an example, consider the following simple ontology $\mathcal{O}$:

$$\begin{aligned} \mathsf{DiabetesPatient} &\equiv \mathsf{Patient} \sqcap \exists \mathsf{has\_disease}.\mathsf{Diabetes} \\ \mathsf{DiabetesPatientwithoutMedication} &\equiv \mathsf{Patient} \sqcap \exists \mathsf{has\_disease}.\mathsf{Diabetes} \sqcap \\ &\qquad \neg \exists \mathsf{on\_medication\_for}.\mathsf{Diabetes} \end{aligned}$$

Assume that $\mathcal{O}$ is used to support querying of a medical patient database which has a unary table for the conept names Patient and Diabetes and binary tables for the role names has_disease and on_medication_for, distinguishing in particular between diabetes of type 1 and type 2. For example, the database could be given by the following set $\mathcal{A}$ of assertions:

$$\begin{gathered} \mathsf{Patient}(a), \quad \mathsf{Patient}(b), \\ \mathsf{has\_disease}(a, \mathrm{type1}), \quad \mathsf{on\_medication\_for}(a, \mathrm{type1}), \quad \mathsf{has\_disease}(b, \mathrm{type2}), \\ \mathsf{Diabetes}(\mathrm{type1}), \quad \mathsf{Diabetes}(\mathrm{type2}). \end{gathered}$$

Thus, a is a patient with diabetes of type 1 who is on medication for it and b is a patient with diabetes of type 2. In OBDA, queries are interpreted under an open world assumption and thus one is interested in the *certain answers* to a query q w.r.t. $\mathcal{O}$ and $\mathcal{A}$, that is, the answers to q on which all extension of $\mathcal{A}$ that satisfy the ontology $\mathcal{O}$ are in agreement. For the concrete query

$$q_1(x) = \mathsf{DiabetesPatient}(x),$$

1. http://www.ihtsdo.org/snomed-ct
2. http://geneontology.org/

	QUERY EVALUATION		EMPTINESS	
DL	$\mathcal{IQ}$	$\mathcal{CQ}$	$\mathcal{IQ}$-query / $\mathcal{CQ}$-predicate	$\mathcal{CQ}$-query
$\mathcal{EL}$	PTIME-c.	NP-c.	PTIME-c.	PTIME-c
$\mathcal{EL}_\bot$	PTIME-c.	NP-c.	EXPTIME-c.	EXPTIME-c.
$\mathcal{ELI}$	EXPTIME-c.	EXPTIME-c.	EXPTIME-c.	EXPTIME-c.
Horn-$\mathcal{ALCIF}$	EXPTIME-c.	EXPTIME-c.	EXPTIME-c.	EXPTIME-c.
DL-Lite$_{core\|\mathcal{F}\|\mathcal{R}}$	NLOGSPACE-c.	NP-c.	NLOGSPACE-c.	coNP-c.
DL-Lite$_{horn}$	PTIME-c.	NP-c.	coNP-c.	coNP-c.
$\mathcal{ALC}$	EXPTIME-c.	EXPTIME-c.	NEXPTIME-c.	NEXPTIME-c.
$\mathcal{ALCI}$	EXPTIME-c.	2EXPTIME-c.	NEXPTIME-c.	2EXPTIME-c.
$\mathcal{ALCF}$	EXPTIME-c.	EXPTIME-c.	undecidable	undecidable

Figure 1: Known complexity results for query evaluation and new complexity results for emptiness

a and b are the certain answers to $q_1(x)$ w.r.t. $\mathcal{O}$ and $\mathcal{A}$—despite the fact that the predicate Diabetes-Patient used in $q_1(x)$ does not occur in $\mathcal{A}$. Since $\mathcal{A}$ is formulated using only the data vocabulary

$$\Sigma = \{\mathsf{Patient}, \mathsf{has\_disease}, \mathsf{on\_medication\_for}, \mathsf{Diabetes}\},$$

we say that $q_1(x)$ *is non-empty for* Σ *given* $\mathcal{O}$. We regard this as evidence that $q_1(x)$ is a potentially useful query on databases formulated in vocabulary Σ. As another example, consider the query

$$q_2(x) = \mathsf{DiabetesPatientwithoutMedication}(x)$$

There is no certain answer to $q_2(x)$ w.r.t. $\mathcal{O}$ and $\mathcal{A}$ since, under the open world assumption, the mere absence of the information that b is on medication for diabetes of type 1 or type 2 does not imply that its negation is true. One can even show that, whatever database $\mathcal{A}'$ formulated in vocabulary Σ we use, there will never be any certain answer to $q_2(x)$ w.r.t. $\mathcal{O}$ and $\mathcal{A}'$. In this case, we say that $q_2(x)$ *is empty for* Σ *given* $\mathcal{O}$. In contrast to $q_1(x)$, this query is thus useless on Σ-databases.

We also consider *predicate emptiness*, the problem to decide whether for a given predicate p and data vocabulary Σ, it is the case that all queries q which involve p yield an empty answer over all Σ-databases. In the example above, the predicate DiabetesPatientwithoutMedication is empty w.r.t. $\mathcal{O}$ and Σ for the important class of conjunctive queries (queries constructed from atomic formulas using conjunction and existential quantification). Predicate emptiness can be used to identify predicates in the ontology that are useless for query formulation, before even starting to construct a concrete query. In a graphical user interface, for example, such predicates would not be offered to users for query formulation. Predicate emptiness is loosely related to predicate emptiness in datalog queries as studied e.g. in (Vardi, 1989; Levy, 1993).

The aim of this paper is to perform a detailed study of query emptiness and predicate emptiness for various DLs including members of the $\mathcal{EL}$, DL-Lite, and $\mathcal{ALC}$ families, concentrating on the two most common query languages in DL-based OBDA: instance queries (IQs) and conjunctive

queries (CQs). For all of the resulting combinations of DLs and query languages, we determine the (un)decidability and exact computational complexity of query emptiness and predicate emptiness. Our results are summarized on the right side of Figure 1 and range from PTIME for basic members of the $\mathcal{EL}$ and DL-Lite families via NEXPTIME for basic members of the $\mathcal{ALC}$ family to undecidable for $\mathcal{ALC}$ extended with functional roles ($\mathcal{ALCF}$). We adopt the standard notion of combined complexity, which is measured in terms of the size of the whole input (TBox, data vocabulary, and query or predicate symbol).

Because of the restricted data vocabulary Σ and the quantification over all Σ-databases in their definition, query emptiness and predicate emptiness do not reduce to standard reasoning problems such as query evaluation and query containment. Formally, this is demonstrated by our undecidability result for $\mathcal{ALCF}$, which should be contrasted with the decidability of query entailment and containment in this DL, cf. (Calvanese, De Giacomo, & Lenzerini, 1998). When emptiness is decidable, the complexity still often differs from that of query evaluation. To simplify the comparison, we display in Figure 1 known complexity results for query evaluation in the considered DLs; please consult (Baader, Brandt, & Lutz, 2005, 2008; Krötzsch, Rudolph, & Hitzler, 2007; Eiter, Gottlob, Ortiz, & Simkus, 2008) for the results concerning $\mathcal{EL}$ and its Horn extensions, (Calvanese, De Giacomo, Lembo, Lenzerini, & Rosati, 2007; Artale, Calvanese, Kontchakov, & Zakharyaschev, 2009) for the results on DL-Lite, and (Tobies, 2001; Hustadt, Motik, & Sattler, 2004; Lutz, 2008; Ortiz, Simkus, & Eiter, 2008) for the results on DLs from the $\mathcal{ALC}$ family. By comparing the two sides of Figure 1, we observe that there is no clear relationship between the complexity of emptiness checking and the complexity of query evaluation. Indeed, while the problems are often of similar complexity, there are several cases in which emptiness checking is more difficult than the corresponding query evaluation problem. It can also be the other way around, and complexities can also be incomparable. Note that for the extension $\mathcal{EL}_\bot$ of $\mathcal{EL}$ with the bottom concept $\bot$ (used to express class disjointness), we observe a particularly significant difference between the tractability of evaluating instance queries and the EXPTIME-completeness of checking $\mathcal{IQ}$-query emptiness.

A key ingredient in developing algorithms and establishing upper complexity bounds for emptiness is to show that, when searching for databases that witness non-emptiness (such as the database $\mathcal{A}$ for the non-emptiness of q_1 for Σ given $\mathcal{O}$ in the above example), one can often focus on a single database constructed specifically for that purpose or on a class of databases that are easier to handle than the class of all databases. Which single database / class of databases to consider depends on the DL in question. For this reason, a secondary theme of this paper is to analyze the shape of witness databases. It turns out that in $\mathcal{ALC}$ and its extension $\mathcal{ALCI}$ with inverse roles, we can consider a single exponential-size database whose construction is reminiscent of type elimination and filtration constructions known from the modal logic literature. For $\mathcal{EL}$ and its extension $\mathcal{ELI}$, we may also concentrate on a single witness candidate, but a much simpler one: it consists of all facts that can be constructed using the data vocabulary and a single constant. For other extensions of $\mathcal{EL}$, we use a class of databases as witness candidates, namely those that have a tree or forest structure. In DL-Lite, we may restrict our attention to the class of databases whose size is bounded polynomially w.r.t. the input query and ontology.

To demonstrate that predicate emptiness is a useful reasoning service for static analysis, we perform experiments using the well-known and large-scale medical ontology SNOMED CT coupled with both a real-world data vocabulary (corresponding to terms obtained by analyzing clinical notes from a hospital) and with randomly generated vocabularies. For the real world vocabulary, which contains 8,858 of the $\sim$370,000 concept names and 16 of the 62 role names in SNOMED CT,

16,212 predicates turned out to be non-empty for IQs and 17,339 to be non-empty for CQs. Thus, SNOMED CT provides a very substantial number of additional predicates for query formulation while a large number of other predicates cannot meaningfully be used in queries over Σ-databases; thus, identifying the relevant predicates via predicate emptiness is potentially very helpful.

We also consider the use of query and predicate emptiness for the extraction of modules from an ontology. Thus, instead of using emptiness directly to support query formulation, we show how it can be used to simplify an ontology. A Σ-*substitute* of an ontology is a subset of the ontology that gives the same certain answers to all conjunctive queries over all Σ-databases. Replacing a large ontology with a (potentially quite small) Σ-substitute supports comprehension of the ontology and thereby the formulation of meaningful queries. We show that, for $\mathcal{ELI}$, one can use predicate emptiness to extract a particularly natural Σ-substitute of an ontology, called its $\mathcal{CQ}_\Sigma$*-core*, containing exactly those axioms from the original ontology that contain only predicates which are non-empty for Σ-databases. Thus, *all* predicates in the $\mathcal{CQ}_\Sigma$-core of an ontology can be meaningfully used in queries posed to Σ-databases. In our example, the $\mathcal{CQ}_\Sigma$-core of the ontology $\mathcal{O}$ is

$$\mathcal{O}' = \{\mathsf{DiabetesPatient} \equiv \mathsf{Patient} \sqcap \exists \mathsf{has\_disease.Diabetes}\}.$$

The second axiom was removed because any CQ that contains DiabetesPatientwithoutMedication is empty for Σ given $\mathcal{O}$. To analyze the practical interest of $\mathcal{CQ}_\Sigma$-cores, we carry out a case study where we compute $\mathcal{CQ}_\Sigma$-cores for the ontology SNOMED CT coupled with various signatures, showing that they tend to be drastically smaller than the original ontology and also smaller than $\bot$-modules, a popular way of extracting modules from ontologies (Grau, Horrocks, Kazakov, & Sattler, 2008).

This article is structured as follows. We begin in Section 2 by recalling the syntax and semantics of the description logics considered in this work. In Section 3, we introduce four notions of emptiness ($\mathcal{IQ}$-query, $\mathcal{IQ}$-predicate, $\mathcal{CQ}$-predicate, and $\mathcal{CQ}$-query) and investigate the formal relationships between them. We first observe that $\mathcal{IQ}$-query and $\mathcal{IQ}$-predicate emptiness coincide (so there are only three problems to consider) and that $\mathcal{CQ}$-predicate emptiness corresponds to $\mathcal{CQ}$-query emptiness where CQs are restricted to a very simple form. We also exhibit two polynomial reductions between predicate and query emptiness: for all DLs considered in this paper except those from the DL-Lite family, $\mathcal{CQ}$-predicate emptiness is polynomially reducible to $\mathcal{IQ}$-query emptiness, and for all Horn-DLs considered in this paper, $\mathcal{IQ}$-query emptiness is polynomially reducible to $\mathcal{CQ}$-predicate emptiness.

In Section 4, we investigate the computational complexity and decidability of query and predicate emptiness in the $\mathcal{ALC}$ family of expressive DLs. For $\mathcal{ALC}$ and $\mathcal{ALCI}$, we provide tight complexity bounds, showing NEXPTIME-completeness for all three emptiness problems in $\mathcal{ALC}$ and for $\mathcal{IQ}$-query emptiness and $\mathcal{CQ}$-predicate emptiness in $\mathcal{ALCI}$, and 2EXPTIME-completeness for $\mathcal{CQ}$-query emptiness in $\mathcal{ALCI}$. The situation is dramatically (and surprisingly) different for $\mathcal{ALCF}$, for which all emptiness problems are proven undecidable. As previously mentioned, the complexity upper bounds for $\mathcal{ALC}$ and $\mathcal{ALCI}$ rely on a characterization of non-emptiness in terms of a special witness database. The complexity lower bounds and undecidability results are proven by means of reductions from tiling problems.

In Section 5, we continue our investigation of query and predicate emptiness by considering the DL $\mathcal{EL}$ and its Horn extensions. For plain $\mathcal{EL}$, we provide a simple characterization of non-emptiness in terms of a maximal singleton database, which allows us to show that all three emptiness

problems can be decided in polynomial time. Using the same characterization and the fact that standard reasoning in $\mathcal{ELI}$ is EXPTIME-complete, we obtain EXPTIME-completeness of emptiness checking in $\mathcal{ELI}$. For extensions of $\mathcal{EL}$ that allow for contradictions, the singleton database may not be consistent with the ontology, requiring another approach. To handle such extensions, we show that it is sufficient to consider tree-shaped databases as witnesses for non-emptiness, and we devise a decision procedure for emptiness checking based upon tree automata. We obtain in this manner an EXPTIME upper bound for Horn-$\mathcal{ALCIF}$, which sharply contrasts with our undecidability result for (non-Horn) $\mathcal{ALCF}$. Interestingly, we can show a matching EXPTIME lower bound for the considerably simpler DL $\mathcal{EL}_\bot$, for which standard reasoning tasks are tractable.

In Section 6, we turn our attention to the DL-Lite family of lightweight DLs, which are the most commonly considered DLs for ontology-based data access. We show that $\mathcal{CQ}$-query emptiness is coNP-complete for all considered DL-Lite dialects. For $\mathcal{IQ}$-query emptiness and $\mathcal{CQ}$-predicate emptiness, we show that the complexity depends on whether the considered dialect allows for conjunctions on the left-hand side of axioms. For standard dialects like DL-Lite$_{core}$, DL-Lite$_\mathcal{R}$, and DL-Lite$_\mathcal{F}$, which do not allow for conjunction, we show that $\mathcal{IQ}$-query emptiness and $\mathcal{CQ}$-predicate emptiness are NLOGSPACE-complete. For dialects like DL-Lite$_{horn}$ that admits conjunctions, both $\mathcal{IQ}$-query emptiness and $\mathcal{CQ}$-predicate emptiness are coNP-complete. This difference in complexity is due to the fact that for dialects allowing conjunction, we need to consider witnesses for non-emptiness that are of polynomial size, whereas in the absence of conjunction, it is sufficient to consider databases that consist of a single assertion.

In Section 7, we apply query and predicate emptiness to extract modules of an ontology. We introduce the notion of a Σ-substitute and of a $\mathcal{CQ}_\Sigma$-core of an ontology and show that for $\mathcal{ELI}$ the $\mathcal{CQ}_\Sigma$-core of an ontology is a Σ-substitute. We also relate Σ-substitutes to other notions of module proposed in the literature. In particular, we observe that semantics and syntactic $\bot$-modules (Grau et al., 2008) are examples of Σ-substitutes, and thus, algorithms for computing such modules can also be used to compute (possibly non-minimal) Σ-substitutes. We then demonstrate the potential utility of Σ-substitutes and emptiness checking by experiments based on SNOMED CT.

Finally, in Sections 8 and 9, we conclude the paper by discussing related and future work. Please note that to improve the readability of the text, some technical proofs are deferred to the appendix.

2. Preliminaries

In DLs, *concepts* are inductively defined with the help of a set of *constructors*, starting with countably infinite sets $\mathsf{N_C}$ of *concept names* and $\mathsf{N_R}$ of *role names*. The constructors that are most important in this paper are summarized in Figure 2. An *inverse role* has the form r^- with r a role name and a *role* is a role name or an inverse role. For uniformity, we define double inverse to be identity, that is, $(r^-)^- := r$ for all role names r. Throughout the paper, we use A, B to denote concept names, C, D to denote (possibly compound) concepts, and r and s to denote roles.

We shall be concerned with a variety of different DLs that are all well-known from the literature. The least expressive ones are $\mathcal{EL}$ and DL-Lite, which are the logical underpinnings of the OWL2 profiles OWL2 EL and OWL2 QL, respectively (Motik, Grau, Horrocks, Wu, Fokoue, & Lutz, 2009). In $\mathcal{EL}$, concepts are constructed according to the following grammar using the constructors top concept, conjunction, and existential restriction:

$$C, D \quad ::= \quad \top \quad | \quad A \quad | \quad C \sqcap D \quad | \quad \exists r.C$$

Name	Syntax	Semantics
concept name	A	$A^{\mathcal{I}}$
role name	r	$r^{\mathcal{I}}$
top concept	$\top$	$\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$
bottom concept	$\bot$	$\bot^{\mathcal{I}} = \emptyset$
negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
existential restriction	$\exists r.C$	$\{d \in \Delta^{\mathcal{I}} \mid \exists e \in C^{\mathcal{I}} \text{ with } (d,e) \in r^{\mathcal{I}}\}$
value restriction	$\forall r.C$	$\{d \in \Delta^{\mathcal{I}} \mid \forall e \in \Delta^{\mathcal{I}} : (d,e) \in r^{\mathcal{I}} \Rightarrow e \in C^{\mathcal{I}}\}$
role inverse	r^-	$\{(d,e) \mid (e,d) \in r^{\mathcal{I}}\}$
concept inclusion	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
concept assertion	$A(a)$	$a^{\mathcal{I}} \in A^{\mathcal{I}}$
role assertion	$r(a,b)$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$

Figure 2: Syntax and semantics of DL constructors, TBox axioms, and ABox assertions.

with A ranging over concept names and r over role names. DL-Lite concepts and TBoxes will be introduced in Section 6. The basic expressive DL we consider in this paper is $\mathcal{ALC}$ which is the extension of $\mathcal{EL}$ with the constructors bottom concept, negation, disjunction and value restriction:

$$C, D \quad ::= \quad \top \quad | \quad \bot \quad | \quad A \quad | \quad C \sqcap D \quad | \quad C \sqcup D \quad | \quad \exists r.C \quad | \quad \forall r.C$$

The availability of additional constructors is indicated by concatenation of letters or subscripts: the letter $\mathcal{I}$ stands for the addition of inverse roles (inside existential and value restrictions, if present) and the subscript $\cdot_\bot$ stands for adding $\bot$. This gives, for example, the extension $\mathcal{ALCI}$ of $\mathcal{ALC}$ with inverse roles, whose constructors are exactly the ones shown in Figure 2. It also defines the extension $\mathcal{ELI}_\bot$ of $\mathcal{EL}$ with inverse roles in existential restrictions and the bottom concept.

A *concept inclusion (CI)* in a DL $\mathcal{L}$ takes the form $C \sqsubseteq D$, where C and D are $\mathcal{L}$-concepts. We use $C \equiv D$ as an abbreviation for the CIs $C \sqsubseteq D$ and $D \sqsubseteq C$. In description logic, ontologies are formalized as TBoxes. Given any of the DLs $\mathcal{L}$ introduced above, an *$\mathcal{L}$-TBox* is a finite set of CIs in $\mathcal{L}$. We use the letter $\mathcal{F}$ and write $\mathcal{LF}$ to denote the description logic in which TBoxes consist not only of CIs in $\mathcal{L}$, but also of *functionality statements* $\mathsf{funct}(r)$, where r is a role name or an inverse role (if inverse roles are admitted in $\mathcal{L}$). For example, $\mathcal{ALCF}$ is thus the extension of $\mathcal{ALC}$ in which TBoxes can contain CIs in $\mathcal{ALC}$ and functionality statements for role names. We use the term *axioms* to refer to concept inclusions and functionality statements in a uniform way.

In addition to the DLs introduced above, we also consider some DLs that impose restrictions on which constructors can be used on which side of concept inclusions. A *Horn-$\mathcal{ALCI}$ concept inclusion (CI)* is of the form $L \sqsubseteq R$, where L and R are concepts defined by the syntax rules

$$R, R' ::= \top \mid \bot \mid A \mid \neg A \mid R \sqcap R' \mid \neg L \sqcup R \mid \exists r.R \mid \forall r.R$$
$$L, L' ::= \top \mid \bot \mid A \mid L \sqcap L' \mid L \sqcup L' \mid \exists r.L$$

with A ranging over concept names and r over (potentially inverse) roles. A *Horn-$\mathcal{ALCIF}$-TBox* $\mathcal{T}$ is a finite set of Horn-$\mathcal{ALCI}$ CIs and functionality statements $\mathsf{funct}(r)$. Note that different

definitions of Horn-DLs can be found in the literature (Hustadt, Motik, & Sattler, 2007; Eiter et al., 2008; Kazakov, 2009). As the original definition from (Hustadt et al., 2007) based on polarity is rather technical, we prefer the above (equivalent) definition.

The *size* $|\mathcal{T}|$ of a TBox $\mathcal{T}$ is obtained by taking the sum of the lengths of its axioms, where the length of an axiom is the number of symbols needed to write it as a word.

Databases are represented using an *ABox*, which is a finite set of *concept assertions* $A(a)$ and *role assertions* $r(a, b)$, where a, b are drawn from a countably infinite set $\mathsf{N_I}$ of individual names, A is a concept name, and r is a role name. Note that role assertions cannot use inverse roles. As a shortcut, though, we sometimes write $r^-(a, b) \in \mathcal{A}$ for $r(b, a) \in \mathcal{A}$. We use $\mathsf{Ind}(\mathcal{A})$ to denote the set of individual names used in the ABox $\mathcal{A}$. A *knowledge base* is a pair $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ with $\mathcal{T}$ a TBox and $\mathcal{A}$ an ABox.

The semantics of description logics is defined in terms of an *interpretation* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$. The *domain* $\Delta^\mathcal{I}$ is a non-empty set and the *interpretation function* $\cdot^\mathcal{I}$ maps each concept name $A \in \mathsf{N_C}$ to a subset $A^\mathcal{I}$ of $\Delta^\mathcal{I}$, each role name $r \in \mathsf{N_R}$ to a binary relation $r^\mathcal{I}$ on $\Delta^\mathcal{I}$, and each individual name a to an element $a^\mathcal{I} \in \Delta^\mathcal{I}$. The extension of $\cdot^\mathcal{I}$ to compound concepts is inductively defined as shown in the third column of Figure 2. An interpretation $\mathcal{I}$ *satisfies* (i) a CI $C \sqsubseteq D$ if $C^\mathcal{I} \subseteq D^\mathcal{I}$, a statement $\mathsf{funct}(r)$ if $r^\mathcal{I}$ is functional, (iii) an assertion $A(a)$ if $a^\mathcal{I} \in A^\mathcal{I}$, and (vi) an assertion $r(a, b)$ if $(a^\mathcal{I}, b^\mathcal{I}) \in r^\mathcal{I}$. Then, $\mathcal{I}$ is a *model* of a TBox $\mathcal{T}$ if it satisfies all axioms in $\mathcal{T}$, and a *model* of an ABox $\mathcal{A}$ if it satisfies all assertions in $\mathcal{A}$. A TBox is *satisfiable* if it has a model and an ABox $\mathcal{A}$ is *satisfiable* w.r.t. a TBox $\mathcal{T}$ if $\mathcal{T}$ and $\mathcal{A}$ have a common model. We write $\mathcal{T} \models C \sqsubseteq D$ if all models of $\mathcal{T}$ satisfy the CI $C \sqsubseteq D$.

We consider two types of queries. First, *instance queries (IQs)* take the form $A(v)$, where A is a concept name and v an individual variable taken from a set $\mathsf{N_V}$. Note that instance queries can only be used to query concept names, but not role names. This is the traditional definition, which is due to the fact that role assertions can only be implied by an ABox if they are explicitly contained in it, and thus querying is 'trivial'.[3] The more general *conjunctive queries (CQs)* take the form $\exists \vec{u}\, \varphi(\vec{v}, \vec{u})$ where φ is a conjunction of atoms of the form $A(v)$ and $r(v, v')$ with v, v' individual variables from $\vec{v} \cup \vec{u} \subseteq \mathsf{N_V}$. Variables that are not existentially quantified are called *answer variables*, and the *arity* of q is defined as the number of its answer variables. Queries of arity 0 are called *Boolean*. We use $\mathsf{var}(q)$, $\mathsf{avar}(q)$, and $\mathsf{qvar}(q)$ to denote the set of variables, answer variables, and quantified variables respectively in the query q. From now on, we use $\mathcal{IQ}$ to refer to the set of all IQs and $\mathcal{CQ}$ to refer to the set of all CQs.

Let $\mathcal{I}$ be an interpretation and q an (instance or conjunctive) query q of arity k with answer variables $v_1, \ldots, v_k$. A *match* of q in $\mathcal{I}$ is a mapping $\pi : \mathsf{var}(q) \to \Delta^\mathcal{I}$ such that $\pi(v) \in A^\mathcal{I}$ for all $A(v) \in q$, $(\pi(v), \pi(v')) \in r^\mathcal{I}$ for all $r(v, v') \in q$, and for every answer variable $v \in \mathsf{var}(q)$, there is an individual name a with $\pi(v) = a^\mathcal{I}$. We write $\mathcal{I} \models q[a_1, \ldots, a_k]$ if there is a match π of q in $\mathcal{I}$ such that $\pi(v_i) = a_i^\mathcal{I}$ for every $1 \leq i \leq k$. For a knowledge base $(\mathcal{T}, \mathcal{A})$, we write $\mathcal{T}, \mathcal{A} \models q[a_1, \ldots, a_k]$ if $\mathcal{I} \models q[a_1, \ldots, a_k]$ for all models $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{A}$. In this case, $(a_1, \ldots, a_k)$ is a *certain answer* to q w.r.t. $\mathcal{T}$ and $\mathcal{A}$. We use $\mathsf{cert}_{\mathcal{T},\mathcal{A}}(q)$ to denote the set of all certain answers to q w.r.t. $\mathcal{T}$ and $\mathcal{A}$. Note that when q is a Boolean query, we have $() \in \mathsf{cert}_{\mathcal{T},\mathcal{A}}(q)$ if there is a match for q in every model of $\mathcal{T}, \mathcal{A}$, and otherwise $\mathsf{cert}_{\mathcal{T},\mathcal{A}}(q) = \emptyset$. The *query evaluation problem for CQs for a DL* $\mathcal{L}$ is the problem to decide for an $\mathcal{L}$-TBox $\mathcal{T}$, ABox $\mathcal{A}$, CQ q of arity k, and tuple $\vec{a} \in \mathsf{Ind}(\mathcal{A})^k$, whether $\vec{a} \in \mathsf{cert}_{\mathcal{T},\mathcal{A}}(q)$.

3. This is no longer true in the presence of role hierarchy statements which, however, we do not consider in this paper.

We use the term *predicate* to refer to a concept name or role name and *signature* to refer to a set of predicates (in the introduction, we informally called a signature a vocabulary). Then $\mathsf{sig}(q)$ denotes the set of predicates used in the query q, and similarly $\mathsf{sig}(\mathcal{T})$ and $\mathsf{sig}(\mathcal{A})$ refer to the signature of a TBox $\mathcal{T}$ and ABox $\mathcal{A}$. A Σ-*ABox* is an ABox that uses only predicates from the signature Σ, and likewise for a Σ-*concept*.

In the context of query answering in DLs, it is sometimes useful to adopt the *unique name assumption* (UNA), which requires that $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ for all interpretations $\mathcal{I}$ and all $a, b \in \mathsf{N_I}$ with $a \neq b$. The results obtained in this paper do not depend on the UNA. The following well-known lemma shows that the UNA does not make a difference in $\mathcal{ALCI}$ (and all its fragments such as $\mathcal{EL}$ and $\mathcal{ALC}$) because the certain answers to queries do not change.

Lemma 1 *Let $\mathcal{T}$ be an $\mathcal{ALCI}$-TBox, $\mathcal{A}$ an ABox, and $q \in \mathcal{CQ}$. Then $\mathsf{cert}_{\mathcal{T},\mathcal{A}}(q)$ is identical with and without the UNA.*

An analogous statement fails for $\mathcal{ALCF}$, e.g. because the ABox $\mathcal{A} = \{f(a, b), f(a, b')\}$ is satisfiable w.r.t. the TBox $\mathcal{T} = \{\mathsf{funct}(r)\}$ without the UNA (and thus $\mathsf{cert}_{\mathcal{T},\mathcal{A}}(A(v)) = \emptyset$), but unsatisfiable with the UNA (and thus $\mathsf{cert}_{\mathcal{T},\mathcal{A}}(A(v)) = \mathsf{Ind}(\mathcal{A})$).

3. Query and Predicate Emptiness

We introduce the central notions and reasoning problems studied in this paper, show how they are interrelated, and make some basic observations that are used throughout the paper. The following definition introduces the different notions of emptiness studied in this paper.

Definition 2 *Let $\mathcal{T}$ be a TBox, Σ a signature, and $\mathcal{Q} \in \{\mathcal{IQ}, \mathcal{CQ}\}$ a query language. Then we call*

- *an $\mathcal{Q}$-query q* empty for Σ given $\mathcal{T}$ *if for all Σ-ABoxes $\mathcal{A}$ that are satisfiable w.r.t. $\mathcal{T}$, we have $\mathsf{cert}_{\mathcal{T},\mathcal{A}}(q) = \emptyset$.*
- *a predicate S* $\mathcal{Q}$-empty for Σ given $\mathcal{T}$ *if every $\mathcal{Q}$-query q with $S \in \mathsf{sig}(q)$ is empty for Σ given $\mathcal{T}$.*

In what follows, the signatures Σ used in ABoxes will be called *ABox signatures*. We quantify over *all* ABoxes that are formulated in the ABox signature to address typical database applications in which the data changes frequently, and thus deciding emptiness based on a concrete ABox is not of much interest. As an example, assume that ABoxes are formulated in the signature

$$\Sigma = \{\mathsf{Person}, \mathsf{hasDisease}, \mathsf{DiseaseA}, \mathsf{DiseaseB}\}$$

where here and in the following, all upper-case words are concept names and all lower-case ones are role names. This signature is typically fixed in the application design phase, similar to schema design in databases. For the TBox, we take

$$\mathcal{T} = \{\mathsf{Person} \sqsubseteq \exists \mathsf{hasFather}.(\mathsf{Person} \sqcap \mathsf{Male}),\ \mathsf{DiseaseA} \sqsubseteq \mathsf{InfectiousDisease}\}.$$

Then both the IQ $\mathsf{InfectiousDisease}(v)$ and the CQ $\exists v\, \mathsf{hasFather}(u, v)$ are non-empty for Σ given $\mathcal{T}$ despite using predicates that cannot occur in the data, as witnessed by the Σ-ABoxes $\{\mathsf{DiseaseA}(a)\}$

and $\{\mathsf{Person}(a)\}$, respectively. This illustrates how the TBox $\mathcal{T}$ enriches the vocabulary that is available for query formulation. By contrast, the CQ

$$\exists v \exists v' \, (\mathsf{hasFather}(u, v) \land \mathsf{hasDisease}(v, v') \land \mathsf{InfectiousDisease}(v')),$$

which uses the same predicates plus an additional one from the ABox signature, is empty for Σ given $\mathcal{T}$.

Regarding predicate emptiness, it is interesting to observe that the choice of the query language is important. For example, the predicate Male is $\mathcal{IQ}$-empty for Σ given $\mathcal{T}$, but not $\mathcal{CQ}$-empty as witnessed by the Σ-ABox $\{\mathsf{Person}(a)\}$ and the CQ $\exists v \, \mathsf{Male}(v)$. It thus makes no sense to use Male in instance queries over Σ-ABoxes given $\mathcal{T}$, whereas it can be meaningfully used in conjunctive queries.

As every IQ is also a CQ, a predicate that is $\mathcal{CQ}$-empty must also be $\mathcal{IQ}$-empty. As illustrated by the above example, the converse does not hold. Also note that all role names are $\mathcal{IQ}$-empty for Σ given any $\mathcal{T}$ since a role name cannot occur in an instance query. By contrast, $\mathsf{hasFather}$ is clearly not $\mathcal{CQ}$-empty in the above example.

It follows from Lemma 1 that, in $\mathcal{ALCI}$ and its fragments, query emptiness and predicate emptiness are oblivious as to whether or not the UNA is made, both for $\mathcal{IQ}$ and $\mathcal{CQ}$. As established by the following lemma, this is also true in $\mathcal{ALCIF}$— despite the fact that the certain answers to queries can differ with and without the UNA.

Lemma 3 *Let $\mathcal{T}$ be an $\mathcal{ALCIF}$-TBox. Then each CQ q is empty for Σ given $\mathcal{T}$ with the UNA iff it is empty for Σ given $\mathcal{T}$ without the UNA.*

The proof of Lemma 3 is given in the appendix. For the direction from left to right one assumes that q is non-empty for Σ given $\mathcal{T}$ without the UNA and takes a witness Σ-ABox $\mathcal{A}$. Using a model $\mathcal{I}$ satisfying $\mathcal{A}$ and $\mathcal{T}$ without the UNA and by identifying any $a, b \in \mathsf{Ind}(\mathcal{A})$ with $a^{\mathcal{I}} = b^{\mathcal{I}}$ one can define a Σ-ABox $\mathcal{A}'$ from $\mathcal{A}$ that shows that q is non-empty for Σ given $\mathcal{T}$ with the UNA. Conversely, one assumes that q is non-empty for Σ given $\mathcal{T}$ with the UNA and takes a witness Σ-ABox $\mathcal{A}$. One can use $\mathcal{A}$ to show that q is non-empty for Σ given $\mathcal{T}$ without the UNA.

With the exception of a DL-Lite dialect (containing role inclusions) all DLs considered in this paper are fragments of $\mathcal{ALCIF}$. Thus, we are free to adopt the UNA or not. In the remainder of the paper, we will choose whatever is more convenient, but be careful to always point out explicitly whether the UNA is made or not. For the DL-Lite dialect not covered by the formulation of Lemma 3 we will observe in our discussion of DL-Lite that even Lemma 1 holds and so we are free to adopt the UNA or not in that case as well.

It is also relevant to note that our decision to disallow individual names in query atoms is without any loss of generality. Indeed, it is easily verified that predicate emptiness is the same whether we admit individuals in queries or not. Moreover, there is an immediate reduction of query emptiness for generalized CQs (which may contain individual names) to query emptiness for CQs as defined in this paper: it suffices to replace every individual a in the query by a fresh answer variable x_a, and then test whether the resulting query (without individuals) is empty for Σ given $\mathcal{T}$.

Definition 2 gives rise to four natural decision problems.

Definition 4 *Let $\mathcal{Q} \in \{\mathcal{IQ}, \mathcal{CQ}\}$. Then*

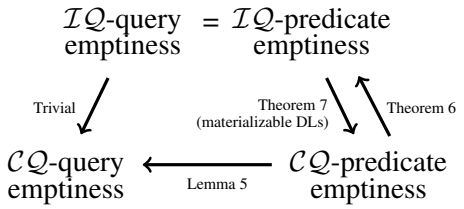

Figure 3: Polytime reductions between emptiness notions.

- $\mathcal{Q}$-query emptiness *is the problem of deciding, given a TBox $\mathcal{T}$, a signature Σ, and an $\mathcal{Q}$-query q, whether q is empty for Σ given $\mathcal{T}$;*
- $\mathcal{Q}$-predicate emptiness *means to decide, given a TBox $\mathcal{T}$, a signature Σ, and a predicate S, whether S is $\mathcal{Q}$-empty for Σ given $\mathcal{T}$.*

Clearly, these four problems are intimately related. In particular, $\mathcal{IQ}$-query emptiness and $\mathcal{IQ}$-predicate emptiness are effectively the same problem since an instance query consists only of a single predicate. For this reason, we will from now on disregard $\mathcal{IQ}$-predicate emptiness and only speak of $\mathcal{IQ}$-query emptiness. In the $\mathcal{CQ}$ case, things are different. Indeed, the following lemma shows that $\mathcal{CQ}$-predicate emptiness corresponds to $\mathcal{CQ}$-query emptiness where CQs are restricted to a very simple form. It is an easy consequence of the fact that, since composite concepts in queries are disallowed, CQs are purely positive, existential, and conjunctive.

Lemma 5 *$A \in \mathsf{N_C}$ (resp. $r \in \mathsf{N_R}$) is $\mathcal{CQ}$-predicate empty for Σ given $\mathcal{T}$ iff the conjunctive query $\exists v\, A(v)$ (resp. $\exists v \exists v'\, r(v,v')$) is empty for Σ given $\mathcal{T}$.*

Lemma 5 allows us to consider only queries of the form $\exists v\, A(v)$ and $\exists v \exists v'\, r(v,v')$ when dealing with $\mathcal{CQ}$-predicate emptiness. From now on, we do this without further notice.

Trivially, $\mathcal{IQ}$-query emptiness is a special case of $\mathcal{CQ}$-query emptiness. The following observation is less obvious and applies to all DLs considered in this paper except those from the DL-Lite family.

Theorem 6 *In any DL contained in $\mathcal{ALCIF}$ that admits CIs $\exists r.B \sqsubseteq B$ and $\exists r.\top \sqsubseteq B$ with B a concept name, $\mathcal{CQ}$-predicate emptiness can be polynomially reduced to $\mathcal{IQ}$-query emptiness.*

Proof. Let $\mathcal{T}$ be a TBox, Σ a signature, B a concept name that does not occur in $\mathcal{T}$ and Σ, and s a role name that does not occur in $\mathcal{T}$ and Σ. We prove that

1. A is $\mathcal{CQ}$-predicate empty for Σ given $\mathcal{T}$ iff the IQ $B(v)$ is empty for $\Sigma \cup \{s\}$ given the TBox $\mathcal{T}' = \mathcal{T} \cup \mathcal{T}_B \cup \{A \sqsubseteq B\}$, where $\mathcal{T}_B = \{\exists r.B \sqsubseteq B \mid r = s \text{ or } r \text{ occurs in } \mathcal{T}\}$;
2. r is $\mathcal{CQ}$-predicate empty for Σ given $\mathcal{T}$ iff the IQ $B(v)$ is empty for $\Sigma \cup \{s\}$ given the TBox $\mathcal{T}' = \mathcal{T} \cup \mathcal{T}_B \cup \{\exists r.\top \sqsubseteq B\}$, where $\mathcal{T}_B$ is as above.

The proofs of Points 1 and 2 are similar, and we concentrate on Point 1. First suppose that A is $\mathcal{CQ}$-predicate non-empty for Σ given $\mathcal{T}$. Then there is a Σ-ABox $\mathcal{A}$ such that $\mathcal{T}, \mathcal{A} \models \exists v\, A(v)$. Choose an $a_0 \in \mathsf{Ind}(\mathcal{A})$ and set $\mathcal{A}' := \mathcal{A} \cup \{s(a_0, b) \mid b \in \mathsf{Ind}(\mathcal{A})\}$. Using the fact that $\mathcal{T}, \mathcal{A} \models \exists v\, A(v)$ and the definition of $\mathcal{A}'$ and $\mathcal{T}'$, it can be shown that $\mathcal{T}', \mathcal{A}' \models B(a_0)$. For the converse direction,

suppose that B is $\mathcal{IQ}$-query non-empty for $\Sigma \cup \{s\}$ given $\mathcal{T}'$. Then there is a $\Sigma \cup \{s\}$-ABox $\mathcal{A}'$ such that $\mathcal{T}', \mathcal{A}' \models B(a)$ for some $a \in \mathsf{Ind}(\mathcal{A}')$. Let $\mathcal{A}$ be obtained from $\mathcal{A}'$ by removing all assertions $s(a, b)$. Using the fact that $\mathcal{T}', \mathcal{A}' \models B(a)$ and the definition of $\mathcal{A}'$ and $\mathcal{T}'$, it can be shown that $\mathcal{T}, \mathcal{A} \models \exists v\, A(v)$. ❑

Figure 3 gives an overview of the available polytime reductions between our four (rather: three) problems. In terms of computational complexity, $\mathcal{CQ}$-query emptiness is thus (potentially) the hardest problem, while $\mathcal{CQ}$-predicate emptiness is the simplest. More precisely, if $\mathcal{CQ}$-query emptiness in a DL $\mathcal{L}$ belongs to a complexity class $\mathfrak{C}$ (larger than or equal to PTIME), then $\mathcal{IQ}$-query emptiness and CQ-predicate emptiness in $\mathcal{L}$ are also in $\mathfrak{C}$. Moreover, for DLs $\mathcal{L}$ satisfying the conditions of Theorem 6, $\mathfrak{C}$-hardness of $\mathcal{CQ}$-predicate emptiness in $\mathcal{L}$ implies $\mathfrak{C}$-hardness of $\mathcal{CQ}$-query emptiness and $\mathcal{IQ}$-query emptiness in $\mathcal{L}$.

Under certain conditions, we can also prove the converse of Theorem 6. Following (Lutz & Wolter, 2012), we call a model $\mathcal{I}$ of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$ a *materialization* of $\mathcal{T}$ and $\mathcal{A}$ if for every CQ q of arity k and tuple $\vec{a} \in \mathsf{Ind}(\mathcal{A})^k$, $\mathcal{I} \models q[\vec{a}]$ iff $\mathcal{T}, \mathcal{A} \models q[\vec{a}]$. A DL $\mathcal{L}$ is called *materializable* if for every ABox $\mathcal{A}$ that is satisfiable w.r.t. $\mathcal{T}$ there exists a materialization of $\mathcal{T}$ and $\mathcal{A}$. Typical DL-Lite dialects, the DL $\mathcal{EL}$, and Horn-extensions of $\mathcal{EL}$ such as $\mathcal{ELIF}_\bot$ are materializable (Lutz & Wolter, 2012).

Theorem 7 *Let $\mathcal{L}$ be a materializable DL that admits CIs of the form $A_1 \sqcap A_2 \sqsubseteq A_3$, where $A_1, A_2, A_3 \in \mathsf{N_C}$. Then, in $\mathcal{L}$, $\mathcal{IQ}$-query emptiness can be polynomially reduced to $\mathcal{CQ}$-predicate emptiness.*

Proof. We claim that the IQ $A(v)$ is empty for Σ given $\mathcal{T}$ iff B is $\mathcal{CQ}$-empty for $\Sigma \cup \{X\}$ given the TBox $\mathcal{T}' = \mathcal{T} \cup \{A \sqcap X \sqsubseteq B\}$, where B and X are concept names that do not occur in $\mathcal{T}$.

For the "if" direction, assume that $A(v)$ is $\mathcal{IQ}$ non-empty for Σ given $\mathcal{T}$, and let $\mathcal{A}$ be a Σ-ABox such that $\mathcal{T}, \mathcal{A} \models A(a)$ for some $a \in \mathsf{Ind}(\mathcal{A})$. Set $\mathcal{A}' := \mathcal{A} \cup \{X(a)\}$. It is easy to see that $\mathcal{T}', \mathcal{A}' \models \exists v\, B(v)$ and thus B is $\mathcal{CQ}$-predicate non-empty for $\Sigma \cup \{X\}$ given $\mathcal{T}'$.

For the "only if" direction, assume that B is $\mathcal{CQ}$ non-empty for $\Sigma \cup \{X\}$ given $\mathcal{T}'$, and let $\mathcal{A}'$ be a $\Sigma \cup \{X\}$-ABox which is satisfiable with $\mathcal{T}'$ and such that $\mathcal{T}', \mathcal{A}' \models \exists v\, B(v)$. We may assume that $X(a) \in \mathcal{A}'$ for all $a \in \mathsf{Ind}(\mathcal{A}')$ as adding these assertions can neither result in unsatisfiability w.r.t. $\mathcal{T}'$ nor invalidate $\mathcal{T}', \mathcal{A}' \models \exists v\, B(v)$. By our assumption of materializability, there exists a materialization $\mathcal{I}'$ of $\mathcal{T}'$ and $\mathcal{A}'$ and we have $\mathcal{I}' \models \exists v\, B(v)$. By definition of $\mathcal{T}'$, we may assume that $X^{\mathcal{I}'} = \mathsf{Ind}(\mathcal{A}')$ and $B^{\mathcal{I}'} = A^{\mathcal{I}'} \cap X^{\mathcal{I}'}$ (if this is not the case, we can take a modified version, $\mathcal{I}''$, of $\mathcal{I}'$ that is defined by setting $X^{\mathcal{I}''} := \mathsf{Ind}(\mathcal{A}')$, $B^{\mathcal{I}''} := A^{\mathcal{I}'} \cap X^{\mathcal{I}'}$, and $Y^{\mathcal{I}''} := Y^{\mathcal{I}'}$ for all remaining concept and role names Y; then $\mathcal{I}''$ still satisfies $\mathcal{T}'$ and $\mathcal{A}'$ since $X^{\mathcal{I}'} \supseteq \mathsf{Ind}(\mathcal{A}')$ and $A \sqcap X \sqsubseteq B$ is the only inclusion containing X or B and it is still a materialization since $Y^{\mathcal{I}''} \subseteq Y^{\mathcal{I}'}$ for all concept and role names Y). But then $\mathcal{I}' \models \exists v\, B(v)$ implies that there is an $a \in \mathsf{Ind}(\mathcal{A}')$ with $\mathcal{I}' \models B(a)$. Since $\mathcal{I}'$ is a materialization of $\mathcal{T}'$ and $\mathcal{A}'$, this implies $\mathcal{T}', \mathcal{A}' \models B(a)$. By definition of $\mathcal{T}'$, this implies $\mathcal{T}, \mathcal{A} \models A(a)$, where $\mathcal{A}$ is obtained from $\mathcal{A}'$ by dropping all assertions of the form $X(b)$. Since $\mathcal{A}$ is a Σ-ABox and satisfiable w.r.t. $\mathcal{T}$ (since $\mathcal{A}'$ is satisfiable w.r.t. $\mathcal{T}'$), it witnesses that $A(v)$ is non-empty for Σ given $\mathcal{T}$. ❑

As a final observation in this section, we note that deciding query and predicate emptiness is essentially just ABox satisfiability whenever Σ contains all symbols used in the TBox. By the described reductions, it suffices to consider CQ-query emptiness. For a CQ $q = \exists \vec{u}\, \varphi(\vec{v}, \vec{u})$ we

associate with every individual variable v in q an individual name a_v and set

$$\mathcal{A}_q = \{A(a_v) \mid A(v) \text{ is a conjunct in } \varphi\} \cup \{r(a_v, a_{v'}) \mid r(v, v') \text{ is a conjunct in } \varphi\}.$$

Theorem 8 *Let $\mathcal{T}$ be an $\mathcal{ALCIF}$-TBox, Σ a signature with $\mathsf{sig}(\mathcal{T}) \subseteq \Sigma$, and q a CQ. Then q is empty for Σ given $\mathcal{T}$ iff $\mathsf{sig}(q) \not\subseteq \Sigma$ or $\mathcal{A}_q$ is unsatisfiable w.r.t. $\mathcal{T}$.*

Proof. ("If") Assume that q is non-empty for Σ given $\mathcal{T}$. Then there is a Σ-ABox $\mathcal{A}$ that is satisfiable w.r.t. $\mathcal{T}$ and such that $\mathsf{cert}_{\mathcal{T},\mathcal{A}}(q) \neq \emptyset$. This clearly implies $\mathsf{sig}(q) \subseteq \Sigma$ since otherwise there is a predicate in $\mathsf{sig}(q) \setminus \Sigma$ and we can find a model of $\mathcal{T}$ and $\mathcal{A}$ in which this predicate is interpreted as the empty set, which would mean $\mathsf{cert}_{\mathcal{T},\mathcal{A}}(q) = \emptyset$. It thus remains to show that $\mathcal{A}_q$ is satisfiable w.r.t. $\mathcal{T}$. To this end, let $\mathcal{I}$ be a model of $\mathcal{T}$ and $\mathcal{A}$. Since $\mathsf{cert}_{\mathcal{T},\mathcal{A}}(q) \neq \emptyset$, there exists a match π of q in $\mathcal{I}$. Modify $\mathcal{I}$ by setting $a_v^{\mathcal{I}} = \pi(v)$ for all variables v used in q. It is readily checked that the modified $\mathcal{I}$ is a model of $\mathcal{A}_q$ and $\mathcal{T}$, thus $\mathcal{A}_q$ is satisfiable w.r.t. $\mathcal{T}$ as required.

("Only if") Assume that $\mathsf{sig}(q) \subseteq \Sigma$ and $\mathcal{A}_q$ is satisfiable w.r.t. $\mathcal{T}$. Then $\mathsf{sig}(\mathcal{A}_q) \subseteq \Sigma$. Since clearly $\mathsf{cert}_{\mathcal{T},\mathcal{A}_q}(q) \neq \emptyset$, this means that q is non-empty for Σ given $\mathcal{T}$. ❑

4. Expressive Description Logics

We consider query and predicate emptiness in the $\mathcal{ALC}$ family of expressive DLs, establishing tight complexity results for $\mathcal{ALC}$ and $\mathcal{ALCI}$, and undecidability for $\mathcal{ALCF}$. We start with upper bound proofs, showing that $\mathcal{IQ}$-query emptiness and $\mathcal{CQ}$-predicate emptiness in $\mathcal{ALCI}$ are in NEXPTIME, and so is $\mathcal{CQ}$-query emptiness in $\mathcal{ALC}$. Moreover, we establish that $\mathcal{CQ}$-query emptiness is in 2EXPTIME. We then move on to the corresponding lower bound proofs and also establish undecidability of all considered emptiness problems in $\mathcal{ALCF}$.

4.1 Upper Bounds

The first main step in our proofs is to show that, when deciding emptiness problems in $\mathcal{ALC}$ or $\mathcal{ALCI}$, it suffices to consider a *single special* Σ-ABox. Specifically, we show how to construct from a given satisfiable TBox $\mathcal{T}$ and ABox signature Σ the *canonical Σ-ABox* $\mathcal{A}_{\mathcal{T},\Sigma}$ such that for every CQ q, we have $\mathsf{cert}_{\mathcal{T},\mathcal{A}_{\mathcal{T},\Sigma}}(q) \neq \emptyset$ if and only if there exists a Σ-ABox $\mathcal{A}$ that is satisfiable w.r.t. $\mathcal{T}$ such that $\mathsf{cert}_{\mathcal{T},\mathcal{A}}(q) \neq \emptyset$. We then prove NEXPTIME upper bounds for $\mathcal{IQ}$-query emptiness in $\mathcal{ALCI}$ by computing $\mathcal{A}_{\mathcal{T},\Sigma}$ (in exponential time) and then guessing a model of $\mathcal{A}_{\mathcal{T},\Sigma}$ (of exponential size in $|\mathcal{T}|$ and Σ) that falsifies the query; an 2EXPTIME upper bound for $\mathcal{CQ}$-query emptiness in $\mathcal{ALCI}$ is obtained in an even simpler way by computing $\mathcal{A}_{\mathcal{T},\Sigma}$ and then checking whether $\mathsf{cert}_{\mathcal{T},\mathcal{A}_{\mathcal{T},\Sigma}}(q) = \emptyset$ using known algorithms. Signficantly more work is required to obtain a NEXPTIME upper bound for $\mathcal{CQ}$-query emptiness in $\mathcal{ALC}$. We again construct $\mathcal{A}_{\mathcal{T},\Sigma}$, but need to exercise a lot of care to check whether $\mathsf{cert}_{\mathcal{T},\mathcal{A}_{\mathcal{T},\Sigma}}(q) = \emptyset$ without leaving NEXPTIME.

Let $\mathcal{T}$ be a satisfiable $\mathcal{ALCI}$-TBox and Σ an ABox signature. To define the canonical Σ-ABox for $\mathcal{T}$, we introduce the well-known notion of types (or Hintikka sets) (Pratt, 1979; Kaminski, Schneider, & Smolka, 2011). The *closure* $\mathsf{cl}(\mathcal{T}, \Sigma)$ of $\mathcal{T}$ and Σ is the smallest set that contains $\Sigma \cap \mathsf{N_C}$ as well as all concepts that occur (potentially as a subconcept) in $\mathcal{T}$ and is closed under single negations. A *type* for $\mathcal{T}$ and Σ is a set $t \subseteq \mathsf{cl}(\mathcal{T}, \Sigma)$ such that for some model $\mathcal{I}$ of $\mathcal{T}$ and some $d \in \Delta^{\mathcal{I}}$, we have $t = t^{\mathcal{I}}(d)$, where $t^{\mathcal{I}}(d) = \{C \in \mathsf{cl}(\mathcal{T}, \Sigma) \mid d \in C^{\mathcal{I}}\}$ is the *type for $\mathcal{T}$ and Σ realized by d in $\mathcal{I}$*. Let $\mathfrak{T}_{\mathcal{T},\Sigma}$ denote the set of all types for $\mathcal{T}$ and Σ. For a role name r and $t, t' \in \mathfrak{T}_{\mathcal{T},\Sigma}$, we say that the pair (t, t') is *r-coherent* and write $t \rightsquigarrow_r t'$ if

- $C \in t'$ whenever $\forall r.C \in t$, and
- $C \in t$ whenever $\forall r^-.C \in t'$.

It can be seen that the above implies also corresponding conditions on existential restrictions, such as $C \in t'$ and $\exists r.C \in \mathsf{cl}(C, \mathcal{T})$ implies $\exists r.C \in t$.

Definition 9 (Canonical Σ-ABox) *Let $\mathcal{T}$ be a satisfiable $\mathcal{ALCI}$-TBox and Σ an ABox signature. Fix a (distinct) individual name a_t for each $t \in \mathfrak{T}_{\mathcal{T},\Sigma}$. The canonical Σ-ABox $\mathcal{A}_{\mathcal{T},\Sigma}$ for $\mathcal{T}$ is defined as follows:*

$$\begin{aligned} \mathcal{A}_{\mathcal{T},\Sigma} \quad = \quad & \{A(a_t) \mid A \in t \text{ and } t \in \mathfrak{T}_{\mathcal{T},\Sigma}, A \in \Sigma \cap \mathsf{N_C}\} \cup \\ & \{r(a_t, a_{t'}) \mid \ t \rightsquigarrow_r t' \text{ and } t, t' \in \mathfrak{T}_{\mathcal{T},\Sigma}, r \in \Sigma \cap \mathsf{N_R}\}. \end{aligned}$$

The cardinality of $\mathfrak{T}_{\mathcal{T},\Sigma}$ is at most exponential in the size of $\mathcal{T}$ and the cardinality of Σ, and the set $\mathfrak{T}_{\mathcal{T},\Sigma}$ can be computed in exponential time by making use of well-known EXPTIME procedures for concept satisfiability w.r.t. TBoxes in $\mathcal{ALCI}$ (Gabbay, Kurucz, Wolter, & Zakharyaschev, 2003) (page 72). Thus, $\mathcal{A}_{\mathcal{T},\Sigma}$ is of exponential size and can be computed in exponential time. It is interesting to note that the ABox $\mathcal{A}_{\mathcal{T},\Sigma}$ is a finitary version of the canonical model for basic modal logic and is essentially identical to the model constructed by Pratt's type elimination procedure (Pratt, 1979); in fact, it is exactly identical when $\Sigma \subseteq \mathsf{sig}(\mathcal{T})$. We now show that $\mathcal{A}_{\mathcal{T},\Sigma}$ is satisfiable w.r.t. $\mathcal{T}$.

Lemma 10 *Let $\mathcal{T}$ be a satisfiable $\mathcal{ALCI}$-TBox and Σ an ABox signature. Then $\mathcal{A}_{\mathcal{T},\Sigma}$ is satisfiable w.r.t. $\mathcal{T}$.*

Proof. Let the interpretation $\mathcal{I}_{\mathcal{T},\Sigma}$ be defined by setting

$$\begin{aligned} \Delta^{\mathcal{I}_{\mathcal{T},\Sigma}} &= \mathfrak{T}_{\mathcal{T},\Sigma} \\ A^{\mathcal{I}_{\mathcal{T},\Sigma}} &= \{t \in \mathfrak{T}_{\mathcal{T},\Sigma} \mid A \in t\} \\ r^{\mathcal{I}_{\mathcal{T},\Sigma}} &= \{(t, t') \in \mathfrak{T}_{\mathcal{T},\Sigma} \times \mathfrak{T}_{\mathcal{T},\Sigma} \mid t \rightsquigarrow_r t'\} \end{aligned}$$

for all $A \in \mathsf{N_C}$ and $r \in \mathsf{N_R}$. One can prove by induction on the structure of C that for all $C \in \mathsf{cl}(\mathcal{T}, \Sigma)$, we have $C \in t$ iff $t \in C^{\mathcal{I}_{\mathcal{T},\Sigma}}$. By definition of types, $C \sqsubseteq D \in \mathcal{T}$ and $C \in t$ imply $D \in t$. Thus, $\mathcal{I}_{\mathcal{T},\Sigma}$ is a model of $\mathcal{T}$. It is an immediate consequence of the definition of $\mathcal{A}_{\mathcal{T},\Sigma}$ that $\mathcal{I}_{\mathcal{T},\Sigma}$ is also a model of $\mathcal{A}_{\mathcal{T},\Sigma}$; in fact, $\mathcal{A}_{\mathcal{T},\Sigma}$ can be regarded as the reduct of $\mathcal{I}_{\mathcal{T},\Sigma}$ to signature Σ. ❑

As a crucial tool for analyzing the properties of canonical ABoxes, we introduce homomophisms between ABoxes. Let $\mathcal{A}$ and $\mathcal{A}'$ be ABoxes. An *ABox homomorphism* from $\mathcal{A}$ to $\mathcal{A}'$ is a total function $h : \mathsf{Ind}(\mathcal{A}) \to \mathsf{Ind}(\mathcal{A}')$ such that the following conditions are satisfied:

- $A(a) \in \mathcal{A}$ implies $A(h(a)) \in \mathcal{A}'$;
- $r(a, b) \in \mathcal{A}$ implies $r(h(a), h(b)) \in \mathcal{A}'$.

The next lemma identifies a central property of ABox homomorphisms regarding query answering.

Lemma 11 *If $\mathcal{T}$ is an $\mathcal{ALCI}$-TBox, q is a CQ such that $\mathcal{T}, \mathcal{A} \models q[a_1, \ldots, a_n]$, and h is an ABox homomorphism from $\mathcal{A}$ to $\mathcal{A}'$, then $\mathcal{T}, \mathcal{A}' \models q[h(a_1), \ldots, h(a_n)]$.*

Proof. We prove the contrapositive. Thus assume that $\mathcal{T}, \mathcal{A}' \not\models q[h(a_1), \ldots, h(a_n)]$. Then there is a model $\mathcal{I}'$ of $\mathcal{T}$ and $\mathcal{A}'$ such that $\mathcal{I}' \not\models q[h(a_1), \ldots, h(a_n)]$. Define a model $\mathcal{I}$ by starting with $\mathcal{I}'$ and reinterpreting the individual names in $\mathsf{Ind}(\mathcal{A})$ by setting $a^{\mathcal{I}} = h(a)^{\mathcal{I}'}$ for each $a \in \mathsf{Ind}(\mathcal{A})$. Since individual names do not occur in $\mathcal{T}$, $\mathcal{I}$ is a model of $\mathcal{T}$. It is also a model of $\mathcal{A}$: if $A(a) \in \mathcal{A}$, then $A(h(a)) \in \mathcal{A}'$ by definition of ABox homomorphisms. Since $\mathcal{I}'$ is a model of $\mathcal{A}'$ and by definition of $\mathcal{I}$, it follows that $a^{\mathcal{I}} \in A^{\mathcal{I}}$. The case $r(a, b) \in \mathcal{A}$ is analogous. Finally, $\mathcal{I}' \not\models q[h(a_1), \ldots, h(a_n)]$ and the definition of $\mathcal{I}$ yield $\mathcal{I} \not\models q[a_1, \ldots, a_n]$. We have thus shown that $\mathcal{T}, \mathcal{A} \not\models q[a_1, \ldots, a_n]$. ❑

The following lemma characterizes satisfiability of Σ-ABoxes w.r.t. $\mathcal{T}$ by the existence of an ABox homomorphism into $\mathcal{A}_{\mathcal{T},\Sigma}$.

Lemma 12 *Let $\mathcal{T}$ be a satisfiable $\mathcal{ALCI}$-TBox and Σ an ABox signature. A Σ-ABox $\mathcal{A}$ is satisfiable w.r.t. $\mathcal{T}$ iff there is an ABox homomorphism from $\mathcal{A}$ to $\mathcal{A}_{\mathcal{T},\Sigma}$.*

Proof. Assume $\mathcal{A}$ is satisfiable w.r.t. $\mathcal{T}$. Let $\mathcal{I}$ be a model of $\mathcal{T}$ and $\mathcal{A}$. Define a homomorphism h from $\mathcal{A}$ to $\mathcal{A}_{\mathcal{T},\Sigma}$ by setting $h(a) = a_t$, where t is the type for $\mathcal{T}$ and Σ realized by $a^{\mathcal{I}}$ in $\mathcal{I}$. Using the definition of $\mathcal{A}_{\mathcal{T},\Sigma}$, one can see that h is indeed an ABox homomorphism. Conversely, let h be an ABox homomorphism from $\mathcal{A}$ to $\mathcal{A}_{\mathcal{T},\Sigma}$. By Lemma 10, $\mathcal{A}_{\mathcal{T},\Sigma}$ is satisfiable w.r.t. $\mathcal{T}$. The proof of Lemma 11 shows how one can construct a model of $\mathcal{T}$ and $\mathcal{A}$ from a model of $\mathcal{T}$ and $\mathcal{A}_{\mathcal{T},\Sigma}$ using the homomorphism h. Thus $\mathcal{A}$ is satisfiable w.r.t. $\mathcal{T}$. ❑

We are now ready to prove the main property of $\mathcal{A}_{\mathcal{T},\Sigma}$ regarding emptiness, as discussed at the beginning of this section.

Theorem 13 *Let $\mathcal{T}$ be a satisfiable $\mathcal{ALCI}$-TBox and Σ an ABox signature. A CQ q is empty for Σ given $\mathcal{T}$ iff $\mathsf{cert}_{\mathcal{T},\mathcal{A}_{\mathcal{T},\Sigma}}(q) = \emptyset$.*

Proof. The "only if" direction follows directly from the fact that $\mathcal{A}_{\mathcal{T},\Sigma}$ is satisfiable w.r.t. $\mathcal{T}$ (by Lemma 10). For the "if" direction, let $\mathsf{cert}_{\mathcal{T},\mathcal{A}_{\mathcal{T},\Sigma}}(q) = \emptyset$. To show that q is empty for Σ given $\mathcal{T}$, take a Σ-ABox $\mathcal{A}$ that is satisfiable w.r.t. $\mathcal{T}$. By Lemmas 11 and 12, $\mathsf{cert}_{\mathcal{T},\mathcal{A}_{\mathcal{T},\Sigma}}(q) = \emptyset$ implies $\mathsf{cert}_{\mathcal{T},\mathcal{A}}(q) = \emptyset$, as required. ❑

We now employ Theorem 13 to prove the NEXPTIME upper bounds for $\mathcal{IQ}$-query emptiness.

Theorem 14 *In $\mathcal{ALCI}$, $\mathcal{IQ}$-query emptiness is in* NEXPTIME.

Proof. Let $\mathcal{T}$ be a satisfiable TBox, Σ an ABox signature, and $A(v)$ an IQ for which emptiness for Σ given $\mathcal{T}$ is to be decided. We employ the following:

Fact. For any ABox $\mathcal{A}$, if $\mathcal{T}, \mathcal{A} \not\models A(a)$, then there exists a model $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{A}$ with $a^{\mathcal{I}} \notin A^{\mathcal{I}}$ and $|\Delta^{\mathcal{I}}| \leq |\mathsf{Ind}(\mathcal{A})| + 2^{|\mathcal{T}|}$.

Proof of Fact. If $\mathcal{T}, \mathcal{A} \not\models A(a)$, then there exists a model $\mathcal{J}$ of $\mathcal{T}$ and $\mathcal{A}$ with $a^{\mathcal{J}} \notin A^{\mathcal{J}}$. We may assume that $\{a^{\mathcal{J}} \mid a \in \mathsf{Ind}(\mathcal{A})\}$ is disjoint from the domain $\mathfrak{T}_{\mathcal{T},\Sigma_0}$ of the interpretation $\mathcal{I}_{\mathcal{T},\Sigma_0}$ defined in the proof of Lemma 10 (where we assume that $\Sigma_0 := \emptyset$). Now define $\mathcal{I}$ as the union of the restriction of $\mathcal{J}$ to $\{a^{\mathcal{J}} \mid a \in \mathsf{Ind}(\mathcal{A})\}$ and the interpretation $\mathcal{I}_{\mathcal{T},\Sigma_0}$ expanded by adding to $r^{\mathcal{I}}$ all pairs

- $(a^{\mathcal{J}}, t)$ such that $t^{\mathcal{J}}(a^{\mathcal{J}}) \rightsquigarrow_r t$, $a \in \mathsf{Ind}(\mathcal{A})$, and $t \in \mathfrak{T}_{\mathcal{T},\Sigma_0}$;

- $(t, a^{\mathcal{J}})$ such that $t \rightsquigarrow_r t^{\mathcal{J}}(a^{\mathcal{J}})$, $a \in \mathsf{Ind}(\mathcal{A})$, and $t \in \mathfrak{T}_{\mathcal{T},\Sigma_0}$.

Then $\mathcal{I}$ is a model of $\mathcal{T}$ and $\mathcal{A}$ with $a^{\mathcal{I}} \notin A^{\mathcal{I}}$ of the required size. This finishes the proof of the fact.

The NEXPTIME algorithm computes the canonical ABox $\mathcal{A}_{\mathcal{T},\Sigma}$ (in exponential time) and guesses for every $a \in \mathsf{Ind}(\mathcal{A}_{\mathcal{T},\Sigma})$ a model $\mathcal{I}_a$ with $|\Delta^{\mathcal{I}_a}| \leq |\mathsf{Ind}(\mathcal{A}_{\mathcal{T},\Sigma})| + 2^{|\mathcal{T}|}$. The algorithm returns "yes" if for all $a \in \mathsf{Ind}(\mathcal{A}_{\mathcal{T},\Sigma})$:

1. $\mathcal{I}_a$ is a model of $\mathcal{A}_{\mathcal{T},\Sigma}$ and $\mathcal{T}$, and
2. $a^{\mathcal{I}_a} \notin A^{\mathcal{I}_a}$.

Both conditions can be checked in exponential time. Thus, by Theorem 13 and the fact above, the algorithm returns "yes" iff $A(v)$ is empty for Σ given $\mathcal{T}$. ❑

Note that by Theorem 6 $\mathcal{CQ}$-predicate emptiness in $\mathcal{ALCI}$ is in NEXPTIME as well. For $\mathcal{CQ}$-query emptiness in $\mathcal{ALCI}$, we can easily derive a 2EXPTIME upper bound using $\mathcal{A}_{\mathcal{T},\Sigma}$ and results from (Calvanese et al., 1998) on the complexity of query answering in DLs.

Theorem 15 *In $\mathcal{ALCI}$, $\mathcal{CQ}$-query emptiness is in* 2EXPTIME.

Proof. The 2EXPTIME algorithm is obtained by first computing the canonical ABox $\mathcal{A}_{\mathcal{T},\Sigma}$ and $\mathsf{cert}_{\mathcal{T},\mathcal{A}_{\mathcal{T},\Sigma}}(q)$, and then checking whether the latter is empty. This can be done in 2EXPTIME since it is shown in (Calvanese et al., 1998) that for all $\mathcal{T}$, $\mathcal{A}$, and q with $\mathcal{T}$ an $\mathcal{ALCI}$-TBox, the set $\mathsf{cert}_{\mathcal{T},\mathcal{A}}(q)$ can be computed in time $2^{p(m)\cdot 2^{p(n)}}$ with p a polynomial, m the size of $\mathcal{T} \cup \mathcal{A}$, and n the size of q. ❑

We provide an improved NEXPTIME upper bound for $\mathcal{CQ}$-query emptiness in $\mathcal{ALC}$, which will allow us to show that for $\mathcal{ALC}$ our three emptiness problems have the same complexity.

Theorem 16 *In $\mathcal{ALC}$, $\mathcal{CQ}$-query emptiness is in* NEXPTIME.

The proof is somewhat technical and reuses the machinery of fork rewritings and spoilers from (Lutz, 2008), where it is proved that the combined complexity of CQ-answering in the DL $\mathcal{SHQ}$ is in EXPTIME. More concretely, we show that one can decide emptiness of a CQ q for an ABox signature Σ given an $\mathcal{ALC}$-TBox $\mathcal{T}$ by guessing an extension $\mathcal{A}^e_{\mathcal{T},\Sigma}$ of the canonical ABox $\mathcal{A}_{\mathcal{T},\Sigma}$ with assertions that prevent any possible match of q and then checking that $\mathcal{A}^e_{\mathcal{T},\Sigma}$ is satisfiable w.r.t. $\mathcal{T}$. For example, if q is $A(x)$, then it obviously suffices to add $\neg A(a)$ for every individual a in $\mathcal{A}_{\mathcal{T},\Sigma}$ (we allow here also complex concepts to be used in an ABox). The general case requires a careful analysis of the assertions that have to be considered as additions, and this is where the mentioned fork rewritings and spoilers enter the picture. In fact, they are used to prove that, since there are no inverse roles in the TBox, it suffices to consider extensions of $\mathcal{A}_{\mathcal{T},\Sigma}$ that contain no additional individual names and where the additional assertions are taken from a candidate set whose size is polynomial in the size of $\mathcal{A}_{\mathcal{T},\Sigma}$ and q. It remains to show that satisfiability of $(\mathcal{T}, \mathcal{A}^e_{\mathcal{T},\Sigma})$ can be decided (non-deterministically) in time single exponential in the size of $\mathcal{T}$ and q. Full details are given in the appendix.

4.2 Lower Bounds and Undecidability

We prove matching lower bounds for the upper complexity bounds presented above and show undecidability of $\mathcal{IQ}$-query emptiness, $\mathcal{CQ}$-predicate emptiness, and $\mathcal{CQ}$-query emptiness for $\mathcal{ALCF}$. The undecidability proof and the NEXPTIME-lower bound proof are by reduction of two different tiling problems, where the first asks for a tiling of a finite rectangle of any (unbounded) size and the second asks for a tiling of the $2^n \times 2^n$-square. The 2EXPTIME lower bound for $\mathcal{CQ}$-query emptiness in $\mathcal{ALCI}$ is by a straightforward reduction to query entailment in $\mathcal{ALCI}$. We begin with the NEXPTIME lower bound.

Theorem 17 *In $\mathcal{ALC}$, $\mathcal{CQ}$-predicate emptiness is* NEXPTIME*-hard.*

Proof. The proof is by reduction of a NEXPTIME-hard $2^n \times 2^n$-tiling problem. An instance of this tiling problem is given by a natural number $n > 0$ (coded in unary) and a triple $(\mathfrak{T}, H, V)$ with $\mathfrak{T}$ a non-empty, finite set of *tile types* including an *initial tile* T_{init} to be placed on the lower left corner, $H \subseteq \mathfrak{T} \times \mathfrak{T}$ a *horizontal matching relation*, and $V \subseteq \mathfrak{T} \times \mathfrak{T}$ a *vertical matching relation*. A *tiling* for $(\mathfrak{T}, H, V)$ is a map $f : \{0, \dots, 2^n - 1\} \times \{0, \dots, 2^n - 1\} \to \mathfrak{T}$ such that $f(0, 0) = T_{\mathsf{init}}$, $(f(i, j), f(i + 1, j)) \in H$ for all $i < 2^n - 1$, and $(f(i, j), f(i, j + 1)) \in V$ for all $j < 2^n - 1$. It is NEXPTIME-complete to decide whether an instance of the $2^n \times 2^n$-tiling problem has a tiling.

For the reduction, let $n > 0$ and $(\mathfrak{T}, H, V)$ be an instance of the $2^n \times 2^n$-tiling problem with $\mathfrak{T} = \{T_1, \dots, T_p\}$. We construct a signature Σ and a TBox $\mathcal{T}$ in $\mathcal{ALC}$ such that $(\mathfrak{T}, H, V)$ has a solution if and only if a selected concept name A is $\mathcal{CQ}$-predicate empty for Σ given $\mathcal{T}$. For the proof, it is convenient to impose the UNA.

When formulating the reduction TBox, we use role names x and y to represent the $2^n \times 2^n$-grid and two binary counters X and Y for counting from 0 to $2^n - 1$. The counters use concept names $X_0, \dots, X_{n-1}$ and $Y_0, \dots, Y_{n-1}$ as their bits, respectively. $\mathcal{T}$ contains the following well-known inclusions stating that the value of the counter $X_0, \dots, X_{n-1}$ is incremented when going to x-successors and the value of the counter $Y_0, \dots, Y_{n-1}$ is incremented when going to y-successors: for $k = 1, \dots, n - 1$,

$$\bigsqcap_{0 \leq j < k} X_j \sqsubseteq (\neg X_k \sqcup \forall x.\neg X_k) \sqcap (X_k \sqcup \forall x.X_k)$$

and

$$\bigsqcup_{0 \leq j < k} \neg X_j \sqsubseteq (\neg X_k \sqcup \forall x.X_k) \sqcap (X_k \sqcup \forall x.\neg X_k)$$

and similarly for $Y_0, \dots, Y_{n-1}$ and y. $\mathcal{T}$ also states that the value of the counter X does not change when going to y-successors and the value of the counter Y does not change when going to x-successors: for $i = 0, \dots, n - 1$,

$$X_i \sqsubseteq \forall y.X_i, \quad \neg X_i \sqsubseteq \forall y.\neg X_i$$

and

$$Y_i \sqsubseteq \forall x.Y_i, \quad \neg Y_i \sqsubseteq \forall x.\neg Y_i.$$

In addition, $\mathcal{T}$ states that when the counter X is $2^n - 1$, it does not have an x-successor and if the counter Y is $2^n - 1$, it does not have a y-successor:

$$X_0 \sqcap \dots \sqcap X_{n-1} \sqsubseteq \forall x.\bot, \quad Y_0 \sqcap \dots \sqcap Y_{n-1} \sqsubseteq \forall y.\bot.$$

$\mathcal{T}$ states that T_{init} holds at $(0,0)$ and that the tiling is complete:

$$\neg X_0 \sqcap \cdots \sqcap \neg X_{n-1} \sqcap \neg Y_0 \sqcap \cdots \sqcap \neg Y_{n-1} \sqsubseteq T_{\mathsf{init}}, \quad \top \sqsubseteq \bigsqcup_{1 \leq i \leq p} T_i,$$

$\mathcal{T}$ states that if a tiling condition is violated, then A is true:

- for all $0 \leq i < j \leq p$: $T_i \sqcap T_j \sqsubseteq A$,
- for all $0 \leq i, j \leq p$ such that $(T_i, T_j) \notin H$: $T_i \sqcap \exists x.T_j \sqsubseteq A$,
- for all $0 \leq i, j \leq p$ such that $(T_i, T_j) \notin V$: $T_i \sqcap \exists y.T_j \sqsubseteq A$.

Finally, since we cannot use negation in ABoxes, $\mathcal{T}$ states that concept names $\overline{X}_0, \ldots, \overline{X}_{n-1}$ and $\overline{Y}_0, \ldots, \overline{Y}_{n-1}$ are equivalent to $\neg X_0, \ldots, \neg X_{n-1}$ and $\neg Y_0, \ldots, \neg Y_{n-1}$, respectively: for $i = 1, \ldots, n-1$:

$$\neg X_i \sqsubseteq \overline{X}_i, \quad X_i \sqsubseteq \neg \overline{X}_i, \quad \neg Y_i \sqsubseteq \overline{Y}_i, \quad Y_i \sqsubseteq \neg \overline{X}_i.$$

We set $\Sigma = \{x, y, X_0, \ldots, X_{n-1}, Y_0, \ldots, Y_{n-1}, \overline{X}_0, \ldots, \overline{X}_{n-1}, \overline{Y}_0, \ldots, \overline{Y}_{n-1}\}$ and show

Claim. $(\mathfrak{T}, H, V)$ has no $2^n \times 2^n$-tiling iff there exists a Σ-ABox $\mathcal{A}$ that is satisfiable w.r.t. $\mathcal{T}$ such that $\mathcal{T}, \mathcal{A} \models \exists v\, A(v)$.

Proof of claim. Assume first that $(\mathfrak{T}, H, V)$ has no $2^n \times 2^n$-tiling. To construct $\mathcal{A}$, we regard the pairs (i, j) with $i < 2^n$ and $j < 2^n$ as individual names and let $x((i,j),(i+1,j)) \in \mathcal{A}$ for $i < 2^n - 1$ and $y((i,j),(i,j+1)) \in \mathcal{A}$ for $j < 2^n - 1$. We also set $X_k(i,j) \in \mathcal{A}$ if the kth bit of i is 1, $\overline{X}_k(i,j) \in \mathcal{A}$ if the kth bit of i is 0, $Y_k(i,j) \in \mathcal{A}$ if the kth bit of j is 1, and $\overline{Y}_k(i,j) \in \mathcal{A}$ if the kth bit of j is 0. It is readily checked that $\mathcal{A}$ is satisfiable w.r.t. $\mathcal{T}$ and that $\mathcal{T}, \mathcal{A} \models \exists v\, A(v)$.

Conversely, assume that $(\mathfrak{T}, H, V)$ has a $2^n \times 2^n$-tiling given by $f : \{0, \ldots, 2^n - 1\} \times \{0, \ldots, 2^n - 1\} \to \mathfrak{T}$. Let $\mathcal{A}$ be a Σ-ABox that is satisfiable w.r.t. $\mathcal{T}$. We show that $\mathcal{T}, \mathcal{A} \not\models \exists v\, A(v)$. Let $\mathcal{I}$ be a model of $\mathcal{T}$ and $\mathcal{A}$. If $A^{\mathcal{I}} = \emptyset$, we are done. Otherwise re-define the interpretation of $T_1, \ldots, T_p$ and A as follows. Associate with every $d \in \Delta^{\mathcal{I}}$ the uniquely determined pair (i_d, j_d) given by the values of the counters X and Y at d in $\mathcal{I}$. Then set $d \in T_k^{\mathcal{I}}$ iff $f(i_d, j_d) = T_k$ and let $A^{\mathcal{I}} = \emptyset$. It is readily checked that the resulting interpretation is still a model of $\mathcal{T}$ and $\mathcal{A}$. ❏

It follows from the preceding result that $\mathcal{IQ}$-query emptiness and $\mathcal{CQ}$-query emptiness for $\mathcal{ALC}$ and $\mathcal{ALCI}$ are NEXPTIME-hard. For $\mathcal{CQ}$-query emptiness in $\mathcal{ALCI}$, we can easily derive a 2EXPTIME lower bound from results on the complexity of query entailment in $\mathcal{ALCI}$.

Theorem 18 *In $\mathcal{ALCI}$, $\mathcal{CQ}$-query emptiness is 2*EXPTIME*-hard.*

Proof. It is shown in (Lutz, 2008) that CQ entailment in $\mathcal{ALCI}$ is 2EXPTIME-hard already for ABoxes of the form $\{A(a)\}$ and for Boolean CQs. This can clearly be strengthened to empty ABoxes: replace $A(a)$ with the empty ABox and compensate for this by adding to the TBox $\top \sqsubseteq \exists r.A$ with r a fresh role name. It thus remains to observe that a Boolean CQ q is entailed by $\mathcal{T}$ and the empty ABox iff q is non-empty for $\Sigma = \emptyset$ and $\mathcal{T}$. ❏

We now show that the simple addition of functional roles to $\mathcal{ALC}$ leads to undecidability of $\mathcal{CQ}$-predicate emptiness, thus also of $\mathcal{IQ}$-query emptiness and $\mathcal{CQ}$-query emptiness. The proof is by reduction from a tiling problem that asks for a tiling of a rectangle of finite size (which is neither fixed nor bounded). The reduction involves a couple of technical tricks such as using concept names that are not in Σ as universally quantified second-order variables. This allows us to enforce a grid structure using standard frame axioms from modal logic (which are second-order in nature). The reduction requires role names that are both functional and inverse functional. Since inverse functionality cannot be expressed in $\mathcal{ALCF}$, we also use a modal logic frame axiom to enforce that a different, (forwards) functional role name is interpreted as the inverse of the role name that we are interested in. Of course, undecidability carries over to variants of $\mathcal{ALCF}$ that use a concept constuctor $(\leq 1\, r)$ instead of functional roles as an additional sort, and to all DLs with qualified or unqualified number restrictions.

Theorem 19 *In $\mathcal{ALCF}$, $\mathcal{CQ}$-predicate emptiness is undecidable.*

An instance of the aforementioned tiling problem is given by a triple $(\mathfrak{T}, H, V)$ with $\mathfrak{T}$ a non-empty, finite set of *tile types* including an *initial tile* T_{init} to be placed on the lower left corner and a *final tile* T_{final} to be placed on the upper right corner, $H \subseteq \mathfrak{T} \times \mathfrak{T}$ a *horizontal matching relation*, and $V \subseteq \mathfrak{T} \times \mathfrak{T}$ a *vertical matching relation*. A *tiling* for $(\mathfrak{T}, H, V)$ is a map $f : \{0, \dots, n\} \times \{0, \dots, m\} \to \mathfrak{T}$ such that $n, m \geq 0$, $f(0,0) = T_{\mathsf{init}}$, $f(n,m) = T_{\mathsf{final}}$, $(f(i,j), f(i+1,j)) \in H$ for all $i < n$, and $(f(i,j), f(i,j+1)) \in v$ for all $i < m$. It is undecidable whether an instance of the tiling problem has a tiling.

For the reduction, let $(\mathfrak{T}, H, V)$ be an instance of the tiling problem with $\mathfrak{T} = \{T_1, \dots, T_p\}$. We construct a signature Σ and a TBox $\mathcal{T}$ such that $(\mathfrak{T}, H, V)$ has a solution if and only if a selected concept name A is $\mathcal{CQ}$-predicate non-empty for Σ given $\mathcal{T}$.

The ABox signature is $\Sigma = \{T_1, \dots, T_p, x, y, x^-, y^-\}$ where $T_1, \dots, T_p$ are used as concept names, and x, y, x^-, and y^- are *functional* role names. We use the role names x and y to represent horizontal and vertical adjacency of points in the rectangle, and the role names x^- and y^- to simulate the inverses of x and y. In $\mathcal{T}$, we use additional auxiliary concept names. In particular U and R mark the upper and right border of the rectangle, $Z_{c,1}$, $Z_{c,2}$, $Z_{x,1}$, $Z_{x,2}$, $Z_{y,1}$, $Z_{y,2}$ serve as second-order variables, C serves as a flag which indicates that grid cells are closed at the position where it is set, and I_x and I_y are similar flags for the intended behaviour of the role names x, x^- and y, y^-. The concept name Y is propagated through the grid from the upper right corner to the lower left one, ensuring that these flags are set everywhere, that every position of the grid is labeled with at least one tile type, and that the horizontal and vertical matching conditions are satisfied. When the lower left corner of the grid is reached, we set A as a flag, which is what the query $\exists v\, A(v)$ asks for.

The TBox $\mathcal{T}$ is defined as the set of the following CIs, where (T_i, T_j, T_ℓ) range over all triples from $\mathfrak{T}$ such that $(T_i, T_j) \in H$ and $(T_i, T_\ell) \in V$ and where B_e, for $e \in \{c, x, y\}$, ranges over all Boolean combinations of the concept names $Z_{e,1}$ and $Z_{e,2}$, i.e., over all concepts $L_1 \sqcap L_2$ where L_i is $Z_{e,i}$ or $\neg Z_{e,i}$:

$$
\begin{aligned}
T_{\mathsf{final}} &\sqsubseteq Y \sqcap U \sqcap R \\
\exists x.(U \sqcap Y \sqcap T_j) \sqcap I_x \sqcap T_i &\sqsubseteq U \sqcap Y \\
\exists y.(R \sqcap Y \sqcap T_\ell) \sqcap I_y \sqcap T_i &\sqsubseteq R \sqcap Y
\end{aligned}
$$

$$\exists x.(T_j \sqcap Y \sqcap \exists y.Y) \sqcap \exists y.(T_\ell \sqcap Y \sqcap \exists x.Y) \sqcap I_x \sqcap I_y \sqcap C \sqcap T_i \sqsubseteq Y$$
$$Y \sqcap T_{\mathsf{init}} \sqsubseteq A$$
$$B_x \sqcap \exists x.\exists x^-.B_x \sqsubseteq I_x$$
$$B_y \sqcap \exists y.\exists y^-.B_y \sqsubseteq I_y$$
$$\exists x.\exists y.B_c \sqcap \exists y.\exists x.B_c \sqsubseteq C$$
$$U \sqsubseteq \forall y.\bot \quad R \sqsubseteq \forall x.\bot \quad U \sqsubseteq \forall x.U \quad R \sqsubseteq \forall y.R \quad \bigsqcup_{1 \le s < t \le p} T_s \sqcap T_t \sqsubseteq \bot$$

The CIs for I_x and I_y are responsible for enforcing that x^- is the inverse of x and y^- the inverse of y, at least at those ABox individuals that we are interested in. In fact, if the ABox contains assertions $x(a,b)$ and $x^-(b,c)$ and thus violates the intended interpretation of x and x^-, then we can interpret $Z_{x,1}$ and $Z_{x,2}$ such that the left-hand sides of all possible instantiations of the CI for I_x are violated, e.g. by making $Z_{x,1}$ and $Z_{x,2}$ true at a, but false at c. If the ABox contains $x(a,b), x^-(b,a)$, then this is not possible. Since x^- and y^- are functional, we thus enforce that x and y are inverse functional. The CIs for C achieve in a similar way the closing of grid cells, i.e., that the x-y-successor and the y-x-successor of every relevant ABox individual coincide. However, as can be seen from the proofs, this only works if x and y are inverse functional.

To establish Theorem 19, it suffices to prove the following lemma (see the appendix for details).

Lemma 20 *$(\mathfrak{T}, H, V)$ admits a tiling iff there is a Σ-ABox $\mathcal{A}$ that is satisfiable with $\mathcal{T}$ and such that $\mathcal{T}, \mathcal{A} \models \exists v\, A(v)$.*

5. $\mathcal{EL}$ and its Horn Extensions

We study query and predicate emptiness for the DL $\mathcal{EL}$ and several of its Horn extensions. First, we show that, in plain $\mathcal{EL}$, all three emptiness problems can be decided in polynomial time. The reason is that in this case, the exponential-size canonical ABox $\mathcal{A}_{\mathcal{T},\Sigma}$ from Section 4 can be replaced with a *total Σ-ABox* $\mathcal{A}_\Sigma$ that contains only a single individual which is an instance of all Σ-predicates. Note that $\mathcal{A}_\Sigma$ is satisfiable w.r.t. any $\mathcal{EL}$-TBox because $\mathcal{EL}$ cannot express unsatisfiability. The same approach works for $\mathcal{ELI}$, but in this case one obtains only an ExpTime upper bound which is optimal since subsumption for $\mathcal{ELI}$ is already ExpTime-hard (Baader et al., 2005, 2008). As soon as unsatisfiability can be expressed, the situation changes drastically. In fact, we show that even in $\mathcal{EL}_\bot$ where subsumption and other standard reasoning tasks are still tractable, (all versions of) emptiness are ExpTime-hard. Nevertheless, emptiness in Horn extensions of $\mathcal{EL}$ turns out to be easier than emptiness in expressive DLs. In contrast to the undecidability result for $\mathcal{ALCF}$ and the NExpTime-hardness result for $\mathcal{ALC}$, emptiness is in ExpTime even in Horn-$\mathcal{ALCIF}$. The reason is the *unravelling tolerance* of Horn description logics observed in (Lutz & Wolter, 2012), which implies that when looking for ABoxes that witness non-emptiness, we can restrict ourselves to *tree-shaped* ones. This enables the use of automata-theoretic techniques to decide emptiness.

5.1 $\mathcal{EL}$ and $\mathcal{ELI}$

We begin by showing that in $\mathcal{EL}$, $\mathcal{CQ}$-query emptiness, $\mathcal{CQ}$-predicate emptiness, and $\mathcal{IQ}$-query emptiness are all in PTime. The proofs are transparent and simple. For any signature Σ, the *total Σ-ABox* is $\mathcal{A}_\Sigma := \{A(a_\Sigma) \mid A \in \Sigma\} \cup \{r(a_\Sigma, a_\Sigma) \mid r \in \Sigma\}$.

Lemma 21 *Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox and Σ a signature. Any CQ q is empty for Σ given $\mathcal{T}$ iff* $\mathsf{cert}_{\mathcal{T},\mathcal{A}_\Sigma}(q) = \emptyset$.

Proof. The proof is a simplified version of the proof of Theorem 13. The (contrapositive of the) "only if" direction follows from the fact that $\mathcal{A}_\Sigma$ is satisfiable w.r.t. $\mathcal{T}$. For the "if" direction, let $\mathsf{cert}_{\mathcal{T},\mathcal{A}_\Sigma}(q) = \emptyset$. To show that q is empty for Σ given $\mathcal{T}$, take a Σ-ABox $\mathcal{A}$. Define an ABox homomorphism from $\mathcal{A}$ to $\mathcal{A}_\Sigma$ by setting $h(a) := a_\Sigma$ for all $a \in \mathsf{Ind}(\mathcal{A}_\Sigma)$. By Lemmas 11 and 12, $\mathsf{cert}_{\mathcal{T},\mathcal{A}_\Sigma}(q) = \emptyset$ implies $\mathsf{cert}_{\mathcal{T},\mathcal{A}}(q) = \emptyset$, as required. ❑

Lemma 21 provides a polytime reduction of $\mathcal{CQ}$-query emptiness (and, therefore, $\mathcal{IQ}$-query emptiness and $\mathcal{CQ}$-predicate emptiness) to the query evaluation problem for CQs over $\mathcal{A}_\Sigma$. In the appendix, we show that due to the simple shape of $\mathcal{A}_\Sigma$, checking whether $\mathsf{cert}_{\mathcal{T},\mathcal{A}_\Sigma}(q) = \emptyset$ can be done in polynomial time. In fact, we give a polytime procedure that either returns "$\mathsf{cert}_{\mathcal{T},\mathcal{A}_\Sigma}(q) = \emptyset$" or succeeds in constructing a Boolean forest-shaped query $\widehat{q}$ that is empty for Σ given $\mathcal{T}$ iff q is. The construction relies on the fact that, as an immediate consequence of results in (Lutz & Wolter, 2010), emptiness of q for Σ given $\mathcal{T}$ implies the existence of a model $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{A}_\Sigma$ with $\mathsf{cert}_{\mathcal{T},\mathcal{A}_\Sigma}(q) = \emptyset$ and such that $\mathcal{I}$ has the shape of a tree extended with reflexive loops at the root. Checking $\mathcal{T}, \mathcal{A}_\Sigma \not\models \widehat{q}$ then only requires to answer concept queries in the extension $\mathcal{EL}^u$ of $\mathcal{EL}$ with the universal role, which is possible in PTIME (Lutz & Wolter, 2010). We obtain the following result.

Theorem 22 *In $\mathcal{EL}$, $\mathcal{IQ}$-query emptiness and $\mathcal{CQ}$-query emptiness can be decided in* PTIME.

A matching PTIME lower bound for Theorem 22 can be shown by a reduction of subsumption in $\mathcal{EL}$, which is PTIME-hard (Haase, 2007). Consider an $\mathcal{EL}$-TBox $\mathcal{T}$ and $\mathcal{EL}$-concepts C and D. Then the CI $C \sqsubseteq D$ follows from $\mathcal{T}$ if, and only if, the IQ $B(v)$ is non-empty for the signature $\{A\}$ given the TBox $\mathcal{T} \cup \{A \sqsubseteq C, D \sqsubseteq B\}$, where A, B are concept names that do not appear in C, D or $\mathcal{T}$. Thus, we obtain

Theorem 23 *In $\mathcal{EL}$, $\mathcal{IQ}$-query emptiness and $\mathcal{CQ}$-query emptiness are* PTIME*-hard.*

Observe that by Lemma 7 and materializability of $\mathcal{EL}$ we obtain that $\mathcal{CQ}$-predicate emptiness is PTIME-complete as well in $\mathcal{EL}$.

Note that we need very little for the proof of Lemma 21 to go through: it suffices that the total Σ-ABox $\mathcal{A}_\Sigma$ is satisfiable with every TBox. We can thus reduce emptiness to query answering over the total Σ-ABox in any extension of $\mathcal{EL}$ that is unable to express contradictions. As another important example, we consider $\mathcal{ELI}$. Since CQ evaluation in $\mathcal{ELI}$ is EXPTIME-complete, we only obtain an EXPTIME upper bound in this case. A matching lower bound is obtained from the EXPTIME-hardness of subsumption in $\mathcal{ELI}$ and the simple reduction of subsumption to $\mathcal{IQ}$-query emptiness given above.

Theorem 24 *In $\mathcal{ELI}$, $\mathcal{IQ}$-query emptiness and $\mathcal{CQ}$-query emptiness are* EXPTIME*-complete.*

It follows from Lemma 7 and materializability of $\mathcal{ELI}$ that $\mathcal{CQ}$-predicate emptiness is EXPTIME-complete in $\mathcal{ELI}$.

5.2 Horn Extensions Involving Negation or Functionality

The simplest extension of $\mathcal{EL}$ that can express unsatisfiability is $\mathcal{EL}_\bot$. We begin by showing that $\mathcal{IQ}$-emptiness in $\mathcal{EL}_\bot$ is EXPTIME-hard, and thus significantly harder than subsumption and instance checking (both of which can be decided in polynomial time). To this end, we first show that to decide $\mathcal{IQ}$-query emptiness in $\mathcal{EL}_\bot$ it is sufficient to consider emptiness w.r.t. directed tree-shaped ABoxes, where an ABox $\mathcal{A}$ is called *directed tree-shaped* if the following conditions hold:

1. the directed graph $G^d_\mathcal{A} = (\mathsf{Ind}(\mathcal{A}), \{(a,b) \mid r(a,b) \in \mathcal{A}\})$ is a tree;
2. for all $a, b \in \mathsf{Ind}(\mathcal{A})$, there is at most one role name r such that $r(a,b) \in \mathcal{A}$ or $r(b,a) \in \mathcal{A}$ (and only one of these is the case).

Proposition 25 *An instance query $B(v)$ is non-empty for a signature Σ given an $\mathcal{EL}_\bot$-TBox $\mathcal{T}$ iff there exists a directed tree-shaped Σ-ABox $\mathcal{A}$ that is satisfiable w.r.t. $\mathcal{T}$ such that $\mathcal{T}, \mathcal{A} \models B(a)$ for the root a of $\mathcal{A}$.*

Proof. We provide a sketch only since this result also follows from the more general Proposition 30 proved below. Assume $B(v)$ is non-empty for Σ given $\mathcal{T}$. We find a Σ-ABox $\mathcal{A}$ that is satisfiable w.r.t. $\mathcal{T}$ such that $\mathcal{T}, \mathcal{A} \models B(a)$. Let the potentially infinite ABox $\mathcal{A}^*$ be obtained by unfolding $\mathcal{A}$ as follows: the individuals of $\mathcal{A}^*$ are the words $a_0 r_0 \cdots r_{n-1} a_n$ such that $a_0 = a$ and $r_i(a_i, a_{i+1}) \in \mathcal{A}$ for all $0 \leq i < n$; then include $A(a_0 r_0 \cdots r_{n-1} a_n)$ in $\mathcal{A}^*$ iff $A(a_n) \in \mathcal{A}$ and include $r(a_0 r_0 \cdots a_n, a_0 r_0 \cdots a_n r_n a_{n+1})$ in $\mathcal{A}^*$ if $r_n(a_n, a_{n+1}) \in \mathcal{A}$. One can show that $\mathcal{A}^*$ is satisfiable w.r.t. $\mathcal{T}$ since $\mathcal{A}$ is, and that $\mathcal{T}, \mathcal{A} \models B(a)$ iff $\mathcal{T}, \mathcal{A}^* \models B(a)$. By compactness of first-order consequence, we obtain a finite ABox $\mathcal{A}' \subseteq \mathcal{A}^*$ with $\mathcal{T}, \mathcal{A}' \models B(a)$. $\mathcal{A}'$ is as required. ❑

Theorem 26 *In $\mathcal{EL}_\bot$, $\mathcal{IQ}$-query emptiness is* EXPTIME*-hard.*

Proof. Let $\mathcal{T}$, Σ, and $B(v)$ be given. By Proposition 25, $B(v)$ is non-empty for Σ given $\mathcal{T}$ iff there exists a directed tree-shaped Σ-ABox $\mathcal{A}$ that is a witness for the non-emptiness of $B(v)$ for Σ given $\mathcal{T}$. Directed tree-shaped Σ-ABoxes can be viewed as $\mathcal{EL}$-concepts using symbols from Σ only, and vice versa. Thus, such a witness Σ-ABox exists iff there exists an $\mathcal{EL}$-concept C using symbols from Σ only such that C is satisfiable w.r.t. $\mathcal{T}$ and $\mathcal{T} \models C \sqsubseteq B$. Now the following can be established by carefully analyzing the reduction underlying Theorem 36 in (Lutz & Wolter, 2010): given an $\mathcal{EL}_\bot$-TBox $\mathcal{T}$, a signature Σ, and a concept name B, it is EXPTIME-hard to decide if there exists an $\mathcal{EL}$-concept C using symbols from Σ only such that C is satisfiable w.r.t. $\mathcal{T}$ and $\mathcal{T} \models C \sqsubseteq B$. This establishes EXPTIME-hardness of non-emptiness. Using the fact that EXPTIME = coEXPTIME, this hardness result transfers to $\mathcal{IQ}$-query emptiness. ❑

Observe that by Lemma 7 and materializability of $\mathcal{EL}_\bot$ we obtain that $\mathcal{CQ}$-predicate emptiness is EXPTIME-hard as well in $\mathcal{EL}_\bot$.

Instead of proving a matching EXPTIME upper bound only for emptiness in $\mathcal{EL}_\bot$, we do this for the expressive Horn DL Horn-$\mathcal{ALCIF}$, of which $\mathcal{EL}_\bot$ is a fragment. In fact, the rest of this section is devoted to the proof of the following theorem. It is interesting to contrast this result with the undecidability of emptiness in $\mathcal{ALCF}$.

Theorem 27 *In Horn-$\mathcal{ALCIF}$, $\mathcal{CQ}$-query emptiness is in* EXPTIME*.*

The strategy for the proof of Theorem 27 is as follows. We first exhibit a polynomial-time reduction from $\mathcal{CQ}$-query emptiness in Horn-$\mathcal{ALCIF}$ to $\mathcal{CQ}$-query emptiness in $\mathcal{ELIF}_\bot$. Then, we show that non-emptiness of a CQ q under an $\mathcal{ELIF}_\bot$-TBox is always witnessed by ABoxes of a certain, forest-like shape. We then consider canonical models of forest-shaped ABoxes (and the TBox under consideration), which can be constructed by a chase-like procedure and are a special kind of materialization (cf. Section 3), that is, the answers returned for this model are precisely the certain answers. A central observation is that matches of q in canonical models of forest-shaped ABoxes can be grouped into equivalence classes that are induced by certain splittings of q. We finally show how to construct, for each equivalence class, a tree automaton that decides the existence of a forest-shaped witness ABox whose canonical model admits a match of q that falls into that class. Throughout this proof, we generally impose the UNA.

We begin with the reduction to $\mathcal{CQ}$-query emptiness in $\mathcal{ELIF}_\bot$. In fact, the reduction even shows that it suffices to consider $\mathcal{ELIF}_\bot$-TBoxes that are in *normal form*, by which we mean that all CIs take one of the forms

$$A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B, \quad A \sqsubseteq \exists r.B, \quad \exists r.A \sqsubseteq B,$$

where $A, A_1, \ldots, A_n, B \in \mathsf{N_C} \cup \{\top, \bot\}$ and r is a role name or inverse role.

Proposition 28 *For every Horn-$\mathcal{ALCIF}$ TBox $\mathcal{T}$, ABox signature Σ, and CQ q, one can construct in polynomial time an $\mathcal{ELIF}_\bot$-TBox $\mathcal{T}'$ in normal form such that q is empty for Σ given $\mathcal{T}$ iff q is empty for Σ given $\mathcal{T}'$.*

The proof of Proposition 28 is standard and given in the appendix. In what follows, we assume that all $\mathcal{ELIF}_\bot$ TBoxes are in normal form.

We next define *canonical models*. Let $(\mathcal{T}, \mathcal{A})$ be an $\mathcal{ELIF}_\bot$ KB such that $\mathcal{A}$ is satisfiable w.r.t. $\mathcal{T}$. To construct the (typically infinite) canonical model $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ of $(\mathcal{T}, \mathcal{A})$, start with $\mathcal{A}$ viewed as an interpretation, that is: $\Delta^{\mathcal{I}_{\mathcal{T},\mathcal{A}}} = \mathsf{Ind}(\mathcal{A})$, $A^{\mathcal{I}_{\mathcal{T},\mathcal{A}}} = \{a \mid A(a) \in \mathcal{A}\}$, and $r^{\mathcal{I}_{\mathcal{T},\mathcal{A}}} = \{(a,b) \mid r(a,b) \in \mathcal{A}\}$. Then exhaustively apply the following completion rules:

1. If $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq A \in \mathcal{T}$ and $d \in A_i^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$ for $1 \leq i \leq n$, and $d \notin A^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$, then add d to $A^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$.

2. If $\exists r.A \sqsubseteq B \in \mathcal{T}$, $(d,e) \in r^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$, $e \in A^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$, and $d \notin B^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$, then add d to $B^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$;

3. If $A \sqsubseteq \exists r.B \in \mathcal{T}$, $d \in A^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$, and either $d \notin (\exists r.B)^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$ and $\mathsf{funct}(r) \notin \mathcal{T}$ or $d \notin (\exists r.\top)^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$, then add (d,e) to $r^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$ and $e^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$ to $\Delta^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$ and $B^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$, where e is a fresh element.

4. If $A \sqsubseteq \exists r.B \in \mathcal{T}$, $\mathsf{funct}(r) \in \mathcal{T}$, $d \in A^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$, $(d,e) \in r^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$, and $e \notin B^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$, then add e to $B^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$.

The construction can be rendered deterministic by using an ordering of the inclusions and domain elements to decide among different possible rule applications. For this reason, we may speak of *the* canonical model. We call a model $\mathcal{U}$ of $\mathcal{T}$ and $\mathcal{A}$ *universal* if there is a homomorphism from $\mathcal{U}$ to any model $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{A}$, that is, a function $h : \Delta^{\mathcal{U}} \to \Delta^{\mathcal{I}}$ such that $d \in A^{\mathcal{U}}$ implies $h(d) \in A^{\mathcal{I}}$, $(d,e) \in r^{\mathcal{U}}$ implies $(h(d), h(e)) \in r^{\mathcal{I}}$, and $h(a^{\mathcal{U}}) = a^{\mathcal{I}}$ for all $a \in \mathsf{Ind}(\mathcal{A})$. The most important

property of $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ is that it is universal.[4] In fact, the following is standard to prove and we omit details, see for example (Lutz & Wolter, 2012).

Lemma 29 *Let $\mathcal{T}$ be an $\mathcal{ELIF}_\bot$-TBox and $\mathcal{A}$ an ABox that is satisfiable w.r.t. $\mathcal{T}$. Then $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ is a universal model of $(\mathcal{T},\mathcal{A})$.*

Let $\mathcal{T}$ be an $\mathcal{ELIF}_\bot$ TBox and $\mathcal{A}$ a Σ-ABox that is satisfiable w.r.t. $\mathcal{T}$. It is an easy consequence of Lemma 29 that a Σ-ABox $\mathcal{A}$ is a witness for a CQ q being non-empty for Σ given $\mathcal{T}$ if and only if $\mathcal{A}$ is satisfiable w.r.t. $\mathcal{T}$ and there is a match of q in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$.

The next step in our proof of Theorem 27 is to establish a proposition that constrains the shape of ABoxes to be considered when deciding emptiness in $\mathcal{ELIF}_\bot$. Here and in what follows, an ABox $\mathcal{A}$ is called *tree-shaped* if

1. the undirected graph $G_\mathcal{A} = (\mathsf{Ind}(\mathcal{A}), \{\{a,b\} \mid r(a,b) \in \mathcal{A}\})$ is a tree;
2. for all $a, b \in \mathsf{Ind}(\mathcal{A})$, there is at most one role name r such that $r(a,b) \in \mathcal{A}$ or $r(b,a) \in \mathcal{A}$, and only one of these is the case.

When working with tree-shaped ABoxes, we often designate one of the individuals as the *root*. If the root of the tree-shaped ABox $\mathcal{A}$ has been fixed, then we use $\mathcal{A}|_a$ to denote the restriction of $\mathcal{A}$ to those individuals b whose unique path to the root in $G_\mathcal{A}$ contains a, and we call $b \in \mathsf{Ind}(\mathcal{A})$ an *r-successor* (resp. *r^-*-successor) of $a \in \mathsf{Ind}(\mathcal{A})$ if $r(a,b) \in \mathcal{A}|_a$ (resp. $r(b,a) \in \mathcal{A}|_a$). We will also consider (rooted) *tree-shaped interpretations* and *tree-shaped queries*, defined analogously to tree-shaped ABoxes.

A Σ-ABox $\mathcal{A}$ is *forest-shaped* if there are ABoxes $\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_k$ such that the following conditions are satisfied:

1. $\mathcal{A}$ is the union of $\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_k$;
2. $k \leq |\mathsf{Ind}(\mathcal{A}_0)|$;
3. for $1 \leq i < j \leq k$: $\mathsf{Ind}(\mathcal{A}_i) \cap \mathsf{Ind}(\mathcal{A}_j) = \emptyset$ and $|\mathsf{Ind}(\mathcal{A}_i) \cap \mathsf{Ind}(\mathcal{A}_0)| = 1$;
4. for $1 \leq i \leq k$: $\mathcal{A}_i$ is a tree-shaped ABox rooted at some individual in $\mathsf{Ind}(\mathcal{A}_0)$.

We call $\mathcal{A}_0$ the *root component* of $\mathcal{A}$ and $\mathcal{A}_1, \ldots, \mathcal{A}_k$ the *tree components*. The *width* of $\mathcal{A}$ if k. The *degree* of $\mathcal{A}$ is the smallest number n such that for every tree component $\mathcal{A}_i$ and every $a \in \mathsf{Ind}(\mathcal{A}_i)$, the number of assertions $r(a,b)$ and $r(b,a)$ in $\mathcal{A}_i$ is bounded by n. The following proposition clarifies the role of forest-shaped ABoxes as witnesses for non-emptiness.

Proposition 30 *Let $\mathcal{T}$ be an $\mathcal{ELIF}_\bot$-TBox, Σ an ABox signature, and q a CQ. If q is non-empty for Σ given $\mathcal{T}$, then this is witnessed by a Σ-ABox that is forest-shaped, has width at most $|q|$, and degree at most $|\mathcal{T}|$.*

4. For readers who wonder about the relationship between universal models and materializations as defined in Section 3, we remark that every universal model of a TBox $\mathcal{T}$ and ABox $\mathcal{A}$ is a materialization of $\mathcal{T}$ and $\mathcal{A}$. Conversely, if there is a materialization of $\mathcal{T}$ and $\mathcal{A}$, then there exists also a universal model of $\mathcal{T}$ and $\mathcal{A}$ (Lutz & Wolter, 2012).

Proposition 30 is proved in the appendix by taking a witness Σ-ABox $\mathcal{A}$, selecting a part of $\mathcal{A}$ of size $|q|$ that is identified by a match of q and that serves as the root component of the forest-shaped ABox, then unraveling $\mathcal{A}$ into a infinite ABox starting from the selected part, afterwards removing unnecessary individual names to obtain the desired degree, and finally applying compactness to make the resulting witness finite.

Clearly, we can assume w.l.o.g. that in forest-shaped witness ABoxes according to Proposition 30, the individual names used in the root component are taken from a fixed set Ind of cardinality $|q|$. We will make this assumption without further notice in what follows.

We next analyze matches in forest-shaped ABoxes, using a splitting of the query into components. These are similar to the splittings of queries used in Appendix B, but simpler. A *forest splitting* of a CQ q is a tuple $F = (q', q_0, q_1, \ldots, q_n, \nu)$ where q' can be obtained from q by identifying variables, $q_0, q_1, \ldots, q_n$ is a partition of the atoms of q', and $\nu : \mathsf{var}(q_0) \to \mathsf{Ind}$ such that the following conditions are satisfied

1. $q_1, \ldots, q_n$ are tree-shaped;
2. $\mathsf{var}(q_i) \cap \mathsf{var}(q_0) \leq 1$ for $1 \leq i \leq n$;
3. $\mathsf{var}(q_i) \cap \mathsf{var}(q_j) = \emptyset$ for $1 \leq i < j \leq n$.

Let $\mathcal{T}$ be an $\mathcal{ELIF}_\bot$-TBox, $\mathcal{A}$ a forest-shaped ABox with root component $\mathcal{A}_0$, and π a match of q in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$. Note that $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ consists of $\mathcal{A}$ extended with (potentially infinite) trees attached to ABox individuals that have been generated by the completion rules. Then π is *of type* $F = (q', q_0, q_1, \ldots, q_n, \nu)$ if q' can be obtained from q by identifying those variables that π sends to the same element, q_0 consists of the atoms in q' that π matches to the $\mathcal{A}_0$-part of $\mathcal{I}_{\mathcal{T},\mathcal{A}}$, $q_0, \ldots, q_n$ are the maximal connected components of $q' \setminus q_0$, and ν is the restriction of π to range Ind. Note that, no matter which match π we choose, the maximal connected components of $q' \setminus q_0$ *must* be tree-shaped because they match into a tree-shaped part of $\mathcal{I}_{\mathcal{T},\mathcal{A}}$, which consists of a tree component of $\mathcal{A}$ plus the attached trees generated by the completion rules. Thus every match π has a type and the following is immediate, where $\mathcal{W}_{\mathcal{T},q,F}$ denotes the set of forest-shaped Σ-ABoxes of width at most $|q|$ and degree at most $|\mathcal{T}|$ that admit a match of q which is of type F.

Lemma 31 *Let $\mathcal{T}$ be an $\mathcal{ELIF}_\bot$-TBox, Σ an ABox signature, and q a CQ. Then q is empty for Σ given $\mathcal{T}$ if and only if $\mathcal{W}_{\mathcal{T},q,F}$ is empty for every forest splitting F of q.*

From now on, let $\mathcal{T}$ be an $\mathcal{ELIF}_\bot$-TBox $\mathcal{T}$ in normal form, Σ an ABox signature, and q a CQ, and assume that we want to decide whether q is empty for Σ given $\mathcal{T}$. By Lemma 31, it suffices to check whether $\mathcal{W}_{\mathcal{T},q,F}$ is empty for every forest splitting F of q.

Note that defining the set $\mathcal{W}_{\mathcal{T},q,F}$ is possible only because the definition of a forest splitting does not refer to a particular ABox, which in turn is due to our use of the fixed set of individual names Ind for the root components of forest ABoxes. In fact, *first* quantifying over forest splittings as in Lemma 31 and *then* quantifying over forest-shaped ABoxes (when testing emptiness of some $\mathcal{W}_{\mathcal{T},q,F}$) is essential for obtaining a single exponential time upper bound. Since the number of forest splittings is single exponential in $|q|$, we obtain such a bound if we can test emptiness of each $\mathcal{W}_{\mathcal{T},q,F}$ in time single exponential in $|\mathcal{T}| + |q|$. We will achieve this by constructing, for each forest splitting F of q, a two-way alternating parity automaton on infinite trees (TWAPA) $\mathfrak{A}_F$ that accepts a non-empty language if and only if $\mathcal{W}_{\mathcal{T},q,F} \neq \emptyset$. Note that *infinite* trees are needed because

automata will take trees as input that represent not only a (finite) forest-shaped Σ-ABox $\mathcal{A}$, but also a (potentially infinite) model of $\mathcal{A}$ and $\mathcal{T}$.

We start by introducing the necessary background for TWAPAs. Let $\mathbb{N}$ denote the *positive* integers. A *tree* is a non-empty (and potentially infinite) set $T \subseteq \mathbb{N}^*$ closed under prefixes. The node ε is the *root* of T. We use standard concatenation on the words from $\mathbb{N}^*$ (nodes of trees) and, as a convention, take $x \cdot 0 = x$ and $(x \cdot i) \cdot -1 = x$ for all $x \in \mathbb{N}^*$ and $i \in \mathbb{N}$. Note that $\varepsilon \cdot -1$ is undefined. When $i \geq 1$, the node $x \cdot i$ is said to be a *child* of the node x, and x is called the *parent* of $x \cdot i$. We will slightly depart from the original definition (Vardi, 1998) of TWAPAs by working with trees which are not full, that is, we define an *m-ary tree* as a tree each of whose nodes has at most (rather than exactly) m children. W.l.o.g., we assume that all nodes in an m-ary tree are from $\{1, \ldots, m\}^*$. An *infinite path* P of T is a prefix-closed set $P \subseteq T$ such that for every $n \geq 0$, there is a unique $x \in P$ with $|x| = n$.

For any set X, we use $\mathcal{B}^+(X)$ to denote the set of all positive Boolean formulas over X, i.e., formulas built using conjunction and disjunction over the elements of X used as propositional variables, and where the special formulas true and false are allowed as well. For an alphabet Γ, a Γ-*labeled tree* is a pair (T, V) with T a tree and $V : T \to \Gamma$ a node labeling function.

Definition 32 (TWAPA) *A* two-way alternating parity automaton (TWAPA) on m-ary trees *is a tuple* $\mathfrak{A} = (S, \Gamma, \delta, s_0, F)$ *where S is a finite set of* states*, Γ is a finite alphabet, $\delta : S \times \Gamma \to \mathcal{B}^+(\mathsf{tran}(\mathfrak{A}))$ is the* transition function *with $\mathsf{tran}(\mathfrak{A}) = \{\langle i \rangle, [i] \mid i \in \{-1, 0, \ldots m\}\} \times S$ the set of* transitions *of $\mathfrak{A}$, $s_0 \in S$ is the* initial state*, and $F = (G_1, \ldots, G_k)$ is a sequence of subsets of S satisfying $G_1 \subseteq G_2 \subseteq \ldots \subseteq G_h = S$, called the* parity condition*.*

Intuitively, a transition $(\langle i \rangle, s)$ with $i > 0$ means that a copy of the automaton in state s is sent to the i-th successor of the current node, *which is then required to exist*; by contrast, the transition $([i], s)$ only sends a copy *if the i-th successor exists*. The transitions $(\langle i \rangle, s)$ and $([i], s)$ with $i \in \{-1, 0\}$ are interpreted similarly where -1 indicates sending a copy to the predecessor and 0 indicates sending a copy to the current node. Note that a transition $(\langle -1 \rangle, s)$ cannot be applied at the root.

Definition 33 (Run, Acceptance) *A* run *of a TWAPA $\mathfrak{A} = (S, \Gamma, \delta, s_0, F)$ on a Γ-labeled tree (T, V) is a $T \times S$-labeled tree (T_ρ, ρ) such that $\rho(\varepsilon) = (\varepsilon, s_0)$ and for all $y \in T_\rho$, $\rho(y) = (x, s)$ and $\delta(s, V(x)) = \varphi$ implies that there is a (possibly empty) set $\{(d_1, s_1), \ldots, (d_n, s_n)\} \subseteq \mathsf{tran}(\mathfrak{A})$ that satisfies φ and is such that for $1 \leq i \leq n$:*

1. *if $d_i = \langle j \rangle$, then $x \cdot j$ is defined, $x \cdot j \in T$, $y \cdot i \in T_\rho$, and $\rho(y \cdot i) = (x \cdot j, s_i)$.*
2. *if $d_i = [j]$ and $x \cdot j$ is defined and belongs to T, then $y \cdot i \in T_\rho$ and $\rho(y \cdot i) = (x \cdot j, s_i)$.*

Given an infinite path $P \subseteq T_\rho$, we denote by $\mathsf{inf}(P)$ the set of all states q such that there are infinitely many $y \in P$ such that $\rho(y)$ is of the form (d, q). We say that the run (T_ρ, ρ) is accepting *if for all infinite paths $P \subseteq T_\rho$, there exists an even k such that $\mathsf{inf}(P) \cap G_k \neq \emptyset$ and $\mathsf{inf}(P) \cap G_{k-1} = \emptyset$.*

A Γ-labeled tree (T, V) is accepted *by $\mathfrak{A}$ if there is an accepting run of $\mathfrak{A}$ on (T, V). We use $L(\mathfrak{A})$ to denote the set of all Γ-labeled trees accepted by $\mathfrak{A}$.*

We note that the original definition of TWAPAs (Vardi, 1998) only uses transitions of the form $(\langle i \rangle, q)$ with $i \in \{1, \ldots, m\}$, since $(\langle i \rangle, q)$ and $([i], q)$ coincide for full m-ary trees. It is easy to see that emptiness for our version of TWAPAs can be reduced in polynomial time to emptiness for

TWAPAs in the original definition since we can encode m-ary trees as full m-ary trees. It is shown in (Vardi, 1998) that the emptiness problem of TWAPAs is EXPTIME-complete. More precisely, there is an algorithm that, given a TWAPA $\mathfrak{A} = (S, \Gamma, \delta, s_0, F)$, decides whether $L(\mathfrak{A}) = \emptyset$ and runs in time exponential in the cardinality of S and polynomial in the cardinality of Γ and size of δ. We also remind the reader that given two TWAPAs $\mathfrak{A}_1$ and $\mathfrak{A}_2$ with $\mathfrak{A}_i = (S_i, \Gamma_i, \delta_i, s_{0,i}, F_i)$, it is very easy to construct (in polynomial time) a TWAPA $\mathfrak{A}$ such that $L(\mathfrak{A}) = L(\mathfrak{A}_1) \cap L(\mathfrak{A}_2)$ and $\mathfrak{A}$ has state set $S_1 \cup S_2$.

To make them accessible to TWAPAs, we encode forest-shaped Σ-ABoxes of width at most $|q|$ and degree at most $|\mathcal{T}|$ as m-ary trees, where $m = |q| \cdot |\mathcal{T}|$. As has already been mentioned, each such tree additionally encodes a model of the encoded ABox. We now explain the alphabet used and the shape of the trees in more detail. The root node is labeled with an element of the alphabet Σ_R that consists of all $\mathsf{sig}(\mathcal{T})$-ABoxes $\mathcal{A}$ such that (i) $\mathsf{Ind}(\mathcal{A}) \subseteq \mathsf{Ind}$, (ii) $r(a, b) \in \mathcal{A}$ implies $r \in \Sigma$, and (iii) $\mathcal{A}$ satisfies all functionality statements in $\mathcal{T}$. Let $\mathsf{sub}(\mathcal{T})$ denote the set of concepts that occur in $\mathcal{T}$ and their subconcepts. Non-root nodes are labeled with elements from the alphabet Σ_N that consists of all subsets

$$\Theta \subseteq (\mathsf{N_C} \cap \mathsf{sub}(\mathcal{T})) \uplus \{M\} \uplus \{r, r^- \mid r \in \mathsf{N_R} \text{ occurs in } \mathcal{T}\} \uplus \mathsf{Ind} \uplus \{A^* \mid B \sqsubseteq \exists r.A \in \mathcal{T}\}$$

such that Θ contains (i) exactly one role name or inverse role, (ii) at most one element of Ind, and (iii) either M and a role name or inverse role from Σ or exactly one element of the form A^* and, in the latter case, also A.

A tree $\langle T, \ell \rangle$ with ℓ a labeling as described above is supposed to represent a forest-shaped Σ-ABox $\mathcal{A}_{\langle T,\ell \rangle}$ together with a model $\mathcal{I}_{\langle T,\ell \rangle}$ of this ABox and of $\mathcal{T}$. The individuals of $\mathcal{A}_{\langle T,\ell \rangle}$ are those in the ABox that labels the root of T, plus all non-root nodes of T whose label contains the marker M. All other nodes of T denote domain elements of $\mathcal{I}_{\langle T,\ell \rangle}$ that are not identified by any ABox individual. Both the assertions in $\mathcal{A}_{\langle T,\ell \rangle}$ and the concept and role memberships in $\mathcal{I}_{\langle T,\ell \rangle}$ are represented by the labels of $\langle T, \ell \rangle$. Note that the ABox $\mathcal{A}$ is a $\mathsf{sig}(\mathcal{T})$-ABox whereas $\mathcal{A}_{\langle T,\ell \rangle}$ uses signature Σ. In fact, only the Σ-assertions in $\mathcal{A}$ will be part of $\mathcal{A}_{\langle T,\ell \rangle}$ while *all* assertions in $\mathcal{A}$ will be part of $\mathcal{I}_{\langle T,\ell \rangle}$.

We need to impose some additional conditions to make sure that a $\Sigma_R \cup \Sigma_N$-labeled tree $\langle T, \ell \rangle$ indeed represents an ABox and model as intended. We call $\langle T, \ell \rangle$ *proper* if it satisfies the following conditions for all $x \in T$:

1. the root is labeled with an element $\mathcal{A}$ of Σ_R and all other nodes with an element of Σ_N;
2. $\ell(x) \in \Sigma_N$ contains an element of $\mathsf{Ind}(\mathcal{A})$ if x is a child of the root of T, and no element of Ind otherwise;
3. if we take any path in T and remove the root node (because it carries a special label), then the nodes whose label contains M form a finite (possibly empty) prefix of the resulting path;
4. if y is a child of x and $A^* \in \ell(y) \in \Sigma_N$, then there is some $B \sqsubseteq \exists r.A \in \mathcal{T}$ such that one of the following is true:
   (a) x is not the root, $B \in \ell(x)$ and $r \in \ell(y)$;
   (b) x is the root and for some $a \in \mathsf{Ind}$, $B(a) \in \ell(x)$ and $\{a, r\} \subseteq \ell(y)$.

The roles and individual names in element labels describe how these elements are connected to other elements via roles in $\mathcal{A}_{\langle T,\ell\rangle}$ and in $\mathcal{I}_{\langle T,\ell\rangle}$. In particular, if a successor of the root contains both the role r and the individual name $a \in \mathsf{Ind}$, then that node represents an r-successor of a. The label elements that are of the form A^* serve a special marking purpose: if $A^* \in \ell(x)$, then this means that the element x (which is part of $\mathcal{I}_{\langle T,\ell\rangle}$ but not of $\mathcal{A}_{\langle T,\ell\rangle}$ since $\ell(x)$ cannot contain M) is there to satisfy some concept inclusion $B \sqsubseteq \exists r.A$. We will later need these special markers to make sure that $\mathcal{I}_{\langle T,\ell\rangle}$ is not just *some* model of $\mathcal{A}_{\langle T,\ell\rangle}$, but a materialization of $\mathcal{A}_{\langle T,\ell\rangle}$ and $\mathcal{T}$. With these explanations and the subsequent definitions, the three conditions imposed on the elements of Θ and the four conditions used to define properness above should make sense to the reader.

Let $\langle T, \ell\rangle$ be a proper $\Sigma_R \cup \Sigma_N$-labeled tree. We now define $\mathcal{A}_{\langle T,\ell\rangle}$ and $\mathcal{I}_{\langle T,\ell\rangle}$ formally. Let $\mathcal{A}$ be the ABox that labels the root ε of T, and let $\mathcal{A}_\Sigma$ be the restriction of $\mathcal{A}$ to signature Σ. Then the Σ-ABox $\mathcal{A}_{\langle T,\ell\rangle}$ described by $\langle T, \ell\rangle$ is

$$\begin{aligned}\mathcal{A}_{\langle T,\ell\rangle} \;=\;& \mathcal{A}_\Sigma \cup \{A(x) \mid A \in \ell(x) \cap \mathsf{N_C} \cap \Sigma \text{ and } M \in \ell(x)\}\\ &\cup \{r(b,x) \mid \{b, r, M\} \subseteq \ell(x)\} \cup \{r(x,b) \mid \{b, r^-, M\} \subseteq \ell(x)\}\\ &\cup \{r(x,y) \mid y \text{ is a child of } x \text{ and } M \in \ell(x) \cap \ell(y) \text{ and } r \in \ell(y)\}\\ &\cup \{r(y,x) \mid y \text{ is a child of } x \text{ and } M \in \ell(x) \cap \ell(y) \text{ and } r^- \in \ell(y)\}\end{aligned}$$

and the interpretation $\mathcal{I}_{\langle T,\ell\rangle}$ is as follows:

$$\begin{aligned}\Delta^{\mathcal{I}_{\langle T,\ell\rangle}} &= (T \setminus \{\varepsilon\}) \cup \mathsf{Ind}(\mathcal{A})\\ A^{\mathcal{I}_{\langle T,\ell\rangle}} &= \{a \mid A(a) \in \mathcal{A}\} \cup \{x \mid A \in \ell(x) \cap \mathsf{N_C}\}\\ r^{\mathcal{I}_{\langle T,\ell\rangle}} &= \{(a,b) \mid r(a,b) \in \mathcal{A}\} \cup \{(a,x) \mid \{a, r\} \subseteq \ell(x)\} \cup \{(x,a) \mid \{a, r^-\} \subseteq \ell(x)\}\\ &\quad \cup \{(x,y) \mid y \text{ is a child of } x \text{ and } r \in \ell(y)\} \cup \{(x,y) \mid x \text{ is a child of } y \text{ and } r^- \in \ell(x)\}\\ c^{\mathcal{I}_{\langle T,\ell\rangle}} &= c \qquad \text{for all } c \in \mathsf{Ind}(\mathcal{A}_{\langle T,\ell\rangle})\end{aligned}$$

Apart from being represented as an ABox instead of as an interpretation, $\mathcal{A}_{\langle T,\ell\rangle}$ is identical to the restriction of $\mathcal{I}_{\langle T,\ell\rangle}$ to the individuals in $\mathcal{A}_{\langle T,\ell\rangle}$ and symbols in Σ, which in particular means that $\mathcal{I}_{\langle T,\ell\rangle}$ is a model of $\mathcal{A}_{\langle T,\ell\rangle}$. Note that the ABox $\mathcal{A}_{\langle T,\ell\rangle}$ is a forest-shaped Σ-ABox. Conversely, for any forest-shaped Σ-ABox of width at most $|q|$ and degree at most $|\mathcal{T}|$, we can define a proper m-ary $\Sigma_R \cup \Sigma_N$-labeled trees $\langle T, \ell\rangle$ such that $\mathcal{A}_{\langle T,\ell\rangle} = \mathcal{A}$ and $\mathcal{I}_{\langle T,\ell\rangle} = \mathcal{I}_{\mathcal{T},\mathcal{A}}$.

Let $F = (q', q_0, q_1, \ldots, q_n, \nu)$ be a splitting for q. We now build a TWAPA $\mathfrak{A}_F$ over m-ary $\Sigma_R \cup \Sigma_N$-labeled trees that accepts exactly those trees $\langle T, \ell\rangle$ such that $\mathcal{A}_{\langle T,\ell\rangle} \in \mathcal{W}_{\mathcal{T},q,F}$. The number of states of $\mathfrak{A}_F$ will be polynomial in $|\mathcal{T}| + |q|$ and since it can be checked in time single-exponential in the number of states whether $L(\mathfrak{A}_F) = \emptyset$, we obtain the desired ExpTime upper bound for deciding whether $\mathcal{W}_{\mathcal{T},q,F} = \emptyset$. We construct $\mathfrak{A}_F$ as the intersection of the following TWAPAs:

1. $\mathfrak{A}_{\mathsf{prop}}$, which makes sure that the input tree is proper;
2. $\mathfrak{A}_{\mathcal{T}}$, which ensures that the input tree $\langle T, \ell\rangle$ is such that $\mathcal{I}_{\langle T,\ell\rangle}$ is a model of $\mathcal{T}$;
3. $\mathfrak{A}_{\mathsf{wf}}$ which ensures that $\langle T, \ell\rangle$ satisfies a certain well-foundedness condition;
4. $\mathfrak{A}_{\mathsf{match}}$ which guarantees, exploiting the conditions ensured by the previous automata, that the input tree $\langle T, \ell\rangle$ is such that $\mathcal{A}_{\langle T,\ell\rangle} \in \mathcal{W}_{\mathcal{T},q,F}$.

The construction of the first automaton $\mathfrak{A}_{\mathsf{prop}}$ is straightforward, and details are left to the reader. Note that, to enforce Condition 3 of proper trees, the automaton needs to make use of the parity acceptance condition (a co-Büchi condition would actually be sufficient). The second TWAPA $\mathfrak{A}_{\mathcal{T}}$ ensures that the following conditions are satisfied for all non-root nodes x, x' of the input tree:

- if $r(a,b) \in \ell(\varepsilon)$ and $\mathsf{funct}(r) \in \mathcal{T}$, then $\{a, r\} \not\subseteq \ell(x)$;
- if $\mathsf{funct}(r) \in \mathcal{T}$ and $\{a, r\} \in \ell(x) \cap \ell(x')$, then $x = x'$;
- if $\mathsf{funct}(r) \in \mathcal{T}$, then x has at most one child y with $r \in \ell(y)$, and if additionally $r^- \in \ell(x)$, then there is no such child y;
- if $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq A \in \mathcal{T}$ and $\{A_1(a), \dots, A_n(a)\} \subseteq \ell(\varepsilon)$, then $A(a) \in \ell(\varepsilon)$;
- if $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq A \in \mathcal{T}$ and $\{A_1, \dots, A_n\} \subseteq \ell(x)$, then $A \in \ell(x)$;
- if $A \sqsubseteq \exists r.B \in \mathcal{T}$ and $A(a) \in \ell(\varepsilon)$, then (i) there is some b such that $\{r(a,b), B(b)\} \subseteq \ell(\varepsilon)$, or (ii) there is a child x of the root such that $\{a, r, B\} \subseteq \ell(x)$;
- if $A \sqsubseteq \exists r.B \in \mathcal{T}$ and $A \in \ell(x)$, then (i) $\{a, r^-\} \subseteq \ell(x)$ and $B(a) \in \ell(\varepsilon)$, (ii) $r^- \in \ell(x)$ and x has non-root parent y with $B \in \ell(y)$, or (iii) x has a child y with $\{r, B\} \subseteq \ell(y)$;
- if $\exists r.A \sqsubseteq B \in \mathcal{T}$ and (i) $\{r(a,b), A(b)\} \subseteq \ell(\varepsilon)$ or (ii) there is some child y of the root such that $\{a, r, A\} \subseteq \ell(y)$, then $B(a) \in \ell(\varepsilon)$;
- if $\exists r.A \sqsubseteq B \in \mathcal{T}$ and (i) $\{a, r^-\} \subseteq \ell(x)$ and $A(a)$ is in the label of the root, (ii) $r^- \in \ell(x)$ and x has parent y with $A \in \ell(y)$, or (iii) x has child y with $\{r, A\} \subseteq \ell(y)$, then $B \in \ell(x)$.

Working out the exact details of $\mathfrak{A}_{\mathsf{prop}}$ is again left to the reader.

Ideally, we would like the third automaton $\mathfrak{A}_{\mathsf{wf}}$ to ensure that $\mathcal{I}_{\langle T,\ell\rangle}$ is the canonical model of $\mathcal{T}$ and $\mathcal{A}_{\langle T,\ell\rangle}$. However, this does not seem to be easily possible because that model is constructed by applying completion rules in a certain order which is difficult to simulate by an automaton—note that applying the rules in a different order might result in the construction of an interpretation that is not isomorphic to the one obtained when following the prescribed application order. We thus define $\mathfrak{A}_{\mathsf{wf}}$ to achieve only the crucial property of canonical models that all 'positive information' (concept and role memberships of domain elements) is there for a reason, namely because it is contained in $\mathcal{A}_{\langle T,\ell\rangle}$ or because it is logically implied by $\mathcal{A}_{\langle T,\ell\rangle}$ together with $\mathcal{T}$. We formalize this in terms of derivations.

Let $\langle T, \ell\rangle$ be a proper $\Sigma_R \cup \Sigma_N$-labeled tree, $A_0 \in (\mathsf{N_C} \cap \mathsf{sub}(\mathcal{T})) \cup \{\top\}$, and $x_0 \in A_0^{\mathcal{I}_{\langle T,\ell\rangle}}$. A *derivation* of A_0 at x_0 in $\langle T, \ell\rangle$ is a finite L-labeled tree $\langle T', \ell'\rangle$, where L is the set of pairs (A, x) with $A \in (\mathsf{N_C} \cap \mathsf{sub}(\mathcal{T})) \cup \{\top\}$ and $x \in A^{\mathcal{I}_{\langle T,\ell\rangle}}$. We require that the root of T' is labeled with (A_0, x_0) and that T' is minimal such that for all nodes z of T' with $\ell'(z) = (A, x)$, one of the following holds:

1. $A \in \Sigma \cup \{\top\}$ and $x \in \mathsf{Ind}(\mathcal{A}_{\langle T,\ell\rangle})$;
2. $A^* \notin \ell(x)$ and there are a CI $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq A \in \mathcal{T}$ and children $z_1, \dots, z_n$ of z in T' such that $\ell'(z_i) = (A_i, x)$ for $1 \leq i \leq n$;

3. $A^* \notin \ell(x)$ and there is a CI $\exists r.A' \sqsubseteq A \in \mathcal{T}$ and a child z' of z in T' with $\ell'(z') = (A', x')$ such that $(x, x') \in r^{\mathcal{I}_{\langle T,\ell\rangle}}$. Moreover, if $B^* \in \ell(x)$, then there is a child z'' of z in T' with $\ell'(z'') = (B, x)$.

4. $A^* \notin \ell(x)$ and there is a CI $A' \sqsubseteq \exists r.A \in \mathcal{T}$ with $\mathsf{funct}(r) \in \mathcal{T}$ and a child z' of z in T' with $\ell'(z') = (A', x')$ such that $(x', x) \in r^{\mathcal{I}_{\langle T,\ell\rangle}}$. Moreover, if $B^* \in \ell(x)$, then there is a child z'' of z in T' with $\ell'(z'') = (B, x)$.

5. $A = \top$, $\top^* \notin \ell(x)$, $B^* \in \ell(x)$, and there is a child z' of z in T' with $\ell'(z') = (B, x)$.

6. $A^* \in \ell(x)$, there is a CI $A' \sqsubseteq \exists r.A \in \mathcal{T}$, and there is a child z' of z in T' with $\ell'(z') = (A', x')$ such that $(x', x) \in r^{\mathcal{I}_{\langle T,\ell\rangle}}$ and either (i) x is a child of x' in T, or (ii) x is a child of the root, $x' \in \mathsf{Ind}$, and $\{r, x'\} \subseteq \ell(x)$.

We say that $\langle T, \ell\rangle$ is *well-founded* if whenever $x \in A^{\mathcal{I}_{\langle T,\ell\rangle}}$, with $A \in \mathsf{N_C} \cup \{\top\}$, then there is a derivation of A at x in $\langle T, \ell\rangle$. It is not hard to construct a TWAPA $\mathfrak{A}_{\mathsf{wf}}$ that accepts precisely the well-founded proper $\Sigma_R \cup \Sigma_N$-labeled trees; essentially, the automaton can verify the existence of all required derivations by implementing the Conditions 1 to 6 above as transitions, additionally using a co-Büchi condition to ensure finiteness of the derivation.

Next let $F = (q', q_0, q_1, \ldots, q_n, \nu)$. The automaton $\mathfrak{A}_{\mathsf{match}}$ checks that

1. ν is a match for q_0 in $\mathcal{I}_{\langle T,\ell\rangle}$ and

2. there is a match π for q_i in $\mathcal{I}_{\langle T,\ell\rangle}$ such that if $v \in \mathsf{var}(q_0) \cap \mathsf{var}(q_i)$, then $\pi(v) = \nu(v)$.

$\mathfrak{A}_{\mathsf{match}}$ is easy to construct and we once more omit details. As announced, we define $\mathfrak{A}_F$ such that it accepts the intersection of the languages accepted by $\mathfrak{A}_{\mathsf{prop}}$, $\mathfrak{A}_{\mathcal{T}}$, $\mathfrak{A}_{\mathsf{wf}}$, and $\mathfrak{A}_{\mathsf{match}}$. It remains to show that $\mathcal{W}_{\mathcal{T},q,F}$ is empty iff $L(\mathfrak{A}_F)$ is empty.

To do this, we first clarify the relation between well-foundedness, canonical models, and universal models. We call a proper $\Sigma_R \cup \Sigma_N$-labeled tree $\langle T, \ell\rangle$ *canonical* if (i) $\mathcal{I}_{\langle T,\ell\rangle}$ is the canonical model of $\mathcal{A}_{\langle T,\ell\rangle}$ and $\mathcal{T}$, and (ii) for every $x \in T \setminus \{\varepsilon\}$ with $M \notin \ell(x)$, the concept $A^* \in \ell(x)$ is such that the element x was created due to an application of the third completion rule to an inclusion of the form $B \sqsubseteq \exists r.A$ and the parent of x in T.

Lemma 34

1. *If a proper $\Sigma_R \cup \Sigma_N$-labeled tree is canonical, then it is well-founded;*

2. *If $\langle T, \ell\rangle$ is a proper $\Sigma_R \cup \Sigma_N$-labeled tree that is well-founded and $\mathcal{I}_{\langle T,\ell\rangle}$ is a model of $\mathcal{T}$, then $\mathcal{I}_{\langle T,\ell\rangle}$ is a universal model of $\mathcal{T}$ and $\mathcal{A}_{\langle T,\ell\rangle}$.*

A proof of Lemma 34 can be found in the appendix. Point 1 is established by tracing the applications of the completion rules applied to construct the canonical model of $\mathcal{A}_{\langle T,\ell\rangle}$ and $\mathcal{T}$ and showing that each addition that they make gives rise to a derivation. For Point 2, we first show that one can make a certain uniformity assumption on derivations and then show how to define a homomorphism from $\mathcal{I}_{\langle T,\ell\rangle}$ to any model of $\mathcal{A}_{\langle T,\ell\rangle}$ and $\mathcal{T}$ by starting at the part of $\mathcal{I}_{\langle T,\ell\rangle}$ that corresponds to the root component of $\mathcal{A}_{\langle T,\ell\rangle}$ and then moving downwards along the tree-shaped parts of $\mathcal{I}_{\langle T,\ell\rangle}$.

Lemma 35 *$\mathcal{W}_{\mathcal{T},q,F} = \emptyset$ iff $L(\mathfrak{A}_F) = \emptyset$.*

Proof. First assume that $\mathcal{W}_{\mathcal{T},q,F} \neq \emptyset$. Then there is a forest-shaped Σ-ABox $\mathcal{A}$ of width at most $|\mathcal{T}|$ and degree at most $|q|$ that is satisfiable w.r.t. $\mathcal{T}$ and a match π of q in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ that is of type F. Let $\langle T, \ell\rangle$ be an m-ary proper $\Sigma_R \cup \Sigma_N$-labeled tree that satisfies $\mathcal{A}_{\langle T,\ell\rangle} = \mathcal{A}$ and is canonical. Then $\langle T, \ell\rangle \in L(\mathfrak{A}_{\mathsf{prop}})$. Since $\mathcal{A}$ is satisfiable w.r.t. $\mathcal{T}$, $\mathcal{I}_{\langle T,\ell\rangle} = \mathcal{I}_{\mathcal{T},\mathcal{A}}$ is a model of $\mathcal{T}$ and thus $\langle T, \ell\rangle \in L(\mathfrak{A}_{\mathsf{prop}})$. By Point 1 of Lemma 34, $\langle T, \ell\rangle \in L(\mathfrak{A}_{\mathsf{wf}})$. Finally, the match π witnesses that Conditions 1 and 2 from the definition of $\mathfrak{A}_{\mathsf{match}}$ are satisfied and thus $\langle T, \ell\rangle \in L(\mathfrak{A}_{\mathsf{match}})$ and we are done.

Conversely, assume that there is a tree $\langle T, \ell\rangle \in L(\mathfrak{A}_F)$. Since $\langle T, \ell\rangle \in L(\mathfrak{A}_{\mathsf{prop}})$, $\mathcal{A} = \mathcal{A}_{\langle T,\ell\rangle}$ is defined; by definition, it is a forest-shaped Σ-ABox of width at most $|\mathcal{T}|$ and degree at most $|q|$. It remains to show that there is a match π of q in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ that is of type F. Since $\langle T, \ell\rangle \in L(\mathfrak{A}_{\mathcal{T}}) \cap L(\mathfrak{A}_{\mathsf{wf}})$, $\mathcal{I}_{\langle T,\ell\rangle}$ is a model of $\mathcal{T}$ and $\langle T, \ell\rangle$ is well-founded. By Point 2 of Lemma 34, $\mathcal{I}_{\langle T,\ell\rangle}$ is thus a universal model of $\mathcal{T}$ and $\mathcal{A}$. Because $\langle T, \ell\rangle \in L(\mathfrak{A}_{\mathsf{match}})$, Conditions 1 and 2 from the definition of $\mathfrak{A}_{\mathsf{match}}$ are satisfied. It can be verified that, consequently, there is a match π of q in $\mathcal{I}_{\langle T,\ell\rangle}$ that is of type F. Composing π with the homomorphism from $\mathcal{I}_{\langle T,\ell\rangle}$ to $\mathcal{I}_{\mathcal{T},\mathcal{A}}$, which exists since $\mathcal{I}_{\langle T,\ell\rangle}$ is universal, yields a match of q in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ that is of type F. ❏

6. The DL-Lite Family

We study query and predicate emptiness in the DL-Lite family of description logics (Calvanese et al., 2007; Artale et al., 2009). To begin with, we introduce the dialects of DL-Lite we consider. *Basic concepts* B are defined by

$$B \quad ::= \quad \top \quad | \quad A \quad | \quad \exists r.\top$$

where A ranges over $\mathsf{N_C}$ and r over all (possibly inverse) roles. A DL-Lite$_{core}$ TBox $\mathcal{T}$ is a finite set of CIs of the form $B_1 \sqsubseteq B_2$ and $B_1 \sqcap B_2 \sqsubseteq \bot$, where B_1 and B_2 are basic concepts. Thus, DL-Lite$_{core}$ is included in $\mathcal{ELI}$ but, because it includes inverse roles, not included in $\mathcal{EL}$. DL-Lite$_{\mathcal{F}}$ is the extension of DL-Lite$_{core}$ with functionality statements. DL-Lite$_{\mathcal{R}}$ is the extension of DL-Lite$_{core}$ with role inclusions $r \sqsubseteq s$, where r, s are roles. DL-Lite$_{\mathcal{R}}$ is the logical underpinning of the OWL profile OWL2 QL (Motik et al., 2009). Finally, DL-Lite$_{horn}$ is the extension of DL-Lite$_{core}$ with conjunctions of basic concepts on the left hand side of CIs. Alternatively, it can be defined as the fragment of $\mathcal{ELI}_{\bot}$ with qualified existential restrictions $\exists r.C$ replaced by unqualified existential restrictions $\exists r.\top$. For further details, we refer readers to (Calvanese et al., 2007; Artale et al., 2009; Calvanese, De Giacomo, Lembo, Lenzerini, & Rosati, 2013).

We briefly discuss the UNA for the DL-Lite dialects introduced above. First observe that DL-Lite$_{horn}$ and DL-Lite$_{\mathcal{F}}$ are fragments of $\mathcal{ALCIF}$. Thus, by Lemma 3, query emptiness and predicate emptiness for DL-Lite$_{horn}$ and DL-Lite$_{\mathcal{F}}$ are oblivious as to whether the UNA is made or not. DL-Lite$_{\mathcal{R}}$ is not a fragment of $\mathcal{ALCIF}$. It is, however, straightforward to show that for DL-Lite$_{\mathcal{R}}$ the certain answers to CQs do not depend on whether one adopts the UNA or not. Thus, also for DL-Lite$_{\mathcal{R}}$ query emptiness and predicate emptiness are oblivious as to whether the UNA is made or not. In the following proofs we make the UNA.

Our main results are as follows: $\mathcal{CQ}$-query emptiness is coNP-complete for all DL-Lite dialects. The coNP-lower bound holds already for the fragment of DL-Lite$_{core}$ without role names. By contrast, the complexity of deciding $\mathcal{IQ}$-query emptiness and $\mathcal{CQ}$-predicate emptiness depends on whether conjunctions are admitted on the left-hand side of concept inclusions or not. If no

conjunctions are admitted (as in DL-Lite$_{core}$, DL-Lite$_{\mathcal{R}}$, and DL-Lite$_{\mathcal{F}}$), then $\mathcal{IQ}$-query emptiness and $\mathcal{CQ}$-predicate emptiness are NLOGSPACE-complete. If conjunctions are admitted (as in DL-Lite$_{horn}$), then both $\mathcal{IQ}$-query emptiness and $\mathcal{CQ}$-predicate emptiness are coNP-complete. Again, the lower bound holds already for the fragments of the DLs without role names.

We note that in what follows we do not use Theorem 6 which gives a polynomial reduction of $\mathcal{CQ}$-predicate emptiness to $\mathcal{IQ}$-query emptiness for certain DLs but does not apply to the DL-Lite dialects. Instead we give direct proofs. The results presented in Figure 1 for the DL-Lite dialects are straightforward consequences of the results established in this section.

We begin by proving the coNP lower bounds. Let $\mathcal{L}_{core}$ be the DL that admits only CIs $A \sqsubseteq B$ and $A \sqcap B \sqsubseteq \bot$, and let $\mathcal{L}_{horn}$ be the DL that admits only CIs $A \sqcap A' \sqsubseteq B$ and $A \sqcap B \sqsubseteq \bot$, where A, A', and B are concept names.

Theorem 36 *In $\mathcal{L}_{horn}$, $\mathcal{IQ}$-query emptiness, $\mathcal{CQ}$-query emptiness, and $\mathcal{CQ}$-predicate emptiness are coNP-hard. In $\mathcal{L}_{core}$, $\mathcal{CQ}$-query emptiness is coNP-hard.*

Proof. The proofs are by reduction from the well-known coNP-complete problem of testing whether a propositional formula in conjunctive normal form (CNF) is unsatisfiable. Let $\varphi = \psi_1 \wedge \cdots \wedge \psi_k$ be a CNF formula, $v_1, \ldots, v_n$ the variables used in φ, $A_{\psi_1}, \ldots, A_{\psi_k}$ concept names for representing clauses, and $A_{v_1}, A_{\neg v_1}, \ldots, A_{v_n}, A_{\neg v_n}$ concept names for representing literals. Let A be an additional concept name, and set $\Sigma = \{A_{v_1}, A_{\neg v_1}, \ldots, A_{v_n}, A_{\neg v_n}\}$. Consider the $\mathcal{L}_{horn}$-TBox $\mathcal{T}$ consisting of the following CIs:

- $A_{v_j} \sqcap A_{\neg v_j} \sqsubseteq \bot$ for all $1 \leq j \leq n$;
- $A_{\ell_j} \sqsubseteq A_{\psi_i}$ for all $1 \leq i \leq k$ and each $\ell_j = (\neg)v_j$ that is a disjunct of ψ_i;
- $A_{\psi_1} \sqcap \cdots \sqcap A_{\psi_k} \sqsubseteq A$.

It is straightforward to show that $A(u)$ is empty for Σ given $\mathcal{T}$ iff $\exists u\, A(u)$ is empty for Σ given $\mathcal{T}$ iff φ is unsatisfiable. Thus, deciding $\mathcal{IQ}$-query emptiness, $\mathcal{CQ}$-predicate emptiness, and $\mathcal{CQ}$-query emptiness in $\mathcal{L}_{horn}$ is coNP-hard. For the coNP-hardness result for $\mathcal{CQ}$-query emptiness in $\mathcal{L}_{core}$, we drop the last CI from $\mathcal{T}$ and use the CQ $A_{\psi_1}(u) \wedge \cdots \wedge A_{\psi_k}(u)$ instead. ❏

We now prove matching upper complexity bounds, considering the logics DL-Lite$_{core}$, DL-Lite$_{\mathcal{R}}$, DL-Lite$_{\mathcal{F}}$, and DL-Lite$_{horn}$. To this end, we formulate general sufficient conditions for deciding emptiness in PTIME and in coNP. We say that a DL $\mathcal{L}$ has the *polysize emptiness witness property* if whenever a CQ q is not empty for Σ given $\mathcal{T}$, then there exists a Σ-ABox $\mathcal{A}$ of polynomial size in the size of $\mathcal{T}$ and q that is satisfiable w.r.t. $\mathcal{T}$ and such that $\mathsf{cert}_{\mathcal{T},\mathcal{A}}(q) \neq \emptyset$.

Lemma 37 *Let $\mathcal{L}$ be any description logic with the polysize emptiness witness property such that the query evaluation problem for CQs for $\mathcal{L}$ is in NP. Moreover, assume that satisfiability of ABoxes w.r.t. $\mathcal{L}$-TBoxes can be decided in polynomial time. Then $\mathcal{CQ}$-query emptiness in $\mathcal{L}$ is in coNP.*

Proof. An NP-algorithm deciding whether a CQ q is not empty w.r.t. $\mathcal{T}$ and Σ guesses (i) a Σ-ABox $\mathcal{A}$ of polynomial size in $\mathcal{T}$ and q, (ii) a tuple $\vec{a}$ of individual names from $\mathsf{Ind}(\mathcal{A})$ of the appropriate length, and (iii) a polysize certificate that $\vec{a} \in \mathsf{cert}_{\mathcal{T},\mathcal{A}}(q)$; it then verifies in polynomial time that $\mathcal{A}$ is satisfiable w.r.t. $\mathcal{T}$ and the guessed certificate is valid. ❏

Theorem 38 *In DL-Lite$_{core}$, DL-Lite$_{\mathcal{R}}$, DL-Lite$_{\mathcal{F}}$, and DL-Lite$_{horn}$, deciding $\mathcal{CQ}$-query emptiness is in coNP.*

Proof. The conditions stated in Lemma 37 are either shown in or follow directly from (Calvanese et al., 2007; Artale et al., 2009). We sketch a proof of the polysize emptiness witness property. Assume $\vec{a} \in \mathsf{cert}_{\mathcal{T},\mathcal{A}}(q)$ for a CQ $q = \exists \vec{u}\, \varphi(\vec{v}, \vec{u})$ and a TBox $\mathcal{T}$ in any of the DLs listed in the theorem statement and further assume that $\mathcal{A}$ is satisfiable w.r.t. $\mathcal{T}$. We use the canonical model $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ from Lemma 29 (for DL-Lite$_{core}$, DL-Lite$_{\mathcal{F}}$, and DL-Lite$_{horn}$ this can be used without any modification since they are fragments of $\mathcal{ELIF}_\bot$; for DL-Lite$_{\mathcal{R}}$, one has to add the following completion rule for the construction of $\mathcal{I}_{\mathcal{T},\mathcal{A}}$: if $(x, y) \in r^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$ and $r \sqsubseteq s \in \mathcal{T}$, then add (x, y) to $s^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$). Let π be a match for q in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$. We recall that $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ consists of its restriction to the individuals $a^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$ with $a \in \mathsf{Ind}(\mathcal{A})$ and tree-shaped interpretations $\mathcal{I}_a$ attached to $a^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$. Let $\mathcal{A}'_0$ be the set of assertions in $\mathcal{A}$ that only use individual names $a \in \mathsf{Ind}(\mathcal{A})$ such that there exists $v \in \mathsf{var}(q)$ with $\pi(v) = a^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$ or $\pi(v) \in \Delta^{\mathcal{I}_a}$. Moreover, for any individual $a \in \mathsf{Ind}(\mathcal{A}'_0)$ selected above such that there exists a role r and b with $r(a, b) \in \mathcal{A}$, select one such $r(a, b^*)$ and include it in $\mathcal{A}'_1$. Let $\mathcal{A}' = \mathcal{A}'_0 \cup \mathcal{A}'_1$. Clearly the ABox $\mathcal{A}'$ is as required: it is of polynomial size, it is satisfiable w.r.t. $\mathcal{T}$ (being a subset of $\mathcal{A}$), and by construction, it satisfies $\vec{a} \in \mathsf{cert}_{\mathcal{T},\mathcal{A}'}(q)$. ❑

We say that a DL $\mathcal{L}$ has the *singleton emptiness witness property* if whenever a CQ q of the form $A(v)$ or $\exists v\, A(v)$ is not empty for Σ given $\mathcal{T}$, then there exists a Σ-ABox $\mathcal{A}$ containing at most one assertion which is satisfiable w.r.t. $\mathcal{T}$ and such that $\mathsf{cert}_{\mathcal{T},\mathcal{A}}(q) \neq \emptyset$.

Lemma 39 *Let $\mathcal{L}$ be any description logic with the singleton emptiness witness property such that the query evaluation problem for CQs of the form $A(v)$ and $\exists v\, A(v)$ for $\mathcal{L}$ is in* NLOGSPACE. *Moreover, assume that satisfiability of singleton ABoxes w.r.t. $\mathcal{L}$-TBoxes can be decided in* NLOGSPACE. *Then $\mathcal{IQ}$-query emptiness and $\mathcal{CQ}$-predicate emptiness in $\mathcal{L}$ are in* NLOGSPACE.

Proof. A non-deterministic logarithmic space algorithm deciding whether a CQ of the form $A(v)$ or $\exists v\, A(v)$ is not empty w.r.t. $\mathcal{T}$ iterates over all Σ-ABoxes $\mathcal{A}$ containing at most one assertion and checks whether at least one of those ABoxes $\mathcal{A}$ is satisfiable w.r.t. $\mathcal{T}$ and satisfies $\mathcal{T}, \mathcal{A} \models \exists v\, A(v)$ or, respectively, $\mathcal{T}, \mathcal{A} \models A(a)$, for an individual $a \in \mathsf{Ind}(\mathcal{A})$. ❑

Theorem 40 *In DL-Lite$_{core}$, DL-Lite$_{\mathcal{R}}$, and DL-Lite$_{\mathcal{F}}$, deciding $\mathcal{IQ}$-query emptiness and $\mathcal{CQ}$-predicate emptiness are* NLOGSPACE*-complete.*

Proof. For the NLOGSPACE-upper bound, the conditions stated in Lemma 39 are either shown in or follow directly from (Calvanese et al., 2007; Artale et al., 2009). We sketch a proof of the singleton emptiness witness property. Assume $\vec{a} \in \mathsf{cert}_{\mathcal{T},\mathcal{A}}(q)$ for a CQ q of the form $A(v)$ or $\exists v\, A(v)$ and a TBox $\mathcal{T}$ in any of the DLs listed in the theorem statement. Further assume that $\mathcal{A}$ is satisfiable w.r.t. $\mathcal{T}$. We consider the case $q = \exists v\, A(v)$; the case $q = A(v)$ is similar. As in the proof of Theorem 38, we use the canonical model $\mathcal{I}_{\mathcal{T},\mathcal{A}}$. Let π be a mapping into $\Delta^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$ such that $\pi(v) \in A^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$ and consider the uniquely determined $a \in \mathsf{Ind}(\mathcal{A})$ such that $\pi(v) = a^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$ or $\pi(v) \in \Delta^{\mathcal{I}_a}$. Using the fact that no conjunctions occur on the left-hand side of CIs in $\mathcal{T}$, one can show that there exists a single assertion of the form $B(a)$ or $r(a, b)$ such that for the ABox $\mathcal{A}'$ consisting of that assertion, we have $() \in \mathsf{cert}_{\mathcal{T},\mathcal{A}'}(q)$. It follows that $\mathcal{A}'$ is the desired witness ABox.

The matching lower bound follows directly from the fact that deciding whether $\mathcal{T} \models A \sqsubseteq B$ is NLOGSPACE-hard for TBoxes $\mathcal{T}$ in DL-Lite$_{core}$ (Artale et al., 2009). ❑

7. Case Study and Application to Modularity

We demonstrate the usefulness of emptiness in two ways. First, we carry out a case study for predicate emptiness in the medical domain, where we find that the use of a realistic ontology adds a significant number of non-empty predicates to the ABox signature while there is also a large number of predicates that are empty. In static analysis, it is thus potentially non-trivial for a user to manually distinguish the non-empty from the empty predicates. Second, we show that (predicate) emptiness can be used to produce a smaller version of a TBox $\mathcal{T}$ that is tailor-made for querying with a given ABox signature (in a sense: a module of the TBox). Replacing $\mathcal{T}$ with the potentially much smaller module facilitates comprehension of the TBox, thus helping with query formulation. We again support our claims by experiments.

In the case study, we use the comprehensive medical ontology SNOMED CT, which provides a systematic vocabulary used for medical information interchange and to enable interoperable electronic health records. It covers diverse medical areas such as clinical findings, symptoms, diagnoses, procedures, body structures, organisms, substances, pharmaceuticals, devices and specimens. SNOMED CT is formulated in $\mathcal{EL}$ extended with role inclusions (which we removed for the experiments). It contains about 370,000 concept names and 62 role names. We use SNOMED CT together with an ABox signature from a real-world application and with randomly generated ABox signatures. The real-world signature was obtained by analyzing clinical notes of the emergency department and the intensive care unit of two Australian hospitals, using natural language processing methods to detect SNOMED CT concepts and roles.[5] It contains 8,858 concepts and 16 roles. For this signature, 16,212 $\mathcal{IQ}$-non-empty predicates and 17,339 $\mathcal{CQ}$-non-empty predicates were computed. Thus, SNOMED CT provides a substantial number of additional predicates for query formulation, roughly identical to the number of predicates in the ABox signature. However, these numbers also show that the majority of predicates in SNOMED CT cannot meaningfully be used in queries over Σ-ABoxes, and thus identifying the relevant ones via predicate emptiness is potentially very helpful. Somewhat surprisingly, the number of $\mathcal{CQ}$-non-empty predicates is only about 10% higher than the number of $\mathcal{IQ}$-non-empty symbols.

We have analyzed randomly generated signatures that contain 500, 1,000, 5,000, and 10,000 concept names and 16 or 31 role names (1/2 and 1/4 of the role names in the ontology). Every signature contains the special role name role-group, which is used in SNOMED CT to implement a certain modeling pattern and should be present also in ABoxes to allow the same pattern there. For each number of concept and role names, we generated 10 signatures. The columns "$\mathcal{IQ}$ non-empty" and "$\mathcal{CQ}$ non-empty" of Figure 4 show the results, where the numbers are averages for the 10 experiments for each size. These additional experiments confirm the findings for our real-world signature: in each case, a substantial number of additional predicates becomes available for query formulation, but there is also a large number of predicates that are empty.

We now come to the application in modularity. Recall that our main motivation for studying emptiness is to support query formulation: as TBoxes can be large and complex, it can be difficult to understand whether a TBox contains sufficient background knowledge so that a given query q

5. See "Current Collaborative Projects" at http://sydney.edu.au/engineering/it/∼hitru.

concepts	roles	$\mathcal{IQ}$ non-empty	$\mathcal{CQ}$ non-empty	axioms $\bot$-mod.	axioms $\mathcal{CQ}_\Sigma$-core
500	16	3557	4631	8910	4597
500	31	3654	4734	8911	4696
1000	16	5827	7385	14110	7349
1000	31	6242	7762	14147	7731
5000	16	18330	21451	33469	21427
5000	31	18469	21557	33616	21532
10000	16	29519	33493	47044	33489
10000	31	30643	34645	47256	34637

Figure 4: Experimental Results

can have a non-empty answer over a Σ-ABox. If this is not the case, it clearly does not make sense to pose q to any Σ-ABox when this TBox is used as the background ontology. Similarly, it can be hard to find out whether a TBox is sufficiently powerful to entail that a given predicate can occur in some query that has a non-empty answer over some Σ-ABox. Again, if this is not the case, then that predicate should not be used when formulating queries. Here, we go one step further: instead of using emptiness directly to support query formulation, we use it to simplify the TBox. More precisely, we consider the problem of extracting a (hopefully small!) subset of a given TBox that gives exactly the same answers to all CQs (or IQs) for any Σ-ABox. Such a subset will be called a Σ-substitute w.r.t. $\mathcal{CQ}$ (or $\mathcal{IQ}$, respectively) of the original TBox and can replace the original TBox when answering CQs (or IQs, respectively). Working with a small Σ-substitute instead of the original TBox supports comprehension of the TBox and thereby the formulation of meaningful queries.

It is beyond the scope of this paper to investigate Σ-substitutes in depth. Instead, we show that, in the description logic $\mathcal{ELI}$, predicate emptiness gives rise to a particularly natural kind of Σ-substitute that we call the *$\mathcal{CQ}_\Sigma$-core*. The $\mathcal{CQ}_\Sigma$-core is obtained by removing all concept inclusions that contain a predicate which is $\mathcal{CQ}$-predicate empty for Σ w.r.t. the TBox. Thus, not only does the $\mathcal{CQ}_\Sigma$-core give the same answers to CQs as the original TBox for Σ-ABoxes, but it also has the appealing property that *all* predicates which occur in it can be used meaningfully in a CQ when querying Σ-ABoxes.

We also show that the widely known semantic $\bot$-modules introduced in (Grau et al., 2008) are Σ-substitutes and that $\mathcal{CQ}_\Sigma$-cores cannot be larger than semantic $\bot$-modules (unless the original ontology contains tautological concept inclusions). To evaluate the method in practice and compare the size of $\mathcal{CQ}_\Sigma$-cores and $\bot$-modules, we also extend our case study based on SNOMED CT to the extraction of $\mathcal{CQ}_\Sigma$-cores and their comparison with $\bot$-modules. We start by defining Σ-substitutes in a formal way.

Definition 41 *Let $\mathcal{T}' \subseteq \mathcal{T}$ and $\mathcal{Q} \in \{\mathcal{IQ}, \mathcal{CQ}\}$. Then $\mathcal{T}'$ is a* Σ-substitute for $\mathcal{T}$ w.r.t. $\mathcal{Q}$ *if for all Σ-ABoxes $\mathcal{A}$ and all $q \in \mathcal{Q}$, we have that* $\mathsf{cert}_{\mathcal{T}',\mathcal{A}}(q) = \mathsf{cert}_{\mathcal{T},\mathcal{A}}(q)$.

We are not aware that Σ-substitutes according to Definition 41 have been studied before, but they are closely related to other types of modules. For example, Σ-modules that give the same answers to all CQs formulated in signature Σ for all Σ-ABoxes are studied in (Lutz & Wolter, 2010; Kontchakov, Wolter, & Zakharyaschev, 2010; Konev, Ludwig, Walther, & Wolter, 2012; Botoeva, Kontchakov, Ryzhikov, Wolter, & Zakharyaschev, 2014, 2016; Romero, Kaminski, Grau, & Horrocks, 2015). A stronger version of a module is provided by Σ-modules that require the original TBox to be a model-conservative extension of the module regarding the signature Σ, as studied in (Konev, Lutz, Walther, & Wolter, 2013; Gatens, Konev, & Wolter, 2014). However, an important difference between all those Σ-modules and our Σ-substitutes is that the latter only restrict the signature of the ABox, but not of the queries. In contrast, the mentioned Σ-modules only guarantee the same answers to CQs formulated in signature Σ (and for Σ-ABoxes). In particular, it follows that minimal modules as defined in (Kontchakov et al., 2010; Konev et al., 2013) can in general not be used as a Σ-substitute.

We now show that in $\mathcal{ELI}$ (and, therefore, also in its fragment $\mathcal{EL}$) one can use $\mathcal{CQ}$-predicate emptiness in a straightforward way to compute a Σ-substitute w.r.t. $\mathcal{CQ}$. Let $\mathcal{T}$ be a TBox and Σ an ABox signature. The *$\mathcal{CQ}_\Sigma$-core of $\mathcal{T}$*, denoted $\mathcal{T}_\Sigma^{\mathcal{CQ}}$, is the set of all concept inclusions $\alpha \in \mathcal{T}$ such that no $X \in \mathsf{sig}(\alpha)$ is $\mathcal{CQ}$-predicate empty for Σ given $\mathcal{T}$.

Theorem 42 *Let $\mathcal{T}$ be a TBox in $\mathcal{ELI}$. Then the $\mathcal{CQ}_\Sigma$-core of $\mathcal{T}$ is a Σ-substitute for $\mathcal{T}$ w.r.t. $\mathcal{CQ}$ (and thus also w.r.t. $\mathcal{IQ}$).*

Proof. Let $\mathcal{T}'$ be the $\mathcal{CQ}_\Sigma$-core of $\mathcal{T}$ and assume that $\mathcal{T}', \mathcal{A} \not\models q[\vec{a}]$ for some Σ-ABox $\mathcal{A}$. Consider the canonical model $\mathcal{I}_{\mathcal{T}',\mathcal{A}}$, introduced in Section 5.2. Then $\mathcal{I}_{\mathcal{T}',\mathcal{A}}$ is a model of $\mathcal{T}'$ and $\mathcal{A}$, and $\mathcal{I}_{\mathcal{T}',\mathcal{A}} \not\models q[\vec{a}]$. It is sufficient to show that $\mathcal{I}_{\mathcal{T}',\mathcal{A}}$ is a model of $\mathcal{T}$. Let $C \sqsubseteq D \in \mathcal{T} \setminus \mathcal{T}'$ and assume that $\mathcal{I}_{\mathcal{T}',\mathcal{A}} \not\models C \sqsubseteq D$. Then $C^{\mathcal{I}_{\mathcal{T}',\mathcal{A}}} \neq \emptyset$. Let $q_C(v)$ be the tree-shaped conjunctive query corresponding to C, constructed in the standard way (see Appendix B for a formal definition of a similar construction). Then $\mathcal{I}_{\mathcal{T}',\mathcal{A}} \models \exists v\, q_C(v)$ and so $\mathcal{T}', \mathcal{A} \models \exists v\, q_C(v)$. Hence $\mathcal{T}, \mathcal{A} \models \exists v\, q_C(v)$ and all $X \in \mathsf{sig}(C)$ are not $\mathcal{CQ}$-empty for Σ given $\mathcal{T}$. Since $C \sqsubseteq D \in \mathcal{T}$, we also obtain $\mathcal{T}, \mathcal{A} \models \exists v\, q_D(v)$, where $q_D(v)$ is the tree-shaped conjunctive query corresponding to D. Thus, no $X \in \mathsf{sig}(D)$ is $\mathcal{CQ}$-empty for Σ given $\mathcal{T}$. But this means that $C \sqsubseteq D \in \mathcal{T}'$, which is a contradiction.

❑

Note that by Theorem 22, the $\mathcal{CQ}_\Sigma$-core can be computed in polynomial time if $\mathcal{T}$ is an $\mathcal{EL}$-TBox. We make some simple observations regarding $\mathcal{CQ}_\Sigma$-cores:

1. Theorem 42 fails in DLs that admit negation. For example, for $\mathcal{T} = \{A \sqsubseteq \neg B, \neg B \sqsubseteq E\}$ and $\Sigma = \{A\}$, any Σ-substitute of $\mathcal{T}$ w.r.t. $\mathcal{CQ}$ coincides with $\mathcal{T}$, but the $\mathcal{CQ}_\Sigma$-core of $\mathcal{T}$ is empty.

2. the $\mathcal{CQ}_\Sigma$-core is not always a minimal Σ-substitute w.r.t. $\mathcal{CQ}$. Consider, for example, $\mathcal{T} = \{A \sqsubseteq B_1, A \sqsubseteq B_2, B_1 \sqsubseteq B_2\}$ and let $\Sigma = \{A\}$. Then $\mathcal{T}' = \{A \sqsubseteq B_1, A \sqsubseteq B_2\}$ is a Σ-substitute w.r.t. $\mathcal{CQ}$ of $\mathcal{T}$ but the $\mathcal{CQ}_\Sigma$-core of $\mathcal{T}$ coincides with $\mathcal{T}$.

3. let the $\mathcal{IQ}_\Sigma$-core of TBox $\mathcal{T}$ be defined in analogy to the $\mathcal{CQ}_\Sigma$-core of $\mathcal{T}$, but based on $\mathcal{IQ}$-emptiness instead of $\mathcal{CQ}$-emptiness. Then the $\mathcal{IQ}_\Sigma$-core cannot serve as a Σ-substitute of $\mathcal{T}$ w.r.t. $\mathcal{IQ}$ even when $\mathcal{T}$ is an $\mathcal{EL}$-TBox. For example, let $\mathcal{T} = \{A \sqsubseteq \exists r.B, \exists r.B \sqsubseteq E\}$ and $\Sigma = \{A\}$. Then B is $\mathcal{IQ}$-empty for Σ given $\mathcal{T}$ and so the $\mathcal{IQ}_\Sigma$-core of $\mathcal{T}$ is empty. However, the empty TBox is not a Σ-substitute of $\mathcal{T}$ w.r.t. $\mathcal{IQ}$ since $\mathcal{T}, \mathcal{A} \models E(a)$ for $\mathcal{A} = \{A(a)\}$.

Interestingly, in contrast to the Σ-modules discussed above, $\bot$-modules as introduced in (Grau et al., 2008) turn out to be examples of Σ-substitutes. To define $\bot$-modules, let Σ be a signature. Two interpretations $\mathcal{I}$ and $\mathcal{I}'$ *coincide* w.r.t. Σ if $\Delta^{\mathcal{I}} = \Delta^{\mathcal{I}'}$ and $X^{\mathcal{I}} = X^{\mathcal{I}'}$ for all $X \in \Sigma$. A subset $\mathcal{T}'$ of a TBox $\mathcal{T}$ is called a *semantic $\bot$-module of $\mathcal{T}$ w.r.t.* Σ if for every interpretation $\mathcal{I}$ the interpretation $\mathcal{I}'$ that coincides with $\mathcal{I}$ w.r.t. $\Sigma \cup \mathsf{sig}(\mathcal{T}')$ and in which $X^{\mathcal{I}'} = \emptyset$ for all $X \notin \Sigma \cup \mathsf{sig}(\mathcal{T}')$ is a model of $\mathcal{T} \setminus \mathcal{T}'$. In (Grau et al., 2008), it is shown that extracting a minimal semantic $\bot$-module is of the same complexity as standard reasoning (that is, subsumption). In addition, it is shown that a syntactic approximation called the *syntactic $\bot$-module* can be computed in polynomial time (every syntactic $\bot$-module is a semantic $\bot$-module, but not necessarily the other way around). The following lemma establishes the relationship between $\bot$-modules and Σ-substitutes. A concept inclusion $C \sqsubseteq D$ is *tautological* if $\emptyset \models C \sqsubseteq D$.

Proposition 43 *Let $\mathcal{T}$ be a TBox formulated in any of the DLs introduced in this paper, and let $\mathcal{T}'$ be a semantic $\bot$-module of $\mathcal{T}$ w.r.t. Σ. Then*

1. *$\mathcal{T}'$ is a Σ-substitute of $\mathcal{T}$ w.r.t. $\mathcal{CQ}$;*
2. *$\Sigma \cup \mathsf{sig}(\mathcal{T}')$ contains all predicates that are not $\mathcal{CQ}$-empty for Σ given $\mathcal{T}$;*
3. *if $\mathcal{T}$ is an $\mathcal{ELI}$-TBox and does not contain tautological CIs, then the $\mathcal{CQ}_\Sigma$-core of $\mathcal{T}$ is contained in $\mathcal{T}'$.*

Proof. For Point 1, suppose $\mathcal{T}', \mathcal{A} \not\models q[\vec{a}]$, where $\mathcal{T}'$ be a semantic $\bot$-module of $\mathcal{T}$ w.r.t. Σ and $\mathcal{A}$ is a Σ-ABox. Let $\mathcal{I}$ be a model of $\mathcal{T}'$ and $\mathcal{A}$ such that $\mathcal{I} \not\models q[\vec{a}]$, and consider the interpretation $\mathcal{I}'$ that coincides with $\mathcal{I}'$ on $\Sigma \cup \mathsf{sig}(\mathcal{T}')$ and in which $X^{\mathcal{I}} = \emptyset$ for all remaining predicates X. Then $\mathcal{I}'$ is a model of $\mathcal{T}$ since $\mathcal{T}'$ is a semantic $\bot$-module of $\mathcal{T}$ w.r.t. Σ. and $\mathcal{I}'$ is a model of $\mathcal{A}$ since $\mathcal{A}$ is a Σ-ABox. Since we only shrinked the extension of predicates when transitioning from $\mathcal{I}$ to $\mathcal{I}'$ and $\mathcal{I} \not\models q[\vec{a}]$, we have $\mathcal{I}' \not\models q[\vec{a}]$. Hence $\mathcal{T}, \mathcal{A} \not\models q[\vec{a}]$, as required.

For Point 2, assume X is not $\mathcal{CQ}$-empty for Σ given $\mathcal{T}$, but $X \notin \Sigma \cup \mathsf{sig}(\mathcal{T}')$. Suppose $X = A$ for a concept name A (the case $X = r$ for a role name r is similar and left to the reader). Take a Σ-ABox $\mathcal{A}$ that is satisfiable w.r.t. $\mathcal{T}$ and such that $\mathcal{T}, \mathcal{A} \models \exists v\, A(v)$. Let $\mathcal{I}$ be a model of $(\mathcal{T}, \mathcal{A})$, and let $\mathcal{I}'$ be the interpretation that coincides with $\mathcal{I}$ on $\Sigma \cup \mathsf{sig}(\mathcal{T}')$ and in which $Y^{\mathcal{I}'} = \emptyset$ for all remaining Y (in particular we have $A^{\mathcal{I}'} = \emptyset$). By definition of semantic $\bot$-modules, $\mathcal{I}'$ is a model of $(\mathcal{T}, \mathcal{A})$. We have derived a contradiction because $\mathcal{I}'$ shows that $\mathcal{T}, \mathcal{A} \not\models \exists v\, A(v)$.

For Point 3, assume $\mathcal{T}$ is formulated in $\mathcal{ELI}$ and contains no tautological inclusions. Let $C \sqsubseteq D \in \mathcal{T} \setminus \mathcal{T}'$. Then, by the definition of semantic $\bot$-modules, $\mathsf{sig}(C \sqsubseteq D)$ contains a predicate $X \notin \Sigma \cup \mathsf{sig}(\mathcal{T}')$ (because otherwise $C \sqsubseteq D$ is a tautology). Thus, by Point 2, $\mathsf{sig}(C \sqsubseteq D)$ contains a predicate that is $\mathcal{CQ}$-empty for Σ given $\mathcal{T}$. But then $C \sqsubseteq D$ is not in the $\mathcal{CQ}_\Sigma$-core of $\mathcal{T}$, as required. ❑

By Point 1, we can use the algorithms for computing syntactic or semantic $\bot$-modules such as the ones in (Grau et al., 2008) to find Σ-substitutes in a large variety of DLs. By Point 2, such modules also provide an over-approximation of the set of predicates that are not CQ-empty. Finally, Point 3 means that, in $\mathcal{ELI}$, $\bot$-modules cannot be smaller than the $\mathcal{CQ}_\Sigma$-core unless there are tautological concept inclusions. In general, however, $\bot$-modules can be larger than the $\mathcal{CQ}_\Sigma$-core of a TBox. The following example shows that this can be the case already for acyclic $\mathcal{EL}$-TBoxes: let

$$\mathcal{T} = \{A \sqsubseteq \exists s_1.\exists r_1.\top \sqcap \exists s_2.\exists r_2.\top, B \equiv \exists r_1.\top \sqcap \exists r_2.\top\}$$

	Query Containment		Query Emptiness	
DL	$\mathcal{IQ}$	$\mathcal{CQ}$	$\mathcal{IQ}$	$\mathcal{CQ}$
$\mathcal{EL}$	ExpTime-c.	ExpTime-c.	PTime-c.	PTime-c
$\mathcal{EL}_\bot$	ExpTime-c.	ExpTime-c.	ExpTime-c.	ExpTime-c
$\mathcal{ELI}$, Horn-$\mathcal{ALCIF}$	ExpTime-c.	2ExpTime-c.	ExpTime-c.	ExpTime-c.
DL-Lite$_{core}$	in PTime	coNP-c.	NLogSpace-c.	coNP-c.
DL-Lite$_{horn}$	coNP-c.	Π_2^p-c.	coNP-c.	coNP-c.
$\mathcal{ALC}$	NExpTime-c.	NExpTime-h., in 2NExpTime	NExpTime-c.	NExpTime-c.
$\mathcal{ALCI}$	NExpTime-c.	2NExpTime-c.	NExpTime-c.	2ExpTime-c.

Figure 5: Query Containment vs Query Emptiness

and $\Sigma = \{A\}$. The predicates that are not $\mathcal{CQ}$-empty for Σ given $\mathcal{T}$ are A, s_1, s_2, r_1, r_2. Hence the $\mathcal{CQ}_\Sigma$-core of $\mathcal{T}$ contains only the first CI of $\mathcal{T}$. However, $\mathcal{T}$ has no non-trivial semantic $\bot$-modules w.r.t. Σ (and thus no syntactic ones either).

We now demonstrate the potential usefulness of Σ-substitutes and the $\mathcal{CQ}_\Sigma$-core by extending our case study from the medical domain. We again use the ontology SNOMED CT and the ABox signatures described at the beginning of this section, analyzing the size of the $\mathcal{CQ}_\Sigma$-core and comparing it to the size of the original ontology and of the syntactic $\bot$-module. For the real-world signature, $\mathcal{CQ}_\Sigma$-core contains 17,322 of the $\sim$370,000 concept inclusions in SNOMED CT. Thus, it is of about 5% the size of the original ontology. The $\bot$-module w.r.t. Σ turns out to be significantly larger than the $\mathcal{CQ}_\Sigma$-core, containing 27,383 axioms. For the random signatures, the sizes of $\mathcal{CQ}_\Sigma$-cores and $\bot$-modules are shown in the two right-most columns of Figure 4. They again confirm the findings for the real-world signature: the $\mathcal{CQ}_\Sigma$-core is much smaller both than the original ontology and than the $\bot$-module.

8. Related Work

Query emptiness is a fundamental problem in the static analysis of database queries. It is also called the *query satisfiability problem*. For XML, for example, it takes the following form: given an XPath query p and a DTD D, does there exist an XML document T such that T conforms to D and the answer of p on T is non-empty. The complexity of this problem ranges from tractable to undecidable depending on the XPath fragment, see e.g. (Benedikt et al., 2008) and references therein. In a DL context, query emptiness has been first considered in (Lubyte & Tessaris, 2008) where it is used as a step to guide the enrichment of ontologies.

The query emptiness problem studied in this paper is a special case of the following query containment problem, first considered (Bienvenu, Lutz, & Wolter, 2012). We can regard a pair $(\mathcal{T}, q)$ which consists of a TBox $\mathcal{T}$ and a query q as a compound query Q, called an *ontology-mediated query (OMQ)*, such that the answers to Q are the certain answers to q w.r.t. $\mathcal{T}$ (Bienvenu et al., 2014). Now take two OMQs $Q_i = (\mathcal{T}_i, q_i)$, $i \in \{1, 2\}$, such that q_1 and q_2 are IQs or CQs

of the same arity. Then Q_1 is Σ-*contained* in Q_2, for an ABox signature Σ, if for all Σ-ABoxes $\mathcal{A}$ that are satisfiable w.r.t. $\mathcal{T}_1$ and $\mathcal{T}_2$, we have $\mathsf{cert}_{\mathcal{T}_1,\mathcal{A}}(q_1) \subseteq \mathsf{cert}_{\mathcal{T}_2,\mathcal{A}}(q_2)$. In this case, we write $Q_1 \subseteq_\Sigma Q_2$. This notion of containment generalizes the more traditional query containment problem in DLs (Calvanese et al., 2007) by relativizing it to an ABox signature and admitting distinct TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$. Query emptiness of an IQ q for Σ given $\mathcal{T}$ can clearly be polynomially reduced to Σ-containment by setting $\mathcal{T}_1 = \mathcal{T}$, $q_1 = q$, $\mathcal{T}_2 = \emptyset$, and $q_2 = A(x)$ for a fresh concept name A, and similarly for CQs. Deciding Σ-containment, however, is often computationally harder than deciding query emptiness. Table 5 summarizes known results; the results about $\mathcal{EL}$ and DL-Lite are from (Bienvenu et al., 2012), the results about $\mathcal{ELI}$ from (Bienvenu, Hansen, Lutz, & Wolter, 2016), and the results about $\mathcal{ALC}$ and $\mathcal{ALCI}$ from (Bienvenu et al., 2014; Bourhis & Lutz, 2016).

Query emptiness is also closely related to explaining negative answers to queries. This problem was studied, for example, in (Calvanese, Ortiz, Simkus, & Stefanoni, 2013). Adopting an abductive reasoning approach, it can be described as follows. Assume $\mathcal{T}, \mathcal{A} \not\models q(\vec{a})$ for a TBox $\mathcal{T}$, ABox $\mathcal{A}$, and query q. To explain that $\vec{a}$ is not an answer to q, one wants to find minimal ABoxes $\mathcal{E}$ over a certain signature Σ of interest such that $\mathcal{A} \cup \mathcal{E}$ is satisfiable w.r.t. $\mathcal{T}$ and $\mathcal{T}, \mathcal{A} \cup \mathcal{E} \models q(\vec{a})$. Such Σ-ABoxes $\mathcal{E}$ are then regarded as an explanation for the missing answer and can be used for debugging purposes. It is shown in (Calvanese et al., 2013) that query emptiness of IQs and Boolean CQs reduces (under many-one logarithmic space reductions) to the problem of deciding the existence of an explanation for $\mathcal{T}, \mathcal{A} \not\models q(\vec{a})$ with $\mathcal{A} = \emptyset$. For DL-Lite$_\mathcal{A}$, the reduction even works for unions of conjunctive queries of any arity. In (Calvanese et al., 2013), this observation is used to obtain lower complexity bounds for explaining negative query answers, exploiting the results published in the conference predecessor of this paper (Baader, Bienvenu, Lutz, & Wolter, 2010). They also conjecture that, conversely, techniques for proving upper complexity bounds for query emptiness (such as the ones in this paper) can be used to obtain upper bounds for explaining negative answers.

9. Conclusion

We have investigated the computational complexity of query and predicate emptiness in the $\mathcal{EL}$, DL-Lite, and $\mathcal{ALC}$ families of DLs, concentrating on instance queries and conjunctive queries and showing that complexities range from NLOGSPACE to undecidable. We have also highlighted that, for different DLs and query languages, different kinds of witness ABoxes are sufficient to establish non-emptiness. DLs and queries that are not considered in this paper but which would be interesting to investigate in future work include the following:

- DLs that include transitive roles, role inclusions, symmetric roles, or role inclusion axioms (Horrocks, Kutz, & Sattler, 2006; Kazakov, 2010).

  In some cases, straightforward reductions to results presented in this paper are possible. For example, $\mathcal{IQ}$-query emptiness in Horn-$\mathcal{SHIF}$ is decidable in EXPTIME since for every Horn-$\mathcal{SHIF}$ TBox $\mathcal{T}$, IQ $A(x)$, and ABox signature Σ, one can construct in polynomial time a Horn-$\mathcal{ALCIF}$ TBox $\mathcal{T}'$ such that $\mathcal{T}, \mathcal{A} \models A(a)$ iff $\mathcal{T}', \mathcal{A}, \models A(a)$ for all Σ-ABoxes $\mathcal{A}$ (Hustadt et al., 2007; Kazakov, 2009). In other cases, such as $\mathcal{CQ}$-query emptiness in Horn-$\mathcal{SHIF}$, there seems to be no such reduction.

- DLs that include nominals.

- Other important classes of queries such as unions of conjunctive queries (UCQs).

  For materializable DLs such as Horn-$\mathcal{ALCIF}$, UCQ query emptiness can be reduced to CQ query emptiness since for every UCQ $q = \bigvee_{i \in I} q_i(\vec{x})$, we have $\mathcal{T}, \mathcal{A} \models q(\vec{a})$ iff there is an $i \in I$ such that $\mathcal{T}, \mathcal{A} \models q_i(\vec{a})$. Such a simple reduction does not work for non-Horn DLs such as $\mathcal{ALC}$.

It would also be interesting to develop practical algorithms for emptiness and to evaluate these algorithms on real-world ontologies and queries. Note that our algorithms for $\mathcal{EL}$ and DL-Lite are easily implementable and efficient as presented in this paper. This was actually confirmed by the case study in Section 7. More work will be required to design efficient algorithms for more expressive DLs. Finally, it would be relevant to investigate the notion of a Σ-substitute introduced in our application to modularity in more detail. For example, it is an open question how to compute minimal Σ-substitutes in expressive DLs such as $\mathcal{ALC}$ in practice, and what are the involved complexities.

Acknowledgements

The first author was partially supported by ANR project PAGODA (ANR-12-JS02-007-01). We are grateful to Julian Mendez and Dirk Walther for supporting us in the case study. We would like to thank the anonymous reviewers who provided excellent comments that helped us to improve the paper.

References

Artale, A., Calvanese, D., Kontchakov, R., & Zakharyaschev, M. (2009). The DL-Lite family and relations. *Journal of Artifical Intelligence Research (JAIR)*, *36*, 1–69.

Baader, F., Bienvenu, M., Lutz, C., & Wolter, F. (2010). Query and predicate emptiness in description logics. In *Proceedings of the 12th International Conference on Principles of Knowledge Representation and Reasoning (KR)*.

Baader, F., Brandt, S., & Lutz, C. (2005). Pushing the $\mathcal{EL}$ envelope. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 364–369.

Baader, F., Brandt, S., & Lutz, C. (2008). Pushing the $\mathcal{EL}$ envelope further. In *Proceedings of the Workshop on OWL: Experiences and Directions (OWLED)*.

Benedikt, M., Fan, W., & Geerts, F. (2008). XPath satisfiability in the presence of DTDs. *Journal of the ACM*, *55*(2), 1–79.

Bienvenu, M., Hansen, P., Lutz, C., & Wolter, F. (2016). First-order rewritability of conjunctive queries in Horn description logics. Forthcoming.

Bienvenu, M., Lutz, C., & Wolter, F. (2012). Query containment in description logics reconsidered. In *Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR)*.

Bienvenu, M., ten Cate, B., Lutz, C., & Wolter, F. (2014). Ontology-based data access: A study through disjunctive datalog, CSP, and MMSNP. *ACM Transactions on Database System (TODS)*, *39*(4), 33.

Botoeva, E., Kontchakov, R., Ryzhikov, V., Wolter, F., & Zakharyaschev, M. (2014). Query inseparability for description logic knowledge bases. In *Proceedings of the 14th International Conference in the Principles of Knowledge Representation and Reasoning (KR)*.

Botoeva, E., Kontchakov, R., Ryzhikov, V., Wolter, F., & Zakharyaschev, M. (2016). Games for query inseparability of description logic knowledge bases. *Artificial Intelligence Journal (AIJ)*, *234*, 78–119.

Bourhis, P., & Lutz, C. (2016). Containment in monadic disjunctive datalog, mmsnp, and expressive description logics. In *Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR)*.

Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rodriguez-Muro, M., & Rosati, R. (2009). Ontologies and databases: The DL-Lite approach. In *Tutorial Lectures of the 5th International Reasoning Web Summer School*, Vol. 5689 of *Lecture Notes in Computer Science*, pp. 255–356. Springer.

Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., & Rosati, R. (2007). Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *Journal of Automated Reasoning (JAR)*, *39*(3), 385–429.

Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., & Rosati, R. (2013). Data complexity of query answering in description logics. *Artificial Intelligence Journal (AIJ)*, *195*, 335–360.

Calvanese, D., De Giacomo, G., & Lenzerini, M. (1998). On the decidability of query containment under constraints. In *Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, pp. 149–158.

Calvanese, D., Ortiz, M., Simkus, M., & Stefanoni, G. (2013). Reasoning about explanations for negative query answers in DL-Lite. *Journal of Artificial Intelligence Research (JAIR)*, *48*, 635–669.

Chortaras, A., Trivela, D., & Stamou, G. B. (2011). Optimized query rewriting for OWL 2 QL. In *Proceedings of the 23rd International Conference on Automated Deduction (CADE)*, pp. 192–206.

Eiter, T., Gottlob, G., Ortiz, M., & Simkus, M. (2008). Query answering in the description logic Horn-$\mathcal{SHIQ}$. In *Proceedings of the 11th European Conference on Logics in Artificial Intelligence (JELIA)*, pp. 166–179.

Eiter, T., Ortiz, M., Simkus, M., Tran, T., & Xiao, G. (2012). Query rewriting for Horn-$\mathcal{SHIQ}$ plus rules. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*.

Gabbay, D., Kurucz, A., Wolter, F., & Zakharyaschev, M. (2003). *Many-Dimensional Modal Logics: Theory and Applications*. Elsevier.

Gatens, W., Konev, B., & Wolter, F. (2014). Lower and upper approximations for depleting modules of description logic ontologies. In *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI)*, pp. 345–350.

Glimm, B., Lutz, C., Horrocks, I., & Sattler, U. (2008). Answering conjunctive queries in the $\mathcal{SHIQ}$ description logic. *Journal of Artificial Intelligence Research (JAIR)*, *31*, 150–197.

Golbeck, J., Fragoso, G., Hartel, F., Hendler, J., Oberthaler, J., & Parsia, B. (2003). The national cancer institute's thesaurus and ontology. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, *1*(1), 75–80.

Grau, B. C., Horrocks, I., Kazakov, Y., & Sattler, U. (2008). Modular reuse of ontologies: Theory and practice. *Journal of Artifical Intelligence Research (JAIR)*, *31*, 273–318.

Haase, C. (2007). Complexity of subsumption in extensions of $\mathcal{EL}$. Master's thesis, Dresden University of Technology.

Horrocks, I., Kutz, O., & Sattler, U. (2006). The even more irresistible $\mathcal{SROIQ}$. In *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pp. 57–67.

Hustadt, U., Motik, B., & Sattler, U. (2004). A decomposition rule for decision procedures by resolution-based calculi. In *Proceedings of the 11th International Conference on Logic for Programming Artificial Intelligence and Reasoning (LPAR)*, pp. 21–35.

Hustadt, U., Motik, B., & Sattler, U. (2007). Reasoning in description logics by a reduction to disjunctive datalog. *Journal of Automated Reasoning (JAR)*, *39*(3), 351–384.

Kaminski, M., Schneider, T., & Smolka, G. (2011). Correctness and worst-case optimality of Pratt-style decision procedures for modal and hybrid logics. In *Proceedings of the 20th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX)*, pp. 196–210.

Kazakov, Y. (2009). Consequence-driven reasoning for Horn-$\mathcal{SHIQ}$ ontologies. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2040–2045.

Kazakov, Y. (2010). An extension of complex role inclusion axioms in the description logic $\mathcal{SROIQ}$. In *Proceedings of the 5th International Joint Conference on Automated Reasoning (IJCAR)*, pp. 472–486.

Konev, B., Ludwig, M., Walther, D., & Wolter, F. (2012). The logical difference for the lightweight description logic $\mathcal{EL}$. *Journal of Artificial Intelligence Research (JAIR)*, *44*, 633–708.

Konev, B., Lutz, C., Walther, D., & Wolter, F. (2013). Model-theoretic inseparability and modularity of description logic ontologies. *Artificial Intelligence Journal (AIJ)*, *203*, 66–103.

Kontchakov, R., Rodriguez-Muro, M., & Zakharyaschev, M. (2013). Ontology-based data access with databases: A short course. In *Proceedings of the International Reasoning Web Summer School*, pp. 194–229.

Kontchakov, R., Wolter, F., & Zakharyaschev, M. (2010). Logic-based ontology comparison and module extraction, with an application to DL-Lite. *Artificial Intelligence*, *174*(15), 1093–1141.

Krötzsch, M. (2012). OWL 2 profiles: An introduction to lightweight ontology languages. In *Tutorial Lectures of the 8th International Reasoning Web Summer School*, Vol. 7487 of *Lecture Notes in Computer Science*, pp. 112–183. Springer.

Krötzsch, M., Rudolph, S., & Hitzler, P. (2007). Complexity boundaries for Horn description logics. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI)*, pp. 452–457.

Levy, A. (1993). *Irrelevance Reasoning in Knowledge Based Systems*. Ph.D. thesis, Stanford University.

Lubyte, L., & Tessaris, S. (2008). Supporting the design of ontologies for data access. In *Proceedings of the 21st International Description Logic Workshop (DL)*.

Lutz, C. (2008). The complexity of CQ answering in expressive description logics. In *Proceedings of the 4th International Joint Conference on Automated Reasoning (IJCAR)*, pp. 179–193.

Lutz, C., Toman, D., & Wolter, F. (2009). Conjunctive query answering in the description logic $\mathcal{EL}$ using a relational database system. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2070–2075.

Lutz, C., & Wolter, F. (2010). Deciding inseparability and conservative extensions in the description logic $\mathcal{EL}$. *Journal of Symbolic Computation*, *45*(2), 194–228.

Lutz, C., & Wolter, F. (2012). Non-uniform data complexity of query answering in description logics. In *Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR)*.

Motik, B., Grau, B. C., Horrocks, I., Wu, Z., Fokoue, A., & Lutz, C. (2009). *OWL 2 Web Ontology Language: Profiles. W3C Recommendation*. Available at http://www.w3.org/TR/owl2-profiles/.

Ortiz, M., Calvanese, D., & Eiter, T. (2008). Data complexity of query answering in expressive description logics via tableaux. *Journal of Automated Reasoning (JAR)*, *41*(1), 61–98.

Ortiz, M., & Simkus, M. (2012). Reasoning and query answering in description logics. In *Proceedings of the 8th International Reasoning Web Summer School*, Vol. 7487 of *Lecture Notes in Computer Science*, pp. 1–53. Springer.

Ortiz, M., Simkus, M., & Eiter, T. (2008). Worst-case optimal conjunctive query answering for an expressive description logic without inverses. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI)*, pp. 504–510.

Patel, C., Cimino, J. J., Dolby, J., Fokoue, A., Kalyanpur, A., Kershenbaum, A., Ma, L., Schonberg, E., & Srinivas, K. (2007). Matching patient records to clinical trials using ontologies. In *Proceedings of the 6th International Semantic Web Conference (ISWC)*, pp. 816–829.

Pérez-Urbina, H., Motik, B., & Horrocks, I. (2009). A comparison of query rewriting techniques for dl-lite. In *Proceedings of the 22nd International Description Logic Workshop (DL)*.

Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., & Rosati, R. (2008). Linking data to ontologies. *Journal of Data Semantics*, *10*, 133–173.

Pratt, V. R. (1979). Models of program logics. In *Proceedings of IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 115–122.

Romero, A. A., Kaminski, M., Grau, B. C., & Horrocks, I. (2015). Ontology module extraction via datalog reasoning. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 1410–1416.

Tobies, S. (2001). *Complexity Results and Practical Algorithms for Logics in Knowledge Representation*. Ph.D. thesis, RWTH Aachen.

Vardi, M. Y. (1989). Automata theory for database theoreticans. In *Proceedings of the 7th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, pp. 83–92.

Vardi, M. Y. (1998). Reasoning about the past with two-way automata. In *Proceedings of the 25th International Colloquium on Automata, Languages and Programming (ICALP)*, pp. 628–641.

Appendix A. Proofs for Section 3

We formulate the result to be proved again.

Lemma 3 Let $\mathcal{T}$ be an $\mathcal{ALCIF}$-TBox. Then each CQ q is empty for Σ given $\mathcal{T}$ with the UNA iff it is empty for Σ given $\mathcal{T}$ without the UNA.

Proof. Consider a CQ q with answer variables $v_1, \dots, v_n$.

("Only if") Assume that q is non-empty for Σ given $\mathcal{T}$ without the UNA. Then there is a Σ-ABox $\mathcal{A}$ such that $\mathcal{A}$ is satisfiable w.r.t. $\mathcal{T}$ without the UNA and $\mathsf{cert}_{\mathcal{T},\mathcal{A}}(q) \neq \emptyset$ without the UNA. Take a model $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$ and suppose without loss of generality that $\Delta^{\mathcal{I}}$ is infinite. Define an equivalence relation $\equiv$ over $\mathsf{Ind}(\mathcal{A})$ by setting $a \equiv b$ whenever $a^{\mathcal{I}} = b^{\mathcal{I}}$. Choose a single representative from each equivalence class, and denote by a^* the representative of the equivalence class containing a. Let $\mathcal{A}'$ be the ABox obtained from $\mathcal{A}$ by replacing each individual a by a^*. We show that $\mathcal{A}'$ is satisfiable w.r.t. $\mathcal{T}$ with the UNA and that $\mathsf{cert}_{\mathcal{T},\mathcal{A}'}(q) \neq \emptyset$ with the UNA.

Regarding satisfiability, it is easy to see that $\mathcal{I}$ is a model of ($\mathcal{T}$ and) $\mathcal{A}'$ and that it satisfies the UNA for the individuals appearing in $\mathcal{A}'$. Moreover, since $\Delta^{\mathcal{I}}$ is infinite, we can reinterpret the individual names in $\mathsf{N_I} \setminus \mathsf{Ind}(\mathcal{A}')$ to obtain an interpretation that is a model of $\mathcal{A}'$ and $\mathcal{T}$ and satisfies the UNA.

Now for showing $\mathsf{cert}_{\mathcal{T},\mathcal{A}'}(q) \neq \emptyset$ with the UNA. Take some $(a_1, \dots, a_n) \in \mathsf{cert}_{\mathcal{T},\mathcal{A}}(q)$ without the UNA. We aim to show that $(a_1^*, \dots, a_n^*) \in \mathsf{cert}_{\mathcal{T},\mathcal{A}'}(q)$. Let $\mathcal{J}'$ be any model of $\mathcal{T}$ and $\mathcal{A}'$ that satisfies the UNA. We have to show that there is a match for q in $\mathcal{J}'$ with $\pi(v_i) = (a_i^*)^{\mathcal{J}'}$ for every $1 \leq i \leq n$. Consider the interpretation $\mathcal{J}$ obtained from $\mathcal{J}'$ by setting $a^{\mathcal{J}} = (a^*)^{\mathcal{J}'}$ for every $a \in \mathsf{Ind}(\mathcal{A})$. It is easy to see that $\mathcal{J}$ is a model of $\mathcal{A}$ and $\mathcal{T}$ without the UNA, so $\mathcal{J} \models q[a_1, \dots, a_n]$ and there is a match π for q in $\mathcal{J}$ such that $\pi(v_i) = a_i^{\mathcal{J}}$ for every $1 \leq i \leq n$. Then π is also the desired match for q in $\mathcal{J}'$, which finishes the proof.

("If") Assume that q is non-empty for Σ given $\mathcal{T}$ with the UNA. Then there is a Σ-ABox $\mathcal{A}$ such that $\mathcal{A}$ is satisfiable w.r.t. $\mathcal{T}$ with the UNA and $\mathsf{cert}_{\mathcal{T},\mathcal{A}}(q) \neq \emptyset$ with the UNA. Clearly, $\mathcal{A}$ is also satisfiable w.r.t. $\mathcal{T}$ without the UNA and it remains to show that $\mathsf{cert}_{\mathcal{T},\mathcal{A}}(q) \neq \emptyset$ without the UNA. Let $(a_1, \dots, a_n) \in \mathsf{cert}_{\mathcal{T},\mathcal{A}}(q)$ with the UNA, and let $\mathcal{I}$ be any model of $\mathcal{A}$ and $\mathcal{T}$ without the UNA. We have to show that $\mathcal{I} \models q[a_1, \dots, a_n]$. For any $a \in \mathsf{Ind}(\mathcal{A})$, let $\mathcal{I}_a$ be the following *unfolding of $\mathcal{I}$ at $a^{\mathcal{I}}$*:

- the domain $\Delta^{\mathcal{I}_a}$ of $\mathcal{I}_a$ consists of all words $d_0 r_0 d_1 \cdots r_{k-1} d_k$ with $d_0, \dots, d_k \in \Delta^{\mathcal{I}}$ and $r_0, \dots, r_{k-1}$ (potentially inverse) roles such that $d_0 = a^{\mathcal{I}}$, $(d_i, d_{i+1}) \in r_i^{\mathcal{I}}$ for all $i < k$, $r_i^- \neq r_{i+1}$ for functional r_{i+1} and $i < k$, and $r_0(a, b) \notin \mathcal{A}$ for any $b \in \mathsf{Ind}(\mathcal{A})$ if r_0 is functional.
- $A^{\mathcal{I}_a} = \{d_0 r_0 d_1 \cdots d_k \mid d_k \in A^{\mathcal{I}}\}$ for all $A \in \mathsf{N_C}$;
- $r^{\mathcal{I}_a} = \{(d_0 r_0 d_1 \cdots d_k, d_0 r_0 d_1 \cdots d_k r_{k+1} d_{k+1}) \mid r = r_{k+1}\} \cup$
  $\{(d_0 r_0 d_1 \cdots d_k r_{k+1} d_{k+1}, d_0 r_0 d_1 \cdots d_k) \mid r^- = r_{k+1}\}$ for all $r \in \mathsf{N_R}$.

Assume that the $\mathcal{I}_a$ are mutually disjoint and let the individual name a be the root of $\mathcal{I}_a$. Then we obtain the interpretation $\mathcal{J}$ by taking the disjoint union of all $\mathcal{I}_a$, $a \in \mathsf{Ind}(\mathcal{A})$, adding (a, b) to $r^{\mathcal{J}}$ whenever $r(a, b) \in \mathcal{A}$, and setting $a^{\mathcal{J}} = a$ for all $a \in \mathsf{Ind}(\mathcal{A})$. One can show that $\mathcal{J}$ is a model of $\mathcal{A}$ and $\mathcal{T}$ with the UNA. Thus $\mathcal{J} \models q[a_1, \dots, a_n]$ and there is a match π for q in $\mathcal{J}$ with $\pi(v_i) = (a_i^*)^{\mathcal{J}}$ for every $1 \leq i \leq n$. It can be verified that π' defined by setting $\pi'(v_i) = a^{\mathcal{I}}$ if $\pi(v_i) = a \in \mathsf{Ind}(\mathcal{A})$

and $\pi'(v_i) = d_k$ if $\pi(v) = d_0, \ldots, d_k$ with $k \geq 1$ is a match of q in $\mathcal{I}$, thus $\mathcal{I} \models q[a_1, \ldots, a_n]$, as required. ❑

Appendix B. Proofs for Section 4

We restate the first result to be proved.

Theorem 16 In $\mathcal{ALC}$, $\mathcal{CQ}$-query emptiness is in NEXPTIME.

The general idea for proving Theorem 16 is as follows. Given an $\mathcal{ALC}$-TBox $\mathcal{T}$, a signature Σ, and a CQ q, by Theorem 13 it suffices to test whether $\mathcal{T}, \mathcal{A}_{\mathcal{T},\Sigma} \not\models q$. We thus start by computing $\mathcal{A}_{\mathcal{T},\Sigma}$. To check whether $\mathcal{T}, \mathcal{A}_{\mathcal{T},\Sigma} \not\models q$, we then guess an extension of $\mathcal{T}$ with a set $\mathcal{T}'$ of concept inclusions and an extension of $\mathcal{A}_{\mathcal{T},\Sigma}$ with a set $\mathcal{A}'$ of ABox assertions such that $\mathcal{T}'$ and $\mathcal{A}'$ that are satisfied only in models of $\mathcal{T}$ and $\mathcal{A}_{\mathcal{T},\Sigma}$ in which q has no match; subsequently, it remains to test the satisfiability of $\mathcal{A}_{\mathcal{T},\Sigma} \cup \mathcal{A}'$ w.r.t. $\mathcal{T} \cup \mathcal{T}'$. The subtlety lies in selecting the class of extensions to be guessed from in a careful enough way so that the final satisfiability check can be carried out in NEXPTIME.

We reuse some technical definitions and results from (Lutz, 2008) where it is proved that the combined complexity of CQ-answering in the DL $\mathcal{SHQ}$ is in EXPTIME. The definitions are slightly modified since in (Lutz, 2008), (i) only CQs without answer variables are considered and (ii) the DL $\mathcal{SHQ}$ is used, of which $\mathcal{ALC}$ is a proper fragment. However, it is straightforward to verify that the proofs given in (Lutz, 2008) also work with our modified definitions.

A CQ q can be viewed as a directed graph $G_q^d = (V_q^d, E_q^d)$ with $V_q^d = \mathsf{var}(q)$ and $E_q^d = \{(v, v') \mid r(v, v') \in q \text{ for some } r \in \mathsf{N_R}\}$. We call q *directed tree-shaped* if G_q^d is a directed tree and $r(v, v'), s(v, v') \in q$ implies $r = s$. If q is directed tree-shaped and v_0 is the root of G_q^d, we call v_0 the *root of* q. For $U \subseteq \mathsf{var}(q)$, we write $q|_U$ to denote the restriction of q to atoms that contain only variables from U. The set $\mathsf{DTrees}(q)$ of *directed tree-shaped subqueries* of q is defined as follows:

$$\mathsf{DTrees}(q) = \{q|_U \mid U = \mathsf{Reach}_q(v), v \in \mathsf{var}(q), q|_U \text{ is directed tree-shaped}\}$$

where $\mathsf{Reach}_q(v)$ is the set of variables that are reachable from v in G_q^d. We say that

- q' is *obtained from q by performing fork elimination* if q' is obtained from q by selecting two atoms $r(v', v)$ and $s(v'', v)$ such that $v', v'', v \in \mathsf{qvar}(q)$ and $v' \neq v''$, and identifying v' and v'';
- q' is a *fork rewriting* of q if q' is obtained from q by repeatedly (but not necessarily exhaustively) performing fork elimination;
- q' is a *maximal fork rewriting* of q if q' is a fork rewriting and no further fork elimination is possible in q'.

The following is shown in (Lutz, 2008), and it plays a central role in the subsequent definitions.

Lemma 44 *Up to variable renaming, every CQ has a unique maximal fork rewriting.*

The following definitions of splittings and spoilers are also taken from (Lutz, 2008). To understand a splitting of a CQ q on an intuitive level, it is useful to consider matches of q in a model $\mathcal{I}$ of

a TBox $\mathcal{T}$ and ABox $\mathcal{A}$ that has a special shape: $\mathcal{I}$ consists of a core part whose elements are exactly (the interpretations of) the ABox individuals in $\mathcal{A}$ and of tree-shaped parts that are attached to the element in the core part and disjoint from each other. In fact, it is proved in (Lutz, 2008) that if $\mathcal{T}, \mathcal{A} \not\models q$, then there is a model $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{A}$ of the described shape such that $\mathcal{I} \not\models q$. A match of q in a model of the described shape partitions the variables in q into several sets: a set R which contains the variables that are matched to an ABox individual; sets $S_1, \ldots, S_n$ which represent disjoint tree-shaped subqueries of q that are matched into a tree part of $\mathcal{I}$ and whose root is connected to some variable in R via a role atom in q; and a set T which represents a collection of tree-shaped subqueries of q that are disconnected from the variables in R and S_i. In addition to this partitioning, splittings record the variable in R to which each set $S_1, \ldots, S_n$ is connected and the ABox elements that the variables in R are mapped to. We now define splittings formally.

Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{ALC}$-knowledge base and q a CQ. A *splitting of q w.r.t. $\mathcal{K}$* is a tuple $\Pi = \langle R, T, S_1, \ldots, S_n, \mu, \nu \rangle$, where $R, T, S_1, \ldots, S_n$ is a partitioning of $\mathsf{var}(q)$, $\mu : \{1, \ldots, n\} \to R$ assigns to each set S_i a variable $\mu(i)$ in R, and $\nu : R \to \mathsf{Ind}(\mathcal{A})$ assigns to each variable in R an individual in $\mathcal{A}$. A splitting has to satisfy the following conditions:

1. the CQ $q|_T$ is a variable-disjoint union of directed tree-shaped queries;
2. the queries $q|_{S_i}$, $1 \leq i \leq n$, are directed tree-shaped;
3. if $r(v, v') \in q$, then one of the following holds: (i) v, v' belong to the same set $R, T, S_1, \ldots, S_n$ or (ii) $v \in R$, $\mu(i) = v$, and $v' \in S_i$ is the root of $q|_{S_i}$;
4. for $1 \leq i \leq n$, there is a unique $r \in \mathsf{N_R}$ such that $r(\mu(i), v_0) \in q$, with v_0 the root of $q|_{S_i}$;
5. $\mathsf{avar}(q) \subseteq R$.

Let q be a directed tree-shaped CQ. We define an $\mathcal{ALC}$-concept $C_{q,v}$ for each $v \in \mathsf{var}(q)$:

- if v is a leaf in G_q^d, then $C_{q,v} = \bigsqcap_{A(v) \in q} C$;
- otherwise, $C_{q,v} = \bigsqcap_{A(v) \in q} A \sqcap \bigsqcap_{r(v,v') \in q} \exists r.C_{q,v'}$.

If v is the root of q, we use C_q to abbreviate $C_{q,v}$.

In the following, we allow compound concepts and negated roles to be used in ABox assertions. The semantics of such assertions and corresponding KBs is defined in the expected way: an interpretation $\mathcal{I}$ satisfies $C(a)$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$ and $\mathcal{I}$ satisfies $\neg r(a, b)$ if $\mathcal{I}$ does not satisfy $r(a, b)$. Let $\Pi = \langle R, T, S_1, \ldots, S_n, \mu, \nu \rangle$ be a splitting of q w.r.t. $\mathcal{K}$ such that $q_1, \ldots, q_k$ are the (directed tree-shaped) disconnected components of $q|_T$. An $\mathcal{ALC}$-knowledge base $(\mathcal{T}', \mathcal{A}')$ is a *spoiler for q, $\mathcal{K}$, and Π* if one of the following conditions hold:

1. $\top \sqsubseteq \neg C_{q_i} \in \mathcal{T}'$, for some i with $1 \leq i \leq k$;
2. there is an atom $A(v) \in q$ with $v \in R$ and $\neg A(\nu(v)) \in \mathcal{A}'$;
3. there is an atom $r(v, v') \in q$ with $v, v' \in R$ and $\neg r(\nu(v), \nu(v')) \in \mathcal{A}'$;
4. $\neg D(\nu(\mu(i))) \in \mathcal{A}'$ for some i with $1 \leq i \leq n$, and where $D = \exists r.C_{q|_{S_i}}$ with v_0 root of $q|_{S_i}$ and $r(\mu(i), v_0) \in q$.

We call $\mathcal{K}'$ a *spoiler for q and $\mathcal{K}$* if (i) for every fork rewriting q' of q, and every splitting Π of q' w.r.t. $\mathcal{K}$, $\mathcal{K}'$ is a spoiler for q', $\mathcal{K}$, and Π; and (ii) $\mathcal{K}'$ is minimal with Property (i). The following result is proved in (Lutz, 2008).

Theorem 45 *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{ALC}$-knowledge base and q a CQ. Then $\mathcal{K} \not\models q$ iff there is a spoiler $(\mathcal{T}', \mathcal{A}')$ for q and $\mathcal{K}$ such that $\mathcal{A} \cup \mathcal{A}'$ is satisfiable w.r.t. $\mathcal{T} \cup \mathcal{T}'$.*

The following lemma, which is observed in (Lutz, 2008), plays a central role for obtaining a NEXPTIME decision procedure.

Lemma 46 *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{ALC}$-knowledge base, q a CQ, q^* its maximal fork rewriting, and $\mathcal{K}' = (\mathcal{T}', \mathcal{A}')$ a spoiler for q and $\mathcal{K}$. Then $\mathcal{K}'$ contains only concept inclusions and ABox assertions of the following form:*

1. $\top \sqsubseteq \neg C_{q'}$ *with* $q' \in \mathsf{DTrees}(q^*)$;
2. $\neg A(a)$ *with* $a \in \mathsf{Ind}(\mathcal{A})$ *and* A *occurring in* q;
3. $\neg r(a, b)$ *with* $a, b \in \mathsf{Ind}(\mathcal{A})$ *and* r *occurring in* q;
4. $\neg D(a)$ *with* $a \in \mathsf{Ind}(\mathcal{A})$ *and* $D = \exists r.C_{q'}$, *where* r *occurs in* q *and* $q' \in \mathsf{DTrees}(q^*)$.

Note that while the definition of a spoiler for q and $\mathcal{K}$ refers to all fork rewritings of q, of which there are exponentially many, Lemma 46 only refers to the unique maximal form rewriting q^*. In fact, since the cardinality of $\mathsf{DTrees}(q^*)$ is clearly bounded by the size of q, the number of concept inclusions and assertions listed in Lemma 46 is only polynomial in the size of $\mathcal{A}$ and q.

We are now set up for the proof of Theorem 16. By Theorems 13 and 45, a CQ q is empty for a signature Σ given a TBox $\mathcal{T}$ iff there is a spoiler $(\mathcal{T}', \mathcal{A}')$ for q and $(\mathcal{T}, \mathcal{A}_{\mathcal{T},\Sigma})$ such that $\mathcal{A}_{\mathcal{T},\Sigma} \cup \mathcal{A}'$ is satisfiable w.r.t. $\mathcal{T} \cup \mathcal{T}'$. Given a CQ q, signature Σ and TBox $\mathcal{T}$, we can thus decide emptiness of q for Σ given $\mathcal{T}$ as follows:

1. compute $\mathcal{A}_{\mathcal{T},\Sigma}$;
2. guess a TBox $\mathcal{T}'$ and an ABox $\mathcal{A}'$ that satisfy Conditions 1 to 4 from Lemma 46 for the KB $\mathcal{K} = (\mathcal{T}, \mathcal{A}_{\mathcal{T},\Sigma})$ and such that there is no role assertion $r(a, b) \in \mathcal{A}_{\mathcal{T},\Sigma}$ with $\neg r(a, b) \in \mathcal{A}'$;
3. verify that $(\mathcal{T}', \mathcal{A}')$ is a spoiler for q and $(\mathcal{T}, \mathcal{A}_{\mathcal{T},\Sigma})$;
4. verify that $\mathcal{A}_{\mathcal{T},\Sigma} \cup \mathcal{A}'$ is satisfiable w.r.t. $\mathcal{T} \cup \mathcal{T}'$.

It remains to argue that this yields a NEXPTIME algorithm. As already noted, Step 1 can be carried out in (deterministic) exponential time. Due to Conditions 1 to 4 from Lemma 46 and since $\mathcal{A}_{\mathcal{T},\Sigma}$ is of size at most exponential in $\mathcal{T}$ and Σ, the TBox $\mathcal{T}'$ and ABox $\mathcal{A}'$ guessed in Step 2 are of size at most exponential in $\mathcal{T}$ and Σ, and of size polynomial in q. Step 3 can be implemented by a straightforward iteration over all fork rewritings q' of q and splittings Π of q' w.r.t. $(\mathcal{T}, \mathcal{A}_{\mathcal{T},\Sigma})$, which requires only exponential time.

It thus remains to deal with Step 4. Let Γ be the closure under single negations of the union of the following sets of concepts:

- $\Sigma \cap \mathsf{N_C}$;

- all concepts that occur in $\mathcal{T}$ (possibly as subconcepts);
- all concept names that occur in q;
- all concepts C_q and $\exists r.C_q$ and their subconcepts, where $q \in \mathsf{DTrees}(q^*)$, q^* the maximal fork rewriting of q, and r occurs in q.

Based on the remark after Lemma 46, it is easy to verify (and crucial for our argument) that $|\Gamma|$ is polynomial in Σ, $\mathcal{T}$, and q. A Γ-*type* is a set $t \subseteq \Gamma$ such that for some model $\mathcal{I}$ of $\mathcal{T} \cup \mathcal{T}'$ and some $d \in \Delta^{\mathcal{I}}$, we have $t = \{C \in \Gamma \mid d \in C^{\mathcal{I}}\}$. As in Section 4, we introduce a notion of coherence between types: we say that the pair of Γ-types (t, t') is r-*coherent*, denoted $t \rightsquigarrow_r t'$, if for all $\exists r.C \in \Gamma$, $C \in t'$ implies $\exists r.C \in t$. The set $\mathfrak{T}_\Gamma$ of all Γ-types can be computed in ExpTime.

To verify satisfiability of $\mathcal{A}_{\mathcal{T},\Sigma} \cup \mathcal{A}'$ w.r.t. $\mathcal{T} \cup \mathcal{T}'$, we guess a map $\pi : \mathsf{Ind}(\mathcal{A}_{\mathcal{T},\Sigma}) \to \mathfrak{T}_\Gamma$, accept if the following two conditions are satisfied and reject otherwise:

(i) $C(c) \in \mathcal{A}_{\mathcal{T},\Sigma} \cup \mathcal{A}'$ implies $C \in \pi(c)$ and

(ii) $r(b, c) \in \mathcal{A}_{\mathcal{T},\Sigma} \cup \mathcal{A}'$, $C \in \pi(c)$, and $\exists r.C \in \Gamma$ imply $\exists r.C \in \pi(b)$.

Clearly, checking whether these two conditions are satisfied can be done in single exponential time. It thus remains to argue that $\mathcal{A}_{\mathcal{T},\Sigma} \cup \mathcal{A}'$ is satisfiable w.r.t. $\mathcal{T} \cup \mathcal{T}'$ just in the case that there exists a map π verifying these conditions. First note that given a model $\mathcal{I}$ of the KB $(\mathcal{T} \cup \mathcal{T}', \mathcal{A}_{\mathcal{T},\Sigma} \cup \mathcal{A}')$, we can define the desired map by setting $\pi(c) = \{C \in \Gamma \mid c^{\mathcal{I}} \in C^{\mathcal{I}}\}$. Conversely, given a map π satisfying conditions (i) and (ii), we define an interpretation $\mathcal{I}_\pi$ as follows:

$$\Delta^{\mathcal{I}_\pi} = \mathfrak{T}_\Gamma \qquad A^{\mathcal{I}_\pi} = \{t \in \mathfrak{T}_\Gamma \mid A \in t\} \qquad r^{\mathcal{I}_\pi} = \{(t, t') \in \mathfrak{T}_\Gamma \times \mathfrak{T}_\Gamma \mid t \rightsquigarrow_r t'\} \qquad c^{\mathcal{I}_\pi} = \pi(c)$$

It is readily verified that for all $C \in \Gamma$ and $t \in \mathfrak{T}_\Gamma$, we have $C \in t$ iff $t \in C^{\mathcal{I}_\pi}$. From this, we can show, using a similar argument to that given in Lemma 10, that $\mathcal{I}_\pi$ is a model of $(\mathcal{T} \cup \mathcal{T}', \mathcal{A}_{\mathcal{T},\Sigma} \cup \mathcal{A}')$.

We now complete the proof of our undecidability result (Theorem 19) by proving Lemma 20.

Lemma 20 $(\mathfrak{T}, H, V)$ admits a tiling iff there is a Σ-ABox $\mathcal{A}$ that is satisfiable w.r.t. $\mathcal{T}$ and such that $\mathcal{T}, \mathcal{A} \models \exists v\, A(v)$.

Proof. ("Only if") Straightforward. Consider a tiling $f : \{0, \ldots, n\} \times \{0, \ldots, m\} \to \mathfrak{T}$ for $(\mathfrak{T}, H, V)$. Create individuals $a_{i,j}$ for $0 \leq i \leq n$ and $0 \leq j \leq m$, and consider the ABox $\mathcal{A}$ composed of the following assertions:

- $x(a_{i,j}, a_{i+1,j})$ for $0 \leq i < n$ and $0 \leq j \leq m$
- $x^-(a_{i+1,j}, a_{i,j})$ for $0 \leq i < n$ and $0 \leq j \leq m$
- $y(a_{i,j}, a_{i,j+1})$ for $0 \leq j < m$ and $0 \leq i \leq n$
- $y^-(a_{i,j+1}, a_{i,j})$ for $0 \leq j < m$ and $0 \leq i \leq n$
- $T_h(a_{i,j})$ if $f(i, j) = T_h$.

It can easily be verified that $\mathcal{A}$ is satisfiable w.r.t. $\mathcal{T}$ and satisfies $\mathcal{T}, \mathcal{A} \models \exists v\, A(v)$.

("If") Let $\mathcal{A}$ be a Σ-ABox satisfiable w.r.t. $\mathcal{T}$ and such that $\mathcal{T}, \mathcal{A} \models \exists v\, A(v)$. We first show that I_x and I_y enforce that x^- is the inverse of x and y^- is the inverse of y, respectively, and that C forces relevant grid cells to be closed. For $r \in \{x, y\}$ we call $a \in \mathsf{Ind}(\mathcal{A})$ an *r-defect* if there exists $b \in \mathsf{Ind}(\mathcal{A})$ such that $r(a, b) \in \mathcal{A}$ and $r^-(b, a) \notin \mathcal{A}$. We call a an *inv-defect* if it is an x-defect or a y-defect. We call $a \in \mathsf{Ind}(\mathcal{A})$ a *cl-defect* if there exist $x(a, b), y(a, c), y(b, d), x(c, e) \in \mathcal{A}$ with $d \neq e$ such that a is not an inv-defect, b is not a y-defect and c is not an x-defect.

Claim 1. There exists a model $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{A}$ such that for all $a \in \mathsf{Ind}(\mathcal{A})$:

(d1) $a^{\mathcal{I}} \notin I_r^{\mathcal{I}}$, for all r-defects $a \in \mathsf{Ind}(\mathcal{A})$ and $r \in \{x, y\}$;

(d2) $a^{\mathcal{I}} \notin C^{\mathcal{I}}$, for all cl-defects $a \in \mathsf{Ind}(\mathcal{A})$.

Moreover, $\mathcal{I}$ satisfies the following conditions for all $a \in \mathsf{Ind}(\mathcal{A})$, role names r, and $h \in \{1, \dots, p\}$:

1. $\Delta^{\mathcal{I}} = \mathsf{Ind}(\mathcal{A})$;
2. $a^{\mathcal{I}} = a$;
3. $(a, a') \in r^{\mathcal{I}}$ implies $r(a, a') \in \mathcal{A}$;
4. $a \in T_h^{\mathcal{I}}$ implies $T_h(a) \in \mathcal{A}$.

Proof of Claim 1. Let $r \in \{x, y\}$. Call a two-element set $\{a, b\}$ a *r-defect witness* if there exists $c \in \mathsf{Ind}(\mathcal{A})$ such that $r(a, c), r^-(c, b) \in \mathcal{A}$. Consider the undirected graph G with nodes $\mathsf{Ind}(\mathcal{A})$ and the set of r-defect witnesses as its edges. Note that G has degree at most two (since r and r^- are functional). Hence G is three-colorable. Choose a three coloring of G with colors $B_{r,1} = Z_{r,1} \sqcap Z_{r,2}$, $B_{r,2} = Z_{r,1} \sqcap \neg Z_{r,2}$ and $B_{r,3} = \neg Z_{r,1} \sqcap Z_{r,2}$ and choose the interpretation of the concept names $Z_{r,1}, Z_{r,2}$ in $\mathcal{I}$ correspondingly. We set $I_r^{\mathcal{I}} = \bigcup_{i=1,2,3} (B_{r,i} \sqcap \exists r. \exists r^-. B_{r,i})^{\mathcal{I}}$.

Call a two-element set $\{d, e\}$ a *cl-defect witness* if there exist $x(a, b), y(a, c), y(b, d), x(c, e) \in \mathcal{A}$ such that a is not an inv-defect, b is not a y-defect and c is not an x-defect. Consider the undirected graph G with nodes $\mathsf{Ind}(\mathcal{A})$ and the set of cl-defect witnessess as its edges. Note that G has degree at most two (again since x, x^-, y, and y^- are all functional). Hence G is three-colorable with colors $C_1 = Z_{c,1} \sqcap Z_{c,2}$, $C_2 = Z_{c,1} \sqcap \neg Z_{c,2}$ and $C_3 = \neg Z_{c,1} \sqcap Z_{c,2}$ and we can choose the interpretation of the concept names $Z_{c,1}, Z_{c,2}$ in $\mathcal{I}$ correspondingly. We set $C^{\mathcal{I}} = \bigcup_{i=1,2,3} (\exists x. \exists y. C_i \sqcap \exists y. \exists x. C_i)^{\mathcal{I}}$.

Since neither existential restrictions nor the concept names $\mathcal{T}_h$ occur in the right-hand side of CIs in $\mathcal{T}$, it is not hard to verify that we can interpret the remaining concept names in $\mathcal{T}$ in such a way that the additional conditions on $\mathcal{I}$ are satisfied. (*End of proof of claim*)

Let $\mathcal{I}$ be a model satisfying the conditions of Claim 1. We additionally assume w.l.o.g. that $\mathcal{I}$ is *A, Y-minimal*: there is no model $\mathcal{J}$ of $\mathcal{T}$ and $\mathcal{A}$ satisfying the conditions of Claim 1 such that $A^{\mathcal{J}} \subseteq A^{\mathcal{I}}$ and $Y^{\mathcal{J}} \subseteq Y^{\mathcal{I}}$ and at least one of these inclusions is proper.

Let $a_A \in A^{\mathcal{I}}$. We now exhibit a grid structure in $\mathcal{A}$ that gives rise to a tiling for $(\mathfrak{T}, H, V)$. We start by identifying a diagonal that starts at a_A and ends at an instance of T_{final}.

Claim 2. There is a set $\mathcal{G} = \{r_1(a_{i_0,j_0}, a_{i_1,j_1}), \dots, r_{k-1}(a_{i_{k-1},j_{k-1}} a_{i_k,j_k}), T_{\mathsf{final}}(a_{i_k,j_k})\} \subseteq \mathcal{A}$ such that

- $i_0 = 0$, $j_0 = 0$, and $a_{0,0} = a_A$;

- for $1 \leq \ell < k$, we either have (i) $r_\ell = x$, $i_{\ell+1} = i_\ell + 1$, and $j_{\ell+1} = j_\ell$ or (ii) $r_\ell = y$, $j_{\ell+1} = j_\ell + 1$, and $i_{\ell+1} = i_\ell$.

Proof of claim. If there is no such sequence, we can convert $\mathcal{I}$ into a new model $\mathcal{J}$ of $\mathcal{T}$ and $\mathcal{A}$ by interpreting Y as false at all points reachable in $\mathcal{I}$ (equivalently: $\mathcal{A}$) from a_A and setting $A^{\mathcal{J}} = A^{\mathcal{I}} \setminus \{a_A\}$, which contradicts the A, Y-minimality of $\mathcal{I}$. (*End of proof of claim*)

Let n be the number of occurrences of the role x in the ABox $\mathcal{G}$ from Claim 1 and m the number of occurrences of y. We next show

Claim 3. We have that

(a) $a_{0,0} \in T^{\mathcal{I}}_{\mathsf{init}}$.

(b) $a_{i,j} \in R^{\mathcal{I}}$ implies $i = n$;

(c) $a_{i,j} \in U^{\mathcal{I}}$ implies $j = m$;

(d) $a_{i,j} \in Y^{\mathcal{I}}$ for all $a_{i,j} \in \mathsf{Ind}(\mathcal{G})$;

(e) for all $a_{i,j} \in \mathsf{Ind}(\mathcal{G})$, there is a (unique) T_h with $a_{i,j} \in T^{\mathcal{I}}_h$, henceforth denoted $T_{i,j}$;

(f) $(T_{i,j}, T_{i+1,j}) \in H$ for all $a_{i,j}, a_{i+1,j} \in \mathsf{Ind}(\mathcal{G})$ and $(T_{i,j}, T_{i,j+1}) \in V$ for all $a_{i,j}, a_{i,j+1} \in \mathsf{Ind}(\mathcal{G})$.

Proof of claim. Point (a) is an easy consequence of the fact that $a_{0,0} = a_A$, $a_A \in A^{\mathcal{I}}$, and $\mathcal{I}$ is A, Y-minimal. For (b), first note that there is a unique $\ell \leq k$ such that $i_s = n$ for all $s \in \{\ell, \dots, k\}$ and $i_s < n$ for all $s \in \{0, \dots, \ell - 1\}$. Due to the CI $R \sqsubseteq \forall x.\bot$, $a_{i_{\ell-1}, j_{\ell-1}} \notin R^{\mathcal{I}}$. To show that $a_{i_s, j_s} \notin R^{\mathcal{I}}$ for all $s < \ell - 1$, it suffices to use the CIs $R \sqsubseteq \forall x.\bot$ and $R \sqsubseteq \forall y.R$. The proof of (c) is similar. We prove (d)-(f) together, showing by induction on ℓ that (d)-(f) are satisfied for all initial parts

$$\mathcal{G}_\ell := \{r_1(a_{i_0,j_0}, a_{i_1,j_1}), \dots, r_{\ell-1}(a_{i_{\ell-1},j_{\ell-1}} a_{i_\ell,j_\ell})\}$$

of $\mathcal{G}$, with $\ell \leq k$. For the base case, $a_{i_0,j_0} = a_A \in A^{\mathcal{I}}$ clearly implies $a_{i_0,j_0} \in Y^{\mathcal{I}}$, thus (d) is satisfied. Point (e) follows from (a) and the disjointness of tiles expressed in $\mathcal{T}$. Point (f) is vacuously true since there is only a single individual in $\mathcal{G}_0$. For the induction step, assume that $\mathcal{G}_{\ell-1}$ satisfies (d)-(f). We distinguish four cases:

- $a_{i_{\ell-1},j_{\ell-1}} \in (\neg U \sqcap \neg R)^{\mathcal{I}}$.

  Since $\mathcal{G}_{\ell-1}$ satisfies (d), we have $a_{i_{\ell-1},j_{\ell-1}} \in Y^{\mathcal{I}}$, and the definition of $\mathcal{T}$ and the A, Y-minimality of $\mathcal{I}$ together with the fact $a_{i_{\ell-1},j_{\ell-1}} \in (\neg U \sqcap \neg R)^{\mathcal{I}}$ ensure that

  $$a_{i_{\ell-1},j_{\ell-1}} \in (\exists x.(T_g \sqcap Y \sqcap \exists y.Y) \sqcap \exists y.(T_h \sqcap Y \sqcap \exists x.Y) \sqcap I_x \sqcap I_y \sqcap C \sqcap T_f)^{\mathcal{I}}$$

  for some $(T_f, T_g) \in H$ and $(T_f, T_h) \in V$. Using the functionality of x and y, it is now easy to show that $\mathcal{G}_\ell$ satisfies (d)-(f).

- $a_{i_{\ell-1},j_{\ell-1}} \in (\neg U \sqcap R)^{\mathcal{I}}$.

  Since $a_{i_{\ell-1},j_{\ell-1}} \in R^{\mathcal{I}}$, $\mathcal{T}$ ensures that there is no x-successor of $a_{i_{\ell-1},j_{\ell-1}}$ in $\mathcal{I}$. Moreover, $a_{i_{\ell-1},j_{\ell-1}} \in Y^{\mathcal{I}}$. Together with the definition of $\mathcal{T}$, we get

  $$a_{i_{\ell-1},j_{\ell-1}} \in (\exists y.(T_g \sqcap Y \sqcap R) \sqcap I_y \sqcap T_f)^{\mathcal{I}}$$

for some $(T_f, T_g) \in V$. We must have $i_\ell = i_{\ell-1}$, $j_\ell = j_{\ell-1} + 1$, and $r_{\ell-1} = y$. Using the functionality of y, it is now easy to show that $\mathcal{G}_\ell$ satisfies (d)-(f).

- $a_{i_{\ell-1},j_{\ell-1}} \in (U \sqcap \neg R)^{\mathcal{I}}$.

  Analogous to the previous case.

- $a_{i_{\ell-1},j_{\ell-1}} \in (U \sqcap R)^{\mathcal{I}}$.

  Then there is neither an x-successor nor a y-successor of $a_{i_{\ell-1},j_{\ell-1}} \in (U \sqcap R)^{\mathcal{I}}$. It follows that $\ell - 1 = k$, in contradiction to $\ell \leq k$.

(*End of proof of claim*)

Next, we extend $\mathcal{G}$ to a full grid such that Conditions (a)-(e) from Claim 3 are still satisfied. Once this is achieved, it is trivial to read off a solution for the tiling problem. The construction of the grid consists of exhaustive application of the following two steps:

1. if $x(a_{i,j}, a_{i+1,j}), y(a_{i+1,j}, a_{i+1,j+1}) \in \mathcal{G}$ and there is no $a_{i,j+1} \in \mathsf{Ind}(\mathcal{G})$ with $y(a_{i,j}, a_{i,j+1}) \in \mathcal{G}$ and $x(a_{i,j+1}, a_{i+1,j+1}) \in \mathcal{G}$, then identify an $a_{i,j+1} \in \mathsf{Ind}(\mathcal{A})$ such that $y(a_{i,j}, a_{i,j+1}) \in \mathcal{A}$ and $x(a_{i,j+1}, a_{i+1,j+1}) \in \mathcal{A}$ and add the latter two assertions to $\mathcal{G}$.

2. if $y(a_{i,j}, a_{i,j+1}), x(a_{i,j+1}, a_{i+1,j+1}) \in \mathcal{G}$ and there is no $a_{i+1,j} \in \mathsf{Ind}(\mathcal{G})$ with $x(a_{i,j}, a_{i+1,j}) \in \mathcal{G}$ and $y(a_{i+1,j}, a_{i+1,j+1}) \in \mathcal{G}$, then identify an $a_{i+1,j} \in \mathsf{Ind}(\mathcal{A})$ such that $x(a_{i,j}, a_{i+1,j}) \in \mathcal{A}$ and $y(a_{i+1,j}, a_{i+1,j+1}) \in \mathcal{A}$ and add the latter two assertions to $\mathcal{G}$.

It is not hard to see that exhaustive application of these rules yields a full grid, i.e., for the final $\mathcal{G}$ we have (i) $\mathsf{Ind}(\mathcal{G}) = \{a_{i,j} \mid i \leq n, j \leq m\}$, (ii) $x(a_{i,j}, a_{i',j'}) \in \mathcal{G}$ iff $i' = i + 1$ and $j = j'$, and (iii) $y(a_{i,j}, a_{i',j'}) \in \mathcal{G}$ iff $i = i'$ and $j' = j + 1$.

Since the two steps of the construction are completely analogous, we only deal with Case 1 in detail. Thus let $x(a_{i,j}, a_{i+1,j}), y(a_{i+1,j}, a_{i+1,j+1}) \in \mathcal{G}$ with $a_{i,j+1} \notin \mathsf{Ind}(\mathcal{G})$. Clearly, $i < n$ and $j < m$. By (b) and (c), we thus have $a_{i,j} \notin (R \sqcup U)^{\mathcal{I}}$. Since $a_{i,j} \in Y^{\mathcal{I}}$ by (d) and $\mathcal{I}$ is A, Y-minimal, we get that

$$a_{i,j} \in (\exists x.(T_g \sqcap Y \sqcap \exists y.Y) \sqcap \exists y.(T_h \sqcap Y \sqcap \exists x.Y) \sqcap I_x \sqcap I_y \sqcap C \sqcap T_f)^{\mathcal{I}}$$

for some $(T_f, T_g) \in H$ and $(T_f, T_h) \in V$. This together with the minimality of $\mathcal{I}$ means we can select $a_{i,j+1}, b \in \mathsf{Ind}(\mathcal{A})$ such that $y(a_{i,j}, a_{i,j+1}), x(a_{i,j+1}, b) \in \mathcal{A}$, $a_{i,j+1}, b \in Y^{\mathcal{I}}$, and $T_{i,j+1} = T_h$. With this choice, (a), (d), (e), and the second half of (f) are clearly satisfied. To get the properties required by Step 1 above, we have to show that $b = a_{i+1,j+1}$. If we can show this, then the satisfaction of (b) and (c) before we apply the construction step, and the CIs

$$R \sqsubseteq \forall x.\bot \quad R \sqsubseteq \forall y.R \quad U \sqsubseteq \forall y.\bot \quad U \sqsubseteq \forall x.U$$

ensure that (b) and (c) are still satisfied after the construction step. Showing $b = a_{i+1,j+1}$ will also give us the first half of (f). Finally, to prove that $b = a_{i+1,j+1}$ it is sufficient to show that $a_{i,j}$ is not a cl-defect in $\mathsf{Ind}(\mathcal{A})$. But this follows from Claim 1 since $a_{i,j} \in C^{\mathcal{I}}$, $a_{i,j} \in I_x^{\mathcal{I}} \cap I_y^{\mathcal{I}}$, $a_{i+1,j} \in I_y^{\mathcal{I}}$, and $a_{i,j+1} \in I_x^{\mathcal{I}}$.

We can now use the completed grid to build a solution to our tiling problem: the tile at point (i, j) is the unique tile which is satisfied by $\mathcal{I}$ at $a_{i,j} \in \mathsf{Ind}(\mathcal{A})$. Property (f) of Claim 2 and the correctness of our grid construction ensure that adjacent tiles satisfy the vertical and horizontal constraints. ❑

Appendix C. Proofs for Section 5

Theorem 22. In $\mathcal{EL}$, $\mathcal{CQ}$-query emptiness can be decided in PTIME.

Proof. By Lemma 21, it suffices to show that for any n-ary CQ q and alphabet Σ, it can be decided in PTIME whether $\mathcal{T}, \mathcal{A}_\Sigma \models q[a_\Sigma, \ldots, a_\Sigma]$ where $\mathcal{A}_\Sigma$ is the total Σ-ABox. First note that we have $\mathcal{T}, \mathcal{A}_\Sigma \models q[a_\Sigma, \ldots, a_\Sigma]$ iff $\mathcal{T}, \widehat{\mathcal{A}}_\Sigma \models \widehat{q}$, where

- $\widehat{\mathcal{A}}_\Sigma$ is obtained from $\mathcal{A}_\Sigma$ by adding the assertion $X(a_\Sigma)$, where X is a concept name that does not occur in Σ, $\mathcal{T}$, and q;
- $\widehat{q}$ is the Boolean CQ obtained from q by adding the conjunct $X(v)$ for each answer variable v and then quantifying away all answer variables.

Recall from the discussion before Lemma 44 that every CQ q can be viewed as a directed graph G_q^d. We say that a Boolean CQ q is *directed forest-shaped* if it is a disjoint union of directed tree-shaped Boolean CQs. Every Boolean CQ q that is directed forest-shaped corresponds to a concept C_q in the description logic $\mathcal{EL}^u$ that extends $\mathcal{EL}$ with the universal role u such that $\mathcal{T}, \mathcal{A} \models q$ iff $\mathcal{T}, \mathcal{A} \models C(a)$ for all $a \in \mathsf{Ind}(\mathcal{A})$ (Lutz & Wolter, 2010). Checking the latter condition is possible in PTIME (Lutz & Wolter, 2010). Thus, it is sufficient to convert $\widehat{q}$ in polynomial time into a directed forest-shaped CQ $\widehat{q}'$ such that $\mathcal{T}, \widehat{\mathcal{A}}_\Sigma \models \widehat{q}$ iff $\mathcal{T}, \widehat{\mathcal{A}}_\Sigma \models \widehat{q}'$.

To construct $\widehat{q}'$ from $\widehat{q}$, we exhaustively apply the following rewriting rules:

1. if $r(v, v'')$ and $r(v', v'')$ are in the query, then identify v and v' by replacing all occurrences of v' with v;
2. if $r(v', v)$ and $s(v'', v)$ are in the query (with $r \neq s$), then identify v, v', and v'' by replacing all occurrences of v' and v'' with v;
3. if a cycle $r_0(v_0, v_1), \ldots, r_{n-1}(v_{n-1}, v_n)$, $v_n = v_0$ is in the query and $\{v_0, \ldots, v_{n-1}\}$ contains at least two variables, then identify all variables $v_0, \ldots, v_{n-1}$ by replacing all occurrences of $v_1, \ldots, v_{n-1}$ with v_0.

If the resulting query contains a reflexive loop $r(v, v)$ with $r \notin \Sigma$, then we immediately return "no". Otherwise, we replace in a final step each reflexive loop $r(v, v)$ with $r \in \Sigma$ with $X(v)$. The query resulting from this last step is $\widehat{q}'$. It is easy to see that the query obtained at this point is directed forest-shaped since every variable has at most one predecessor and there are no cycles in the corresponding directed graph.

To prove correctness of this algorithm, we first establish the following claim:

Claim. If $\widehat{q}'$ is defined, then $\mathcal{T}, \widehat{\mathcal{A}}_\Sigma \models \widehat{q}$ iff $\mathcal{T}, \widehat{\mathcal{A}}_\Sigma \models \widehat{q}'$.

It suffices to prove that each rule application preserves (non)entailment of the query by $\mathcal{T}$ and $\widehat{\mathcal{A}}_\Sigma$. As a preliminary, we recall that as shown in (Lutz & Wolter, 2010), there exists a materialization $\mathcal{J}_{\mathcal{T}, \widehat{\mathcal{A}}_\Sigma}$ of $(\mathcal{T}, \widehat{\mathcal{A}}_\Sigma)$ which is a directed tree-shaped interpretation with the individual a_Σ as its root and (potentially) additional reflexive loops added to this root (an interpretation is *directed tree-shaped* if the corresponding CQ in which the domain elements of the interpretation are regarded as variables is directed tree-shaped). Assume that rewriting rule 1 is applied to a query p resulting in a query p'. It is clear that $\mathcal{T}, \widehat{\mathcal{A}}_\Sigma \models \widehat{p}'$ implies $\mathcal{T}, \widehat{\mathcal{A}}_\Sigma \models \widehat{p}$. For the converse, assume $\mathcal{T}, \widehat{\mathcal{A}}_\Sigma \models \widehat{p}$ and let $\mathcal{J}_{\mathcal{T}, \widehat{\mathcal{A}}_\Sigma}$ be the materialization of $\mathcal{T}$ and $\widehat{\mathcal{A}}_\Sigma$ introduced above. Then there is a match of p

in $\mathcal{J}_{\mathcal{T},\widehat{\mathcal{A}}_\Sigma}$. Since $\mathcal{J}_{\mathcal{T},\widehat{\mathcal{A}}_\Sigma}$ does not contain domain elements d, d', d'' with $d \neq d'$ and such that for some role name r, $(d, d'') \in r^{\mathcal{J}_{\mathcal{T},\widehat{\mathcal{A}}_\Sigma}}$ and $(d', d'') \in r^{\mathcal{J}_{\mathcal{T},\widehat{\mathcal{A}}_\Sigma}}$, this match of p in $\mathcal{J}_{\mathcal{T},\widehat{\mathcal{A}}_\Sigma}$ must map the identified variables v and v' to the same domain element and is thus also a match of p'. The other two rules and the replacement of $r(v, v)$, $r \in \Sigma$, with $X(v)$ can be dealt with in a similar way.

By the claim, we can substitute $\widehat{q}$ with $\widehat{q}'$ as intended. Moreover, it is easy to see that we have $\mathcal{T}, \widehat{\mathcal{A}}_\Sigma \not\models \widehat{q}$ if the algorithm returns "no" due to a reflexive loop $r(v, v)$ with $r \notin \Sigma$: simply use the interpretation $\mathcal{J}_{\mathcal{T},\widehat{\mathcal{A}}_\Sigma}$ as in the proof of the claim. ❑

Proposition 28. For every Horn-$\mathcal{ALCIF}$ TBox $\mathcal{T}$, ABox signature Σ, and CQ q, one can construct in polynomial time an $\mathcal{ELIF}_\bot$-TBox $\mathcal{T}'$ in normal form such that q is empty for Σ given $\mathcal{T}$ iff q is empty for Σ given $\mathcal{T}'$.

Proof. The proof is similar to reductions provided in (Hustadt et al., 2007; Kazakov, 2009). Nevertheless, because (Kazakov, 2009) considers reductions preserving subsumption only and because both (Hustadt et al., 2007) and (Kazakov, 2009) do not reduce to $\mathcal{ELIF}_\bot$ TBoxes, we give a detailed proof.

The following rules can be used to rewrite $\mathcal{T}$ into an $\mathcal{ELIF}_\bot$-TBox in normal form (all freshly introduced concept names are not in $\mathsf{sig}(\mathcal{T}) \cup \Sigma \cup \mathsf{sig}(q)$. Assume $L \sqsubseteq R$ is given.

- If L is of the form $L_1 \sqcap L_2$ and R is not a concept name, then take a fresh concept name A and replace $L \sqsubseteq R$ by $L \sqsubseteq A$ and $A \sqsubseteq R$. If R is a concept name, and either L_1 or L_2 are not concept names, then take fresh concept names A_1, A_2 and replace $L \sqsubseteq R$ by $L_1 \sqsubseteq A_1$, $L_2 \sqsubseteq A_2$ and $A_1 \sqcap A_2 \sqsubseteq R$;

- If L is of the form $L_1 \sqcup L_2$ and R is a concept name, then replace $L \sqsubseteq R$ by $L_1 \sqsubseteq R$ and $L_2 \sqsubseteq R$. Otherwise take a fresh concept name A and replace $L \sqsubseteq R$ by $L \sqsubseteq A$ and $A \sqsubseteq R$;

- If L is of the form $\exists r.L'$ and L' is not a concept name, then take a fresh concept name A' and replace $L \sqsubseteq R$ by $L' \sqsubseteq A'$ and $\exists r.A' \sqsubseteq R$;

- If R is of the form $\neg A$, then replace $L \sqsubseteq R$ by $L \sqcap A \sqsubseteq \bot$;

- If R is of the form $R_1 \sqcap R_2$ and L is not a concept name, then take a fresh concept name A and replace $L \sqsubseteq R$ by $L \sqsubseteq A$ and $A \sqsubseteq R$. Otherwise take fresh concept names A_1, A_2 and replace $L \sqsubseteq R$ by $L \sqsubseteq A_1$, $L \sqsubseteq A_2$, $A_1 \sqsubseteq R_1$, and $A_2 \sqsubseteq R_2$;

- If R is of the form $\neg L' \sqcup R'$, then replace $L \sqsubseteq R$ by $L \sqcap L' \sqsubseteq R'$;

- If R is of the form $\exists r.R'$ and R' is not a concept name, then take a fresh concept name A' and replace $L \sqsubseteq R$ by $L \sqsubseteq \exists r.A'$ and $A' \sqsubseteq R'$;

- If R is of the form $\forall r.R'$, then replace $L \sqsubseteq R$ by $\exists r^-.L \sqsubseteq R$.

The resulting TBox $\mathcal{T}'$ is as required. In particular, for every Σ-ABox $\mathcal{A}$ and model $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}'$, we have that $\mathcal{I}$ is also a model of $\mathcal{T}$; conversely, every model $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$ can be extended to a model of $\mathcal{T}$ by appropriately interpreting the fresh concept names. Consequently, we have $\mathsf{cert}_{\mathcal{T}}(q, \mathcal{A}) = \mathsf{cert}_{\mathcal{T}'}(q, \mathcal{A})$ and thus q is empty for Σ given $\mathcal{T}$ iff q is empty for Σ given $\mathcal{T}'$. ❑

Proposition 30. Let $\mathcal{T}$ be an $\mathcal{ELI}\mathcal{F}_\bot$-TBox, Σ an ABox signature, and q a CQ. If q is non-empty for Σ given $\mathcal{T}$, then this is witnessed by a Σ-ABox that is forest-shaped, has width at most $|q|$, and degree at most $|\mathcal{T}|$.

Proof. Assume that q has answer variables $v_1, \dots, v_n$ and is non-empty for Σ given $\mathcal{T}$. Then we can find a Σ-ABox $\mathcal{A}$ that is satisfiable w.r.t. $\mathcal{T}$ and such that $\mathsf{cert}_{\mathcal{T},\mathcal{A}}(q) \neq \emptyset$. To identify a forest-shaped witness for the non-emptiness of q for Σ given $\mathcal{T}$, consider the canonical model $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ of $(\mathcal{T}, \mathcal{A})$. By construction, $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ consists of an 'ABox part' $\mathcal{I}_0$, which is the restriction of $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ to $\mathsf{Ind}(\mathcal{A})$, and tree-shaped interpretations $\mathcal{I}_a$, $a \in \mathsf{Ind}(\mathcal{A})$, rooted at a and containing no other ABox individuals. Since $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ is universal, there is a match π of q in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$. Let I_π consist of all individuals $a \in \mathsf{Ind}(\mathcal{A})$ such that there is some $v \in \mathsf{var}(q)$ with $\pi(v)$ in $\mathcal{I}_a$ (possibly $\pi(v) = a$). Let $\mathcal{A}_0$ be the ABox obtained by restricting $\mathcal{A}$ to the individuals in I_π; this is going to be the root component of the forest-shaped witness we are seeking to define (observe that $|\mathsf{Ind}(\mathcal{A}_0)| \leq |q|$). To add the tree components, we consider, for each $a \in I_\pi$, the (typically infinite) tree-shaped ABox $\mathcal{A}_a^u$ that is obtained by *unraveling* $\mathcal{A}$ starting from a, as in (Lutz & Wolter, 2012):

- $\mathsf{Ind}(\mathcal{A}_a^u)$ is the set of sequences $\alpha = c_0 r_0 c_1 \dots r_{m-1} c_m$ with $c_0, \dots, c_m \in \mathsf{Ind}(\mathcal{A})$ and $r_0, \dots, r_{m-1}$ (possibly inverse) roles such that (i) $c_0 = a$, (ii) $c_1 \notin I_\pi$, (iii) $r_j(c_{j-1}, c_j) \in \mathcal{A}$ for all $0 \leq j < m$, and (iv) $(c_{j-1}, r_{j-1}^-) \neq (c_{j+1}, r_j)$ for $j > 0$; we say that α is *a copy of* c_m;
- if $A(c) \in \mathcal{A}$ and $\alpha \in \mathsf{Ind}(\mathcal{A}_a^u)$ is a copy of c, then $A(\alpha) \in \mathcal{A}_a^u$;
- if $\alpha \in \mathsf{Ind}(\mathcal{A}_a^u)$ is a copy of c, and $\beta = rc' \in \mathsf{Ind}(\mathcal{A}_a^u)$, then $r(\alpha, \beta) \in \mathcal{A}_a^u$;
- if $\alpha \in \mathsf{Ind}(\mathcal{A}_a^u)$ is a copy of c, and $\beta = r^- c' \in \mathsf{Ind}(\mathcal{A}_a^u)$, then $r(\beta, \alpha) \in \mathcal{A}_a^u$.

We let $\widehat{\mathcal{A}}$ be the union of $\mathcal{A}_0$ and the tree-shaped ABoxes $\{\mathcal{A}_a^u \mid a \in I_\pi\}$. Observe that by Conditions (ii) and (iv) of the first item and since $\mathcal{A}$ satisfies all functionality statements in $\mathcal{T}$, so does $\widehat{\mathcal{A}}$. Note that $\widehat{\mathcal{A}}$ is forest-shaped, but need neither be finite nor of degree at most $|\mathcal{T}|$; we are going to fix this later.

We next aim to show that $\widehat{\mathcal{A}}$ is satisfiable w.r.t. $\mathcal{T}$ and that $\mathsf{cert}_{\mathcal{T},\widehat{\mathcal{A}}}(q) \neq \emptyset$. To this end, we construct a universal model $\mathcal{J}$ of $\widehat{\mathcal{A}}$ and $\mathcal{T}$. Start with $\widehat{\mathcal{A}}$ viewed as an interpretation $\mathcal{J}_0$, as in the construction of canonical models. Then take, for each $a \in \mathsf{Ind}(\mathcal{A})$ and each of a's copies $\alpha \in \mathsf{Ind}(\widehat{\mathcal{A}})$, a copy $\mathcal{I}_\alpha$ of the tree interpretation $\mathcal{I}_a$ such that (i) the root of $\mathcal{I}_\alpha$ is α, (ii) $\Delta^{\mathcal{J}_0} \cap \Delta^{\mathcal{I}_\alpha} = \{\alpha\}$, (iii) $\alpha \neq \beta$ implies disjointness of $\mathcal{I}_\alpha$ and $\mathcal{I}_\beta$, and (iv) if $\alpha = a$ then $\mathcal{I}_\alpha$ is identical to the original tree interpretation $\mathcal{I}_a$ (and not a copy). If $d' \in \Delta^{\mathcal{I}_\alpha}$ is the result of renaming $d \in \Delta^{\mathcal{I}_a}$, then d' is called a *copy* of d. The desired interpretation $\mathcal{J}$ is obtained by taking the union of $\mathcal{J}_0$ and all $\mathcal{I}_\alpha$. Note that every element of $\mathcal{J}$ is the copy of an element in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$, and that, by construction, $\mathcal{J}$ is a model of $\widehat{\mathcal{A}}$.

It is straightforward to show by induction on the structure of C that for every $\mathcal{ELI}$-concept C and every element $e \in \Delta^{\mathcal{J}}$ that is a copy of $d \in \Delta^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$, $e \in C^{\mathcal{J}}$ iff $d \in C^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$. Since $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ is a model of $\mathcal{T}$, it follows that $\mathcal{J}$ is a model of $\mathcal{T}$ and thus $\widehat{\mathcal{A}}$ is satisfiable w.r.t. $\mathcal{T}$. We only sketch the proof that $\mathcal{J}$ is universal. Let $\mathcal{I}$ be a model of $\widehat{\mathcal{A}}$ and $\mathcal{T}$. We start to define a homomorphism h_0 from $\mathcal{J}$ to $\mathcal{I}$ by setting $h_0(a) = a^{\mathcal{I}}$ for all $a \in \mathsf{Ind}(\widehat{\mathcal{A}})$. It remains to extend h_0 to the $\mathcal{I}_\alpha$ components of

$\mathcal{J}$. Each such $\mathcal{I}_\alpha$ is a copy of a tree interpretation $\mathcal{I}_a$ in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ such that α is a copy of a. It is shown in (Lutz & Wolter, 2012) that[6]

($*$) if $\alpha \in \mathsf{Ind}(\widehat{\mathcal{A}})$ is a copy of $a \in \mathsf{Ind}(\mathcal{A})$, then $a \in A^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$ implies $\mathcal{T}, \widehat{\mathcal{A}} \models A(\alpha)$ for all concept names A.

Recall that $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ was generated by the derivation rules for building canonical models. Using a straightforward induction on the number of rule applications and exploiting ($*$) and the fact that $\mathcal{T}$ is in normal form, one can construct a homomorphism h_a from $\mathcal{I}_a$ to $\mathcal{I}$ such that $h(a) = \alpha^{\mathcal{I}}$. By renaming, we obtain a homomorphism h_α from $\mathcal{I}_\alpha$ to $\mathcal{I}$ such that $h_\alpha(\alpha) = \alpha^{\mathcal{I}}$. The desired homomorphism h is the union of h_0 and all h_α. We have thus established that $\mathcal{J}$ is universal. Going through the construction of $\mathcal{J}$ (and in particular using Point (iv)), it can be verified that the match π of q in $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ is also a match of q in $\mathcal{J}$. Since $\mathcal{J}$ is universal, this yields $\mathsf{cert}_{\mathcal{T},\widehat{\mathcal{A}}}(q) \neq \emptyset$ as desired.

We now want to remove individuals from $\widehat{\mathcal{A}}$ such that the resulting ABox is of degree at most $|\mathcal{T}|$ and still witnesses non-emptiness of q for Σ given $\mathcal{T}$. Since $\mathcal{J}$ is universal, there is a homomorphism h from $\mathcal{J}$ to the canonical model $\mathcal{I}_{\mathcal{T},\widehat{\mathcal{A}}}$. Composing the match π with h, we obtain a match τ of q in $\mathcal{I}_{\mathcal{T},\widehat{\mathcal{A}}}$ that sends every variable to an individual in $\mathcal{A}_0$ or to an element of a tree below such an individual. We inductively mark individuals in $\widehat{\mathcal{A}}$ that are 'relevant' for the match τ, starting with all individuals in $\mathcal{A}_0$ and then proceeding as follows: whenever Rule 2 or 4 adds a marked individual x to $A^{\mathcal{I}_{\mathcal{T},\widehat{\mathcal{A}}}}$ during the construction of $\mathcal{I}_{\mathcal{T},\widehat{\mathcal{A}}}$ because of the presence of $(x, y) \in r^{\mathcal{I}_{\mathcal{T},\mathcal{A}}}$ (please see the formulation of the mentioned rules), then mark y. It can be verified that every individual outside of $\mathcal{A}_0$ has at most one marked neighbor for each existential restriction in $\mathcal{T}$. The (potentially infinite) forest-shaped ABox $\widehat{\mathcal{A}}_d$ obtained from $\widehat{\mathcal{A}}$ by dropping all assertions that involve at least one unmarked individual is thus of degree at most $|\mathcal{T}|$. Moreover, the marking construction ensures that the canonical model $\mathcal{I}_{\mathcal{T},\widehat{\mathcal{A}}_d}$ contains $\mathcal{A}_0$ and each interpretation $\mathcal{I}_a$, $a \in \mathsf{Ind}(\mathcal{A}_0)$, hence τ is a match for q in $\mathcal{I}_{\mathcal{T},\widehat{\mathcal{A}}_d}$.

At this point, the ABox $\widehat{\mathcal{A}}_d$ is almost the required forest witness, except that it may be infinite. It remains to invoke compactness to obtain a finite subset $\widehat{\mathcal{A}}_f \subseteq \widehat{\mathcal{A}}_d$ such that $\mathsf{cert}_{\mathcal{T},\widehat{\mathcal{A}}_f}(q) \neq \emptyset$. Clearly, $\widehat{\mathcal{A}}_f$ contains a forest witness for the non-emptiness of q for Σ given $\mathcal{T}$. ❑

The following lemmas establish the two statements in Lemma 34.

Lemma 47 *Every canonical proper $\Sigma_R \cup \Sigma_N$-labeled tree is well-founded.*

Proof. Let $\langle T, \ell\rangle$ be a canonical proper $\Sigma_R \cup \Sigma_N$-labeled tree, and let $\mathcal{I}_0, \mathcal{I}_1, \ldots$ be the interpretations encountered during the construction of the canonical model of $\mathcal{T}$ and $\mathcal{A}_{\langle T,\ell\rangle}$. Since $\langle T, \ell\rangle$ is canonical, $\mathcal{I}_{\langle T,\ell\rangle}$ is the canonical model of $\mathcal{A}_{\langle T,\ell\rangle}$ and $\mathcal{T}$.

We will slightly abuse terminology by using the term *concept atom* to refer to statements of the form $B(e)$ where B is a concept name (or $\top$) and e is a domain element. A *role atom* will take the form $r(e, e')$ with r a role and e, e' domain elements. We will say that a concept atom $B(e)$ (resp. role atom $r(e, e')$) is in an interpretation $\mathcal{J}$ if $e \in B^{\mathcal{J}}$ (resp. $(e, e') \in r^{\mathcal{J}}$). For each atom α in $\mathcal{I}_{\langle T,\ell\rangle}$, the *rank of* α is the smallest i such that α is in $\mathcal{I}_i$. We show by induction on the rank that every concept atom in $\mathcal{I}_{\langle T,\ell\rangle}$ has a derivation, thus $\langle T, \ell\rangle$ is well-founded.

6. In (Lutz & Wolter, 2012), this is actually shown for the case where the root component $\mathcal{A}_0$ of $\widehat{\mathcal{A}}$ from which we start to unravel consists of all the individual names in $\mathcal{A}$, but contains no concept and role assertions; the proof also goes through in our case.

The induction start is straightforward as concept atoms in $\mathcal{I}_0$ involve a concept from $\Sigma \cup \{\top\}$ and an element x such that either $x \in \mathsf{Ind}(\mathcal{A})$ with $\mathcal{A} = \ell(\varepsilon)$ or $x \in T \setminus \{\varepsilon\}$ with $M \in \ell(x)$, and every such atom has a derivation of depth 0. For the induction step, let $B(x)$ be a concept atom in $\mathcal{I}_{i+1} \setminus \mathcal{I}_i$. We consider the rule application that resulted in the addition of $B(x)$:

1. Assume that $B(x)$ is in $\mathcal{I}_{i+1}$ because of an application of Rule 1, that is, $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B \in \mathcal{T}$ and $x \in A_j^{\mathcal{I}_i}$ for $1 \leq j \leq n$.

   For every $1 \leq j \leq n$, the atom $A_j(x)$ has rank at most i, so by the IH, there is a derivation $\langle T'_j, \ell'_j \rangle$ of A_j at x. We obtain a derivation $\langle T', \ell' \rangle$ of B at x by setting $T' = \{\varepsilon\} \cup \{jw \mid w \in T'_j\}$, $\ell'(\varepsilon) = (B, x)$, and $\ell'(jw) = \ell'_j(w)$.

2. Assume that $B(x)$ is in $\mathcal{I}_{i+1}$ because of an application of Rule 2, that is, there is $\exists r.A \sqsubseteq B \in \mathcal{T}$ such that $x \in (\exists r.A)^{\mathcal{I}_i}$.

   As $x \in (\exists r.A)^{\mathcal{I}_i}$, there must exist some $y \in \Delta^{\mathcal{I}_i}$ such that $(x, y) \in r^{\mathcal{I}_i}$ and $y \in A^{\mathcal{I}_i}$. The atom $A(y)$ has rank at most i, so by the IH, there is a derivation $\langle T'', \ell'' \rangle$ of A at y. If $x \in \Delta^{\mathcal{I}_0}$, then $x \in \mathsf{Ind}(\mathcal{A}_{\langle T, \ell \rangle})$, and so we can define a derivation $\langle T', \ell' \rangle$ of B at x by setting $T' = \{\varepsilon\} \cup \{1w \mid w \in T''\}$, $\ell'(\varepsilon) = (B, x)$, and $\ell'(1w) = \ell''(w)$.

   Next consider the case in which $x \notin \Delta^{\mathcal{I}_0}$. Then $x \notin \mathsf{Ind}(\mathcal{A}_{\langle T, \ell \rangle})$, so by properness of T, there is a concept $E \in \mathsf{N_C} \cup \{\top\}$ such that $E^* \in \ell(x)$. Since $x \notin \Delta^{\mathcal{I}_0}$ but $x \notin \Delta^{\mathcal{I}_i}$, there is some $0 < j < i$ such that $x \in \Delta^{\mathcal{I}_j} \setminus \Delta^{\mathcal{I}_{j-1}}$. Since $\langle T, \ell \rangle$ is canonical, the element x was created due to an application of Rule 3 using a concept inclusion of the form $F \sqsubseteq \exists s.E$, so $x \in E^{\mathcal{I}_j}$. Applying the IH, we obtain a derivation $\langle T''', \ell''' \rangle$ of E at x. We can thus define a derivation $\langle T', \ell' \rangle$ of B at x by setting $T' = \{\varepsilon\} \cup \{1w \mid w \in T''\} \cup \{2w \mid w \in T'''\}$, $\ell'(\varepsilon) = (B, x)$, $\ell'(1w) = \ell''(w)$, and $\ell'(2w) = \ell'''(w)$.

3. Assume that $B(x)$ is in $\mathcal{I}_{i+1}$ because of an application of Rule 3 involving $A \sqsubseteq \exists r.B \in \mathcal{T}$, that is, there is some $y \in \Delta^{\mathcal{I}_i}$ such that $y \in A^{\mathcal{I}_i}$ and $(y, x) \in r^{\mathcal{I}_{i+1}} \setminus r^{\mathcal{I}_i}$.

   The atom $A(y)$ has rank at most i, so by the IH, there exists a derivation $\langle T'', \ell'' \rangle$ of A at y. Moreover, since x was created by applying the inclusion $A \sqsubseteq \exists r.B \in \mathcal{T}$ to y, the second condition of canonicity ensures that $B^* \in \ell(x)$. We can thus define a derivation of B at x by taking the tree $\langle T', \ell' \rangle$ with $T' = \{\varepsilon\} \cup \{1w \mid w \in T''\}$, $\ell'(\varepsilon) = (B, x)$, and $\ell'(1w) = \ell''(w)$.

4. Assume that $B(x) = \top(x)$ is in $\mathcal{I}_{i+1}$ because of an application of Rule 3 involving $A \sqsubseteq \exists r.E \in \mathcal{T}$ ($E \neq \top$), that is, there is some $y \in \Delta^{\mathcal{I}_i}$ such that $y \in A^{\mathcal{I}_i}$ and $(y, x) \in r^{\mathcal{I}_{i+1}} \setminus r^{\mathcal{I}_i}$.

   The atom $A(y)$ has rank at most i, so by the IH, there exists a derivation $\langle T'', \ell'' \rangle$ of A at y. Moreover, since x was created by applying the inclusion $A \sqsubseteq \exists r.E \in \mathcal{T}$ to y, the second condition of canonicity ensures that $E^* \in \ell(x)$. We can thus define a derivation of $\top$ at x by taking the tree $\langle T', \ell' \rangle$ with $T' = \{\varepsilon, 1\} \cup \{11w \mid w \in T''\}$, $\ell'(\varepsilon) = (\top, x)$, $\ell'(1) = (E, x)$, and $\ell'(11w) = \ell''(w)$.

5. Assume that $B(x)$ is in $\mathcal{I}_{i+1}$ because of an application of Rule 4, that is, $A \sqsubseteq \exists r.B \in \mathcal{T}$, $\mathsf{funct}(r) \in \mathcal{T}$, $y \in A^{\mathcal{I}_i}$, and $(y, x) \in r^{\mathcal{I}_i}$.

   The atom $A(y)$ has rank at most i, so by the IH, there exists a derivation $\langle T'', \ell'' \rangle$ of A at y. If $x \in \Delta^{\mathcal{I}_0}$, we obtain a derivation of B at x by taking the tree $\langle T', \ell' \rangle$ with $T' = \{\varepsilon\} \cup \{1w \mid w \in T''\}$, $\ell'(\varepsilon) = (B, x)$, and $\ell'(1w) = \ell''(w)$. If If $x \notin \Delta^{\mathcal{I}_0}$, then we can use the same

argument as in Point 2 to find a derivation $\langle T''', \ell'''\rangle$ of E at x. We then obtain a derivation $\langle T', \ell'\rangle$ of B at x by setting $T' = \{\varepsilon\} \cup \{1w \mid w \in T''\} \cup \{2w \mid w \in T'''\}$, $\ell'(\varepsilon) = (B, x)$, $\ell'(1w) = \ell''(w)$, and $\ell'(2w) = \ell'''(w)$.

❑

Lemma 48 *Let $\langle T, \ell\rangle$ be a proper $\Sigma_R \cup \Sigma_N$-labeled tree that is well-founded and such that $\mathcal{I}_{\langle T,\ell\rangle}$ is a model of $\mathcal{T}$. Then $\mathcal{I}_{\langle T,\ell\rangle}$ is a universal model of $\mathcal{T}$ and $\mathcal{A}_{\langle T,\ell\rangle}$.*

Proof. Assume that $\langle T, \ell\rangle$ is well-founded proper $\Sigma_R \cup \Sigma_N$-labeled tree and that $\mathcal{I}_{\langle T,\ell\rangle}$ is a model of $\mathcal{T}$. An *obligation* is a pair (A, x) such that $x \in T$ and $A \in \ell(x)$. For every obligation (A, x), choose a derivation $\langle T_{A,x}, \ell_{A,x}\rangle$ of A at x in $\langle T, \ell\rangle$ that is of minimal depth. For obligations (A_1, x_1),(A_2, x_2), we write $(A_1, x_1) \prec (A_2, x_2)$ if (A_1, x_1) occurs as a node label in $\langle T_{A_2,x_2}, \ell_{A_2,x_2}\rangle$.

Claim. The $\prec$ relation is acyclic.

Proof of claim. Assume to the contrary that there are obligations $(A_0, x_0), \ldots, (A_n, x_n)$ such that $(A_i, x_i) \prec (A_{i+1}, x_{i+1})$ for all $i \leq n$ and $(A_{n+1}, x_{n+1}) := (A_0, x_0)$. We may assume without loss of generality that for all $0 \leq i < j \leq n$, $(A_i, x_i) \neq (A_j, x_j)$, i.e., the obligations $(A_0, x_0), \ldots, (A_n, x_n)$ are pairwise distinct. Let k_i be the depth of $\langle T_{A_i,x_i}, \ell_{A_i,x_i}\rangle$ and ℓ_i the depth of the most shallow derivation of (A_i, x_i) contained in $\langle T_{A_{i+1},x_{i+1}}, \ell_{A_{i+1},x_{i+1}}\rangle$. Because $\langle T_{A_i,x_i}, \ell_{A_i,x_i}\rangle$ is of minimal depth, we have $k_i \leq \ell_i$. Moreover, we clearly also have $\ell_i \leq k_{i+1}$. We thus have shown that $k_0 = \ell_0 = \cdots = k_n = \ell_n$. Consequently, the derivation of (A_0, x_0) in $\langle T_{A_1,x_1}, \ell_{A_1,x_1}\rangle$ must start at the root of $\langle T_{A_i,x_i}, \ell_{A_i,x_i}\rangle$, which implies $(A_0, x_0) = (A_1, x_1)$ in contradiction to the fact that these obligations are distinct. This finishes the proof of the claim.

By the claim, we can assume w.l.o.g. that if in some chosen derivation $\langle T_{A,x}, \ell_{A,x}\rangle$, a node is labeled with (B, y), then the subtree of $\langle T_{A,x}, \ell_{A,x}\rangle$ rooted at this node is the chosen derivation $\langle T_{B,y}, \ell_{B,y}\rangle$ (uniformity assumption).

To prove that $\mathcal{I}_{\langle T,\ell\rangle}$ is a universal model of $\mathcal{T}$ and $\mathcal{A}_{\langle T,\ell\rangle}$, take a model $\mathcal{I}$ of $\mathcal{A}_{\langle T,\ell\rangle}$ and $\mathcal{T}$. We show that there is a homomorphism h from $\mathcal{I}_{\langle T,\ell\rangle}$ to $\mathcal{I}$, constructing h in a step-by-step fashion. To start, set $h(a) = a^{\mathcal{I}}$ for all individual names a that occur in $\mathcal{A}_{\langle T,\ell\rangle}$. Now and after each extension of h, we argue that

1. if $x \in A^{\mathcal{I}_{\langle T,\ell\rangle}}$ with A a concept name, $h(x)$ is defined and $\langle T_{A,x}, \ell_{A,x}\rangle$ uses only elements from the domain of h, then $h(x) \in A^{\mathcal{I}}$;
2. if $(x, y) \in r^{\mathcal{I}_{\langle T,\ell\rangle}}$ with r a role and $h(x), h(y)$ are defined, then $(h(x), h(y)) \in r^{\mathcal{I}}$;
3. if $(x, y) \in r^{\mathcal{I}_{\langle T,\ell\rangle}}$, y is a child of x in T, and $h(y)$ is defined, then $h(x)$ is also defined.

We start by observing that for the initial mapping h, Point 2 is trivial since $\mathcal{I}$ is a model of $\mathcal{A}_{\langle T,\ell\rangle}$ and all role edges in the restriction of $\mathcal{I}_{\langle T,\ell\rangle}$ to the domain of h are from $\mathcal{A}_{\langle T,\ell\rangle}$. For Point 3, we use the fact that if y is an individual in $\mathcal{A}_{\langle T,\ell\rangle}$ and x is the parent of y in T, then properness of T implies that x is also an individual in $\mathcal{A}_{\langle T,\ell\rangle}$ (and hence x belongs to the domain of h).

Point 1 is proved by induction on the depth of $\langle T_{A,x}, \ell_{A,x}\rangle$. For the induction start, consider depth zero. Then $A \in \Sigma \cup \{\top\}$, $x \in \mathsf{Ind}(\mathcal{A}_{\langle T,\ell\rangle})$, and $A(x) \in \mathcal{A}_{\langle T,\ell\rangle}$. Since $\mathcal{I}$ is a model of $\mathcal{A}_{\langle T,\ell\rangle}$ and by definition of h, we have $h(x) \in A^{\mathcal{I}}$.

Now for the induction step. Assume that $\langle T_{A,x}, \ell_{A,x}\rangle$ uses only elements from the domain of h. The definition of derivations gives rise to the following cases:

- $A^* \notin \ell(x)$, and there is a CI $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq A \in \mathcal{T}$ such that for $1 \leq i \leq n$, there is a child z' of z in $T_{A,x}$ with $\ell_{A,x}(z') = (A_i, x)$.

  For $1 \leq i \leq n$, let z_i be the child of z with $\ell_{A,x}(z_i) = (A_i, x)$. Then the subderivation of $\langle T_{A,x}, \ell_{A,x}\rangle$ rooted at z_i is the chosen derivation $\langle T_{A_i,x}, \ell_{A_i,x}\rangle$ of A_i at x. It follows that $\langle T_{A_i,x}, \ell_{A_i,x}\rangle$ only uses elements from the domain of h and its depth is strictly smaller than that of $\langle T_{A,x}, \ell_{A,x}\rangle$. We can therefore apply the induction hypothesis to get $h(x) \in A_i^{\mathcal{I}}$. Since $\mathcal{I}$ is a model of $\mathcal{T}$ and $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq A \in \mathcal{T}$, we obtain $h(x) \in A^{\mathcal{I}}$.

- $A^* \notin \ell(x)$, there is a CI $\exists r.A' \sqsubseteq A \in \mathcal{T}$ and a child z' of z in $T_{A,x}$ with $\ell_{A,x}(z') = (A', x')$ such that $(x, x') \in r^{\mathcal{I}_{\langle T,\ell\rangle}}$.

  The subderivation of $\langle T_{A,x}, \ell_{A,x}\rangle$ rooted at z' is the chosen derivation $\langle T_{A',x'}, \ell_{A',x'}\rangle$ of A' at x', and thus it contains only elements from the domain of h and has a strictly smaller depth than $\langle T_{A,x}, \ell_{A,x}\rangle$. We can thus use the IH to infer that $h(x') \in B^{\mathcal{I}}$, and we can use Point 2 to get $(h(x), h(x')) \in r^{\mathcal{I}}$. Since $\mathcal{I}$ is a model of $\mathcal{T}$ and $\exists r.A' \sqsubseteq A \in \mathcal{T}$, we have $h(x) \in A^{\mathcal{I}}$.

- $A^* \notin \ell(x)$, there is a CI $A' \sqsubseteq \exists r.A \in \mathcal{T}$ with $\mathsf{funct}(r) \in \mathcal{T}$ and a child z' of z in $T_{A,x}$ with $\ell_{A,x}(z') = (A', x')$ such that $(x', x) \in r^{\mathcal{I}_{\langle T,\ell\rangle}}$.

  As in the previous item, we can use the IH and Point 2 to get $h(x') \in B^{\mathcal{I}}$ and $(h(x), h(x')) \in r^{\mathcal{I}}$. Since $\mathcal{I}$ is a model of $\mathcal{T}$ and $\mathcal{T}$ contains both $A' \sqsubseteq \exists r.A$ and $\mathsf{funct}(r)$, it follows that $h(x) \in A^{\mathcal{I}}$.

- $A = \top$, $\top^* \notin \ell(x)$, $B^* \in \ell(x)$,and there is a child z of ε in $T_{A,x}$ with $\ell_{A,x}(z) = (B, x)$.

  This case is not applicable since if $B^* \in \ell(x)$, then $M \notin \ell(x)$, hence x is not in the domain of h.

- $A^* \in \ell(x)$, there is a CI $A' \sqsubseteq \exists r.A \in \mathcal{T}$, and there is a child z' of z in $T_{A,x}$ with $\ell_{A,x}(z') = (A', x')$ such that $(x', x) \in r^{\mathcal{I}_{\langle T,\ell\rangle}}$ and either (i) x is a child of x' in T, or (ii) x is a child of the root, $x' \in \mathsf{Ind}$, and $\{r, x'\} \subseteq \ell(x)$.

  This case is not applicable since if $A^* \in \ell(x)$, then $M \notin \ell(x)$, hence x is not in the domain of h.

To extend h, we first show that if h is not yet total, then there exists an edge $(\widehat{x}, \widehat{y}) \in r^{\mathcal{I}_{\langle T,\ell\rangle}}$ and a concept name A such that $h(\widehat{x})$ is defined, $h(\widehat{y})$ is undefined, $A^* \in \ell(\widehat{y})$ (and consequently $A \in \ell(\widehat{y})$), and $\langle T_{A,\widehat{y}}, \ell_{A,\widehat{y}}\rangle$ is such that all elements in it except the root node are in the domain of h.

Assume to the contrary that h is not total but there is no such edge, i.e., for every edge $(\widehat{x}, \widehat{y}) \in r^{\mathcal{I}_{\langle T,\ell\rangle}}$ such that $h(\widehat{x})$ is defined, $h(\widehat{y})$ is undefined, and $A^* \in \ell(\widehat{y})$, the derivation $\langle T_{A,\widehat{y}}, \ell_{A,\widehat{y}}\rangle$ contains a non-root node that is not in the domain of h. Pick one such edge $(\widehat{x}, \widehat{y}) \in r^{\mathcal{I}_{\langle T,\ell\rangle}}$ such that its associated derivation $\langle T_{A,\widehat{y}}, \ell_{A,\widehat{y}}\rangle$ is of minimal depth. Since $h(\widehat{x})$ is defined and $h(\widehat{y})$ is undefined, it follows from Point 3 that either $\widehat{x}$ is the parent of $\widehat{y}$ in T, or $\widehat{y}$ is a child of the root node and $\{\widehat{x}, r\} \in \ell(\widehat{y})$. Since derivation rule 6 is the only applicable rule when the node label contains A^* and by the formulation of that rule, there must thus be a CI $A' \sqsubseteq \exists r.A \in \mathcal{T}$ such that the unique child z of ε in $T_{A,\widehat{y}}$ satisfies $\ell_{A,\widehat{y}}(z) = (A', \widehat{x})$. Since $\widehat{x}$ is in the domain of h, the non-root node that

is not in the domain of h must be somewhere below z. Consequently, we find nodes $z_1, z_2 \in T_{A,\widehat{y}}$ such that z_2 is a successor of z_1 and the domain element $\widehat{x}'$ in $\ell_{A,\widehat{y}}(z_1)$ is in the domain of h, but the domain element $\widehat{y}'$ in $\ell_{A,\widehat{y}}(z_2)$ is not in the domain of h. By definition of the derivation rules, we must have $(\widehat{x}', \widehat{y}') \in s^{\mathcal{I}_{\langle T,\ell\rangle}}$ for some role s, and by Point 3, either $\widehat{y}'$ is a child of $\widehat{x}'$ in T, or $\widehat{y}'$ is a child of the root node and its label contains $\{\widehat{x}', s\}$. It follows that $\widehat{x}'$ is related to $\widehat{y}'$ by one of the derivation rules 4 and 5. Consequently, there is a $B^* \in \ell(\widehat{y}')$ such that $\langle T_{A,\widehat{y}}, \ell_{A,\widehat{y}}\rangle$ contains the obligation $(B, \widehat{y}')$. Thus, the edge $(\widehat{x}', \widehat{y}') \in s^{\mathcal{I}_{\langle T,\ell\rangle}}$ satisfies the above conditions and its associated derivation $\langle T_{B,\widehat{y}'}, \ell_{B,\widehat{y}'}\rangle$ is of strictly smaller depth than $\langle T_{A,\widehat{y}}, \ell_{A,\widehat{y}}\rangle$, contradicting the minimality of $\langle T_{A,\widehat{y}}, \ell_{A,\widehat{y}}\rangle$.

We now extend h using the edge $(\widehat{x}, \widehat{y}) \in r^{\mathcal{I}_{\langle T,\ell\rangle}}$ whose existence we have just established. By the definition of derivations, there is a CI $A' \sqsubseteq \exists r.A \in \mathcal{T}$ and a child z of ε in $T_{A,\widehat{y}}$ with $\ell_{A,\widehat{y}}(z) = (A', \widehat{x})$. Since all elements in $\langle T_{A,\widehat{y}}, \ell_{A,\widehat{y}}\rangle$ except the root node are in the domain of h, the subderivation of $\langle T_{A,\widehat{y}}, \ell_{A,\widehat{y}}\rangle$ rooted at z uses only elements from the domain of h. By our uniformity assumption, this derivation is just $\langle T_{A',\widehat{x}}, \ell_{A',\widehat{x}}\rangle$, and thus IH yields $h(\widehat{x}) \in A'^{\mathcal{I}_{\langle T,\ell\rangle}}$. Since $A' \sqsubseteq \exists r.A \in \mathcal{T}$, there is a $(\widehat{x}, d) \in r^{\mathcal{I}_{\langle T,\ell\rangle}}$ with $d \in A^{\mathcal{I}_{\langle T,\ell\rangle}}$. Set $h(\widehat{y}) = d$.

It remains to show that Points 1 and 2 are satisfied for the extended h (Point 3 obviously is). We start with Point 2. Assume that $(x, y) \in s^{\mathcal{I}_{\langle T,\ell\rangle}}$. If $h(x)$ and $h(y)$ were defined already before the extension of h, we are done. Otherwise, by construction of h we must have $(x, y, s) = (\widehat{x}, \widehat{y}, r)$ (or $(x, y, s) = (\widehat{y}, \widehat{x}, r^-)$, which is equivalent). By the choice of $h(\widehat{y})$, we have $(h(\widehat{x}), h(\widehat{y})) \in r^{\mathcal{I}}$, hence $(x, y) \in s^{\mathcal{I}}$. Point 1 is proved by induction on the depth of $A(x)$ as for the initial version of h. The induction start is exactly the same, and for the induction step, the only cases that differ are the following ones:

- $A^* \notin \ell(x)$, $B^* \in \ell(x)$, $A = \top$, and there is a child z of ε in $T_{A,x}$ with $\ell_{A,x}(z) = (B, x)$.

  Immediate since $A = \top$.

- $A^* \in \ell(x)$, there is a CI $A' \sqsubseteq \exists r.A \in \mathcal{T}$, and there is a child z' of z in $T_{A,x}$ with $\ell_{A,x}(z') = (A', x')$ such that $(x', x) \in r^{\mathcal{I}_{\langle T,\ell\rangle}}$ and either (i) x is a child of x' in T, or (ii) x is a child of the root, $x' \in \mathsf{Ind}$, and $\{r, x'\} \subseteq \ell(x)$.

  Since $A^* \in \ell(x)$, we have $x \notin \mathsf{Ind}(\mathcal{A}_{\langle T,\ell\rangle})$, so x must have been introduced into the domain of h during the examination of edge $(x', x) \in r^{\mathcal{I}_{\langle T,\ell\rangle}}$. Since the child z' is labelled (A', x'), we will use the CI $A' \sqsubseteq \exists r.A \in \mathcal{T}$ and choose $h(x)$ such that $h(x) \in A^{\mathcal{I}_{\langle T,\ell\rangle}}$.

❏