Inseparability and Conservative Extensions of Description Logic Ontologies: A Survey

Elena Botoeva[1], Boris Konev[2], Carsten Lutz[3], Vladislav Ryzhikov[1], Frank Wolter[2], and Michael Zakharyaschev[4]

[1] Free University of Bozen-Bolzano, Italy
{botoeva,ryzhikov}@inf.unibz.it
[2] University of Liverpool, UK
{konev,wolter}@liverpool.ac.uk
[3] University of Bremen, Germany
clu@informatik.uni-bremen.de
[4] Birkbeck, University of London, UK
michael@dcs.bbk.ac.uk

Abstract. The question whether an ontology can safely be replaced by another, possibly simpler, one is fundamental for many ontology engineering and maintenance tasks. It underpins, for example, ontology versioning, ontology modularization, forgetting, and knowledge exchange. What 'safe replacement' means depends on the intended application of the ontology. If, for example, it is used to query data, then the answers to any relevant ontology-mediated query should be the same over any relevant data set; if, in contrast, the ontology is used for conceptual reasoning, then the entailed subsumptions between concept expressions should coincide. This gives rise to different notions of ontology *inseparability* such as query inseparability and concept inseparability, which generalize corresponding notions of conservative extensions. We survey results on various notions of inseparability in the context of description logic ontologies, discussing their applications, useful model-theoretic characterizations, algorithms for determining whether two ontologies are inseparable (and, sometimes, for computing the difference between them if they are not), and the computational complexity of this problem.

1 Introduction

Description logic (DL) ontologies provide a common vocabulary for a domain of interest together with a formal modeling of the semantics of the vocabulary items (concept names and role names). In modern information systems, they are employed to capture domain knowledge and to promote interoperability. Ontologies can become large and complex as witnessed, for example, by the widely used healthcare ontology Snomed CT, which contains more than 300,000 concept names, and the National Cancer Institute (NCI) Thesaurus ontology, which contains more than 100,000 concept names. Engineering, maintaining and deploying such ontologies is challenging and labour intensive; it crucially relies on extensive tool support for tasks such as ontology versioning, ontology modularization, ontology summarization, and forgetting parts of an ontology. At the core of many of these tasks lie notions of *inseparability* of two ontologies, indicating that inseparable ontologies can safely be replaced by each other for the task at hand. The aim of this article is to survey the current research on inseparability of DL ontologies. We present and discuss different types of inseparability, their applications and interrelation, model-theoretic characterizations, as well as results on the decidability and computational complexity of inseparability.

The exact formalization of when an ontology 'can safely be replaced by another one' (that is, of inseparability) depends on the task for which the ontology is to be used. As we are generally going to abstract away from the syntactic presentation of an ontology, a natural first candidate for the notion of inseparability between two ontologies is their logical equivalence. However, this can be an unnecessarily strong requirement for most applications since also ontologies that are not logically equivalent can be replaced by each other without adverse effects. This is due to two main reasons. First, applications of ontologies often make use of only a fraction of the vocabulary items. As an example, consider Snomed CT, which contains a vocabulary for a multitude of domains related to health case, including clinical findings, symptoms, diagnoses, procedures, body structures, organisms, pharmaceuticals, and devices. In a concrete application such as storing electronic patient records, only a small part of this vocabulary is going to be used. Thus, two ontologies should be separable only if they differ with respect to the *relevant* vocabulary items. Consequently, all our inseparability notions will be parameterized by a signature (set of concept and role names) Σ; when we want to emphasize Σ, we speak of Σ-inseparability. Second, even for the relevant vocabulary items, many applications do not rely on all details of the semantics provided by the ontology. For example, if an ontology is employed for conjunctive query answering over data sets that use vocabulary items from the ontology, then only the existential positive aspects of the semantics are relevant since the queries are positive existential, too.

A fundamental decision to be taken when defining an inseparability notion is whether the definition should be model-theoretic or in terms of logical consequences. Under the first approach, two ontologies are inseparable when the reducts of their models to the signature Σ coincide. We call the resulting inseparability notion *model inseparability.* Under the second approach, two ontologies are inseparable when they have the same logical consequences in the signature Σ. This actually gives rise to potentially many notions of inseparability since we can vary the logical language in which the logical consequences are formulated. Choosing the same language as the one used for formulating the ontologies results in what we call *concept inseparability*, which is appropriate when the ontologies are used for conceptual reasoning. Choosing a logical language that is based on database-style queries results in notions of *query inseparability*, which are appropriate for querying applications. Model inseparability implies all the resulting consequence-based notions of inseparability, but the converse does not hold for all standard DLs. The notion of query inseparability suggests some additional aspects. In particular, this type of inseparability is important both for ontologies that contain data as an integral part (*knowledge bases or KBs*, in DL parlance) and for those that do not (*TBoxes*, in DL parlance) and are maintained independently of the data. In the latter case, two TBoxes should be regarded as inseparable if they give the same answers to any relevant query *for any possible data*. One might then even want to work with two signatures: one for the data and one for the query. It turns out that, for both KBs and TBoxes, one obtains notions of inseparability that behave very differently from concept inseparability.

Inseparability generalizes conservative extensions, as known from classical logic. In fact, conservative extensions can also be defined in a model-theoretic and in a consequence-based way, and they correspond to the special case of inseparability where one ontology is syntactically contained in the other and the signature is the set of vocabulary items in the smaller ontology. Note that none of these two additional assumptions is appropriate for many applications of inseparability, such as ontology versioning. Instead of directly working with inseparability, we will often consider corresponding notions of entailment which, intuitively, is inseparability 'in one direction'; for example, two ontologies are concept Σ-inseparable if and only if they concept Σ-entail each other. Thus, one could say that an ontology concept (or model) entails another ontology if it is *sound* to replace the former by the latter in applications for which concept inseparability is the 'right' inseparability notion. Algorithms and complexity upper bounds are most general when established for entailment instead of inseparability, as they carry over to inseparability and conservative extensions. Similarly, lower bounds for conservative extensions imply lower bounds for inseparability and for entailment.

In this survey, we provide an in-depth discussion of four inseparability relations, as indicated above. For TBoxes, we look at Σ-concept inseparability (do two TBoxes entail the same concept inclusions over Σ?), Σ-model inseparability (do the Σ-reducts of the models of two TBoxes coincide?), and (Σ_1, Σ_2)-$\mathcal{Q}$-inseparability (do the answers given by two TBoxes coincide for all Σ_1-ABoxes and all Σ_2-queries from the class $\mathcal{Q}$ of queries?). Here, we usually take $\mathcal{Q}$ to be the class of conjunctive queries (CQs) and unions thereof (UCQs), but some smaller classes of queries are considered as well. For KBs, we consider Σ-$\mathcal{Q}$-inseparability (do the answers to Σ-queries in $\mathcal{Q}$ given by two KBs coincide?). When discussing proof techniques in detail, we focus on the standard expressive DL $\mathcal{ALC}$ and tractable DLs from the $\mathcal{EL}$ and *DL-Lite* families. We shall, however, also mention results for extensions of $\mathcal{ALC}$ and other DLs such as Horn-$\mathcal{ALC}$.

The structure of this survey is as follows. In Section 2, we introduce description logics. In Section 3, we introduce an abstract notion of inseparability and discuss applications of inseparability in ontology versioning, refinement, reuse, modularity, the design of ontology mappings, knowledge base exchange, and forgetting. In Section 4, we discuss concept inseparability. We focus on the description logics $\mathcal{EL}$ and $\mathcal{ALC}$ and give model-theoretic characterizations of Σ-concept inseparability which are then used to devise automata-based approaches

Name	Syntax	Semantics
top concept	$\top$	$\Delta^{\mathcal{I}}$
bottom concept	$\bot$	$\emptyset$
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	$\exists r.C$	$\{d \in \Delta^{\mathcal{I}} \mid \exists e \in C^{\mathcal{I}}\ (d,e) \in r^{\mathcal{I}}\}$
universal restriction	$\forall r.C$	$\{d \in \Delta^{\mathcal{I}} \mid \forall e \in \Delta^{\mathcal{I}}\ ((d,e) \in r^{\mathcal{I}} \rightarrow e \in C^{\mathcal{I}})\}$

Table 1. Syntax and semantics of $\mathcal{ALC}$.

to deciding concept inseparability. We also present polynomial time algorithms for acyclic $\mathcal{EL}$ TBoxes. In Section 5, we discuss model inseparability. We show that it is undecidable even in simple cases, but that by restricting the signature Σ to concept names, it often becomes decidable. We also consider model inseparability from the empty TBox, which is important for modularization and locality-based approximations of model inseparability. In Section 6, we discuss query inseparability between KBs. We develop model-theoretic criteria for query inseparability and use them to obtain algorithms for deciding query inseparability between KBs and their complexity. We consider description logics from the $\mathcal{EL}$ and *DL-Lite* families, as well as $\mathcal{ALC}$ and its Horn fragment. In Section 7, we consider query inseparability between TBoxes and analyse in how far the techniques developed for KBs can be generalized to TBoxes. We again consider a wide range of DLs. Finally, in Section 8 we discuss further inseparability relations, approximation algorithms and the computation of representatives of classes of inseparable TBoxes.

2 Description Logic

In description logic, knowledge is represented using concepts and roles that are inductively defined starting from a set $\mathsf{N_C}$ of *concept names* and a set $\mathsf{N_R}$ of *role names*, and using a set of concept and role constructors [7]. Different sets of concept and role constructors give rise to different DLs.

We start by introducing the description logic $\mathcal{ALC}$. The concept constructors available in $\mathcal{ALC}$ are shown in Table 1, where r is a role name and C and D are concepts. A concept built from these constructors is called an *$\mathcal{ALC}$-concept.* $\mathcal{ALC}$ does not have any role constructors. An *$\mathcal{ALC}$ TBox* is a finite set of *$\mathcal{ALC}$ concept inclusions* (CIs) of the form $C \sqsubseteq D$ and *$\mathcal{ALC}$ concept equivalences* (CEs) of the form $C \equiv D$. (A CE $C \equiv D$ can be regarded as an abbreviation for the two CIs $C \sqsubseteq D$ and $D \sqsubseteq C$.) The *size* $|\mathcal{T}|$ of a TBox $\mathcal{T}$ is the number of occurrences of symbols in $\mathcal{T}$.

DL semantics is given by *interpretations* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ in which the *domain* $\Delta^{\mathcal{I}}$ is a non-empty set and the *interpretation function* $\cdot^{\mathcal{I}}$ maps each concept

Name	Syntax	Semantics	Identifier
number restrictions	$(\leqslant n\ r\ C)$	$\{d \mid \#\{e \mid (d,e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\} \leq n\}$	$\mathcal{Q}$
	$(\geqslant n\ r\ C)$	$\{d \mid \#\{e \mid (d,e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\} \geq n\}$	
inverse role	r^-	$\{(d,e) \mid (e,d) \in r^{\mathcal{I}}\}$	$\mathcal{I}$
universal role	u	$\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$	$\cdot^u$
role inclusions (RIs)	$r \sqsubseteq s$	$r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$	$\mathcal{H}$

Table 2. Additional constructors: syntax and semantics.

name $A \in \mathsf{N_C}$ to a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, and each role name $r \in \mathsf{N_R}$ to a binary relation $r^{\mathcal{I}}$ on $\Delta^{\mathcal{I}}$. The extension of $\cdot^{\mathcal{I}}$ to arbitrary concepts is defined inductively as shown in the third column of Table 1. We say that an interpretation $\mathcal{I}$ *satisfies* a CI $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, and that $\mathcal{I}$ is a *model* of a TBox $\mathcal{T}$ if it satisfies all the CIs in $\mathcal{T}$. A TBox is *consistent* (or *satisfiable*) if it has a model. A concept C is *satisfiable w.r.t.* $\mathcal{T}$ if there exists a model $\mathcal{I}$ of $\mathcal{T}$ such that $C^{\mathcal{I}} \neq \emptyset$. A concept C is *subsumed by a concept D w.r.t.* $\mathcal{T}$ ($\mathcal{T} \models C \sqsubseteq D$, in symbols) if every model $\mathcal{I}$ of $\mathcal{T}$ satisfies the CI $C \sqsubseteq D$. For TBoxes $\mathcal{T}_1, \mathcal{T}_2$, we write $\mathcal{T}_1 \models \mathcal{T}_2$ and say that $\mathcal{T}_1$ *entails* $\mathcal{T}_2$ if $\mathcal{T}_1 \models \alpha$ for all $\alpha \in \mathcal{T}_2$. TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ are *logically equivalent* if they have the same models. This is the case if and only if $\mathcal{T}_1$ entails $\mathcal{T}_2$ and vice versa.

A *signature* Σ is a finite set of concept and role names. The *signature* $\mathsf{sig}(C)$ of a concept C is the set of concept and role names that occur in C, and likewise for TBoxes $\mathcal{T}$, CIs $C \sqsubseteq D$, assertions $r(a,b)$ and $A(a)$, ABoxes $\mathcal{A}$, KBs $\mathcal{K}$, UCQs $\boldsymbol{q}$. Note that the universal role is not regarded as a role name, and so does not belong in any signature. Similarly, individual names are not in any signature and, in particular, not in the signature of an assertion, ABox, or KB. We are often interested in concepts, TBoxes, KBs, and ABoxes that are formulated using a specific signature. Therefore, we talk of a Σ-TBox $\mathcal{T}$ if $\mathsf{sig}(\mathcal{T}) \subseteq \Sigma$, and likewise for Σ-concepts, etc.

There are several extensions of $\mathcal{ALC}$ relevant for this paper, which fall into three categories: extensions with (*i*) additional concept constructors, (*ii*) additional role constructors, and (*iii*) additional types of statements in TBoxes. These extensions are detailed in Table 2, where $\#\mathcal{X}$ denotes the size of a set $\mathcal{X}$ and double horizontal lines delineate different types of extensions. The last column gives an identifier for each extension, which is simply appended to the name $\mathcal{ALC}$ for constructing extensions of $\mathcal{ALC}$. For example, $\mathcal{ALC}$ extended with number restrictions, inverse roles, and the universal role is denoted by $\mathcal{ALCQI}^u$.

We next define a number of syntactic fragments of $\mathcal{ALC}$ and its extensions, which often have dramatically lower computational complexity. The fragment of $\mathcal{ALC}$ obtained by disallowing the constructors $\bot$, $\neg$, $\sqcup$ and $\forall$ is known as $\mathcal{EL}$. Thus, $\mathcal{EL}$ concepts are constructed using $\top$, $\sqcap$ and $\exists$ only [6]. We also consider extensions of $\mathcal{EL}$ with the constructors in Table 2. For example, $\mathcal{ELI}^u$ denotes the extension of $\mathcal{EL}$ with inverse roles and the universal role. The fragments of

$\mathcal{ALCI}$ and $\mathcal{ALCHI}$, in which CIs are of the form

$$B_1 \sqsubseteq B_2 \qquad \text{and} \qquad B_1 \sqcap B_2 \sqsubseteq \bot,$$

and the B_i are concept names, $\top$, $\bot$ or $\exists r.\top$, are denoted by *DL-Lite*$_{core}$ and *DL-Lite*$^{\mathcal{H}}_{core}$ (or *DL-Lite*$_{\mathcal{R}}$), respectively [19, 4].

Example 1. The CI $\forall \mathsf{childOf}^-.\mathsf{Tall} \sqsubseteq \mathsf{Tall}$ (saying that everyone with only tall parents is also tall) is in $\mathcal{ALCI}$ but not in $\mathcal{ALC}$, $\mathcal{EL}$ or *DL-Lite*$^{\mathcal{H}}_{core}$. The RI $\mathsf{childOf}^- \sqsubseteq \mathsf{parentOf}$ is in both $\mathcal{ALCHI}$ and *DL-Lite*$^{\mathcal{H}}_{core}$.

$\mathcal{EL}$ and the *DL-Lite* logics introduced above are examples of Horn DLs, that is, fragments of DLs in the $\mathcal{ALC}$ family that restrict the syntax in such a way that conjunctive query answering (see below) becomes tractable in data complexity. A few additional Horn DLs have become important in recent years. Following [46, 49], we say that a concept C occurs positively in C itself and, if C occurs positively (negatively) in C', then

- C occurs positively (respectively, negatively) in $C' \sqcup D$, $C' \sqcap D$, $\exists r.C'$, $\forall r.C'$, $D \sqsubseteq C'$, and
- C occurs negatively (respectively, positively) in $\neg C'$ and $C' \sqsubseteq D$.

Now, we call a TBox $\mathcal{T}$ *Horn* if no concept of the form $C \sqcup D$ occurs positively in $\mathcal{T}$, and no concept of the form $\neg C$ or $\forall R.C$ occurs negatively in $\mathcal{T}$. For any DL $\mathcal{L}$ from the $\mathcal{ALC}$ family introduced above (e.g., $\mathcal{ALCHI}$), the DL *Horn-*$\mathcal{L}$ only allows for Horn TBoxes in $\mathcal{L}$. Note that $\forall \mathsf{childOf}^-.\mathsf{Tall}$ occurs negatively in the CI α from Example 1, and so the TBox $\mathcal{T} = \{\alpha\}$ is not Horn.

TBoxes $\mathcal{T}$ used in practice often turn out to be *acyclic* in the following sense:

- all CEs in $\mathcal{T}$ are of the form $A \equiv C$ (*concept definitions*) and all CIs in $\mathcal{T}$ are of the form $A \sqsubseteq C$ (*primitive concept inclusions*), where A is a concept name;
- no concept name occurs more than once on the left-hand side of a statement in $\mathcal{T}$;
- $\mathcal{T}$ contains no cyclic definitions, as detailed below.

Let $\mathcal{T}$ be a TBox that contains only concept definitions and primitive concept inclusions. The relation $\prec_{\mathcal{T}} \subseteq \mathsf{N_C} \times \mathsf{sig}(\mathcal{T})$ is defined by setting $A \prec_{\mathcal{T}} X$ if there exists a TBox statement $A \bowtie C$ such that X occurs in C, where $\bowtie$ ranges over $\{\sqsubseteq, \equiv\}$. A concept name A *depends* on a symbol $X \in \mathsf{N_C} \cup \mathsf{N_R}$ if $A \prec^+_{\mathcal{T}} X$, where $\cdot^+$ denotes transitive closure. We use $\mathsf{depend}_{\mathcal{T}}(A)$ to denote the set of all symbols X such that A depends on X. We can now make precise what it means for $\mathcal{T}$ to *contain no cyclic definitions*: $A \notin \mathsf{depend}_{\mathcal{T}}(A)$, for all $A \in \mathsf{N_C}$. Note that the TBox $\mathcal{T} = \{\alpha\}$ with α from Example 1 is cyclic.

In DL, data is represented in the form of ABoxes. To introduce ABoxes, we fix a set $\mathsf{N_I}$ of *individual names*, which correspond to constants in first-order logic. An *assertion* is an expression of the form $A(a)$ or $r(a, b)$, where A is a concept name, r a role name, and a, b individual names. An *ABox* $\mathcal{A}$ is just a finite set of assertions. We call the pair $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ of a TBox $\mathcal{T}$ in a DL $\mathcal{L}$ and an ABox $\mathcal{A}$ an $\mathcal{L}$ *knowledge base* (KB, for short). By $\mathsf{ind}(\mathcal{A})$ and $\mathsf{ind}(\mathcal{K})$, we denote the set of individual names in $\mathcal{A}$ and $\mathcal{K}$, respectively.

To interpret ABoxes $\mathcal{A}$, we consider interpretations $\mathcal{I}$ that map all individual names $a \in \mathsf{ind}(\mathcal{A})$ to elements $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ in such a way that $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ if $a \neq b$ (thus, we adopt the *unique name assumption*). We say that $\mathcal{I}$ *satisfies* an assertion $A(a)$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$, and $r(a, b)$ if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$. It is a *model* of an ABox $\mathcal{A}$ if it satisfies all assertions in $\mathcal{A}$, and of a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ if it is a model of both $\mathcal{T}$ and $\mathcal{A}$. We say that $\mathcal{K}$ is *consistent* (or *satisfiable*) if it has a model. We use the terminology introduced for TBoxes for KBs as well. For example, KBs $\mathcal{K}_1$ and $\mathcal{K}_2$ are *logically equivalent* if they have the same models (or, equivalently, entail each other).

We next introduce query answering for KBs, beginning with conjunctive queries [39, 18, 20]. An *atom* is of the form $A(x)$ or $r(x, y)$, where x, y are from a set of *variables* $\mathsf{N_V}$, A is a concept name, and r a role name. A *conjunctive query* (or CQ) is an expression of the form $\boldsymbol{q}(\boldsymbol{x}) = \exists \boldsymbol{y}\, \varphi(\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{x}$ and $\boldsymbol{y}$ are disjoint sequences of variables and φ is a conjunction of atoms that only contain variables from $\boldsymbol{x} \cup \boldsymbol{y}$—we (ab)use set-theoretic notation for sequences where convenient. We often write $A(x) \in \boldsymbol{q}$ and $r(x, y) \in \boldsymbol{q}$ to indicate that $A(x)$ and $r(x, y)$ are conjuncts of φ. We call a CQ $\boldsymbol{q}$ *rooted* (rCQ) if every $y \in \boldsymbol{y}$ is connected to some $x \in \boldsymbol{x}$ by a path in the undirected graph whose nodes are the variables in $\boldsymbol{q}$ and edges are the pairs $\{u, v\}$ with $r(u, v) \in \boldsymbol{q}$, for some r. A *union of CQs* (UCQ) is a disjunction $\boldsymbol{q}(\boldsymbol{x}) = \bigvee_i \boldsymbol{q}_i(\boldsymbol{x})$ of CQs $\boldsymbol{q}_i(\boldsymbol{x})$ with the same *answer variables* $\boldsymbol{x}$; it is *rooted* (rUCQ) if all the $\boldsymbol{q}_i$ are rooted. If the sequence $\boldsymbol{x}$ is empty, $\boldsymbol{q}(\boldsymbol{x})$ is called a *Boolean* CQ or UCQ.

Given a UCQ $\boldsymbol{q}(\boldsymbol{x}) = \bigvee_i \boldsymbol{q}_i(\boldsymbol{x})$ and a KB $\mathcal{K}$, a sequence $\boldsymbol{a}$ of individual names from $\mathcal{K}$ of the same length as $\boldsymbol{x}$ is called a *certain answer to* $\boldsymbol{q}(\boldsymbol{x})$ *over* $\mathcal{K}$ if, for every model $\mathcal{I}$ of $\mathcal{K}$, there exist a CQ $\boldsymbol{q}_i$ in $\boldsymbol{q}$ and a map (homomorphism) h of its variables to $\Delta^{\mathcal{I}}$ such that

- if x is the j-th element of $\boldsymbol{x}$ and a the j-th element of $\boldsymbol{a}$, then $h(x) = a^{\mathcal{I}}$;
- $A(z) \in \boldsymbol{q}$ implies $h(z) \in A^{\mathcal{I}}$, and $r(z, z') \in \boldsymbol{q}$ implies $(h(z), h(z')) \in r^{\mathcal{I}}$.

If this is the case, we write $\mathcal{K} \models \boldsymbol{q}(\boldsymbol{a})$. For a Boolean UCQ $\boldsymbol{q}$, we also say that the certain answer over $\mathcal{K}$ is 'yes' if $\mathcal{K} \models \boldsymbol{q}$ and 'no' otherwise. *CQ* or *UCQ answering* means to decide, given a CQ or UCQ $\boldsymbol{q}(\boldsymbol{x})$, a KB $\mathcal{K}$ and a tuple $\boldsymbol{a}$ from $\mathsf{ind}(\mathcal{K})$, whether $\mathcal{K} \models \boldsymbol{q}(\boldsymbol{a})$.

3 Inseparability

Since there is no single inseparability relation between ontologies that is appropriate for all applications, we start by identifying basic properties that any semantic notion of inseparability between TBoxes or KBs should satisfy. We also introduce notation that will be used throughout the survey and discuss a few applications of inseparability.

For uniformity, we assume that the term 'ontology' refers to both TBoxes and KBs.

Definition 1 (inseparability). Let $\mathcal{S}$ be the set of ontologies (either TBoxes or KBs) formulated in a description logic $\mathcal{L}$. A map that assigns to each signature Σ an equivalence relation $\equiv_\Sigma$ on $\mathcal{S}$ is an *inseparability relation on* $\mathcal{S}$ if the following conditions hold:

(i) if $\mathcal{O}_1$ and $\mathcal{O}_2$ are logically equivalent, then $\mathcal{O}_1 \equiv_\Sigma \mathcal{O}_2$, for all signatures Σ and $\mathcal{O}_1, \mathcal{O}_2 \in \mathcal{S}$;
(ii) $\Sigma_1 \subseteq \Sigma_2$ implies $\equiv_{\Sigma_1} \supseteq \equiv_{\Sigma_2}$, for all finite signatures Σ_1 and Σ_2.

By condition (i), an inseparability relations does not depend on the syntactic presentation of an ontology, but only on its semantics. Condition (ii) formalizes the requirement that if the set of relevant symbols increases ($\Sigma_2 \supseteq \Sigma_1$), then more ontologies become separable. Depending on the intended application, additional properties may also be required. For example, we refer the reader to [55] for a detailed discussion of robustness properties that are relevant for applications to modularity. We illustrate inseparability relations by three very basic examples.

Example 2. (1) Let $\mathcal{S}$ be the set of ontologies formulated in a description logic $\mathcal{L}$, and let $\mathcal{O}_1 \equiv_{\mathsf{equiv}} \mathcal{O}_2$ if and only if $\mathcal{O}_1$ and $\mathcal{O}_2$ are logically equivalent, for any $\mathcal{O}_1, \mathcal{O}_2 \in \mathcal{S}$. Then $\equiv_{\mathsf{equiv}}$ is an inseparability relation that does not depend on the concrete signature. It is the finest inseparability relation possible. The inseparability relations considered in this survey are more coarse.

(2) Let $\mathcal{S}$ be the set of KBs in a description logic $\mathcal{L}$, and let $\mathcal{K}_1 \equiv_{\mathsf{sat}} \mathcal{K}_2$ if and only if $\mathcal{K}_1$ and $\mathcal{K}_2$ are equisatisfiable, for any $\mathcal{K}_1, \mathcal{K}_2 \in \mathcal{S}$. Then $\equiv_{\mathsf{sat}}$ is another inseparability relation that does not depend on the concrete signature. It has two equivalence classes—the satisfiable KBs and the unsatisfiable KBs—and is not sufficiently fine-grained for most applications.

(3) Let $\mathcal{S}$ be the set of TBoxes in a description logic $\mathcal{L}$, and let $\mathcal{T}_1 \equiv_\Sigma^{\mathsf{hierarchy}} \mathcal{T}_2$ if and only if

$$\mathcal{T}_1 \models A \sqsubseteq B \quad \Longleftrightarrow \quad \mathcal{T}_2 \models A \sqsubseteq B, \quad \text{for all concept names } A, B \in \Sigma.$$

Then each relation $\equiv_\Sigma^{\mathsf{hierarchy}}$ is an inseparability relation. It distinguishes between two TBoxes if and only if they do not entail the same subsumption hierarchy over the concept names in Σ, and it is appropriate for applications that are only concerned with subsumption hierarchies such as producing a systematic catalog of vocabulary items, which is in fact the prime use of SNOMED CT[5].

As discussed in the introduction, the inseparability relations considered in this paper are more sophisticated than those in Example 2. Details are given in the subsequent sections. We remark that some versions of query inseparability that we are going to consider are, strictly speaking, not covered by Definition 1 since two signatures are involved (one for the query and one for the data). However, it is easy to extend Definition 1 accordingly.

We now present some important applications of inseparability.

[5] http://www.ihtsdo.org/snomed-ct

Versioning. Maintaining and updating ontologies is very difficult without tools that support versioning. One can distinguish *three* approaches to versioning [53]: versioning based on syntactic difference (syntactic diff), versioning based on structural difference (structural diff), and versioning based on logical difference (logical diff). The *syntactic diff* underlies most existing version control systems used in software development [24] such as RCS, CVS, SCCS. It works with text files and represents the difference between versions as blocks of text present in one version but not in the other. As observed in [82], ontology versioning cannot rely on a purely syntactic diff operation since many syntactic differences (e.g., the order of ontology axioms) do not affect the semantics. The *structural diff* extends the syntactic diff by taking into account information about the structure of ontologies. Its main characteristic is that it regards ontologies as structured objects, such as an *is-a* taxonomy [82], a set of RDF triples [51] or a set of class defining axioms [87, 47]. Though helpful, the structural diff still has no unambiguous semantic foundation and is syntax dependent. Moreover, it is tailored towards applications of ontologies that are based on the induced concept hierarchy (or some mild extension thereof), but does not capture other applications such as querying data under ontologies. In contrast, the *logical diff* [58, 60] completely abstracts away from the presentation of the ontology and regards two versions of an ontology as identical if they are inseparable with respect to an appropriate inseparability relation such as concept inseparability or query inseparability. The result of the logical diff is then presented in terms of witnesses for separability.

Ontology refinement. When extending an ontology with new concept inclusions or other statements, one usually wants to preserve the semantics of a large part Σ of its vocabulary. For example, when extending SNOMED CT with 50 additional concept names on top of the more than 300K existing ones, one wants to ensure that the meaning of unrelated parts of the vocabulary does not change. This preservation problem is formalized by demanding that the original ontology $\mathcal{O}_{\mathsf{original}}$ and the extended ontology $\mathcal{O}_{\mathsf{original}} \cup \mathcal{O}_{\mathsf{add}}$ are Σ-inseparable (for an appropriate notion of inseparability) [36]. It should be noted that ontology refinement can be regarded as a versioning problem as discussed above, where $\mathcal{O}_{\mathsf{original}}$ and $\mathcal{O}_{\mathsf{original}} \cup \mathcal{O}_{\mathsf{add}}$ are versions of an ontology that have to be compared.

Ontology reuse. A frequent operation in ontology engineering is to import an existing ontology $\mathcal{O}_{\mathsf{im}}$ into an ontology $\mathcal{O}_{\mathsf{host}}$ that is currently being developed, with the aim of reusing the vocabulary of $\mathcal{O}_{\mathsf{im}}$. Consider, for example, a host ontology $\mathcal{O}_{\mathsf{host}}$ describing research projects that imports an ontology $\mathcal{O}_{\mathsf{im}}$, which defines medical terms Σ to be used in the definition of research projects in $\mathcal{O}_{\mathsf{host}}$. Then one typically wants to use the medical terms Σ exactly as defined in $\mathcal{O}_{\mathsf{im}}$. However, using those terms to define concepts in $\mathcal{O}_{\mathsf{host}}$ might have unexpected consequences also for the terms in $\mathcal{O}_{\mathsf{im}}$, that is, it might 'damage' the modeling of those terms. To avoid this, one wants to ensure that $\mathcal{O}_{\mathsf{host}} \cup \mathcal{O}_{\mathsf{im}}$ and $\mathcal{O}_{\mathsf{im}}$ are Σ-inseparable [25]. Again, this can be regarded as a versioning problem for the ontology $\mathcal{O}_{\mathsf{im}}$.

Modularity. Modular ontologies and the extraction of modules are an important ontology engineering challenge [97, 64]. Understanding Σ-inseparability of ontologies is crucial for most approaches to this problem. For example, a very natural and popular definition of a module $\mathcal{M}$ of an ontology $\mathcal{O}$ demands that $\mathcal{M} \subseteq \mathcal{O}$ and that $\mathcal{M}$ is Σ-inseparable from $\mathcal{O}$ for the signature Σ of $\mathcal{M}$ (called self-contained module). Under this definition, the ontology $\mathcal{O}$ can be safely replaced by the module $\mathcal{M}$ in the sense specified by the inseparability relation and as far as the signature Σ of $\mathcal{M}$ is concerned. A stronger notion of module (called depleting module [61]) demands that $\mathcal{M} \subseteq \mathcal{O}$ and that $\mathcal{O} \setminus \mathcal{M}$ is Σ-inseparable from the empty ontology for the signature Σ of $\mathcal{M}$. The intuition is that the ontology statements outside of $\mathcal{M}$ should not say anything non-tautological about signature items in the module $\mathcal{M}$.

Ontology mappings. The construction of mappings (or alignments) between ontologies is an important challenge in ontology engineering and integration [95]. Given two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ in different signatures Σ_1 and Σ_2, the problem is to align the vocabulary items in Σ_1 with those in Σ_2 using a TBox $\mathcal{T}_{12}$ that states logical relationships between Σ_1 and Σ_2. For example, $\mathcal{T}_{12}$ could consist of statements of the form $A_1 \equiv A_2$ or $A_1 \sqsubseteq A_2$, where A_1 is a concept name in Σ_1 and A_2 is a concept name in Σ_2. When constructing such mappings, we typically do not want one ontology to interfere with the semantics of the other ontology via the mapping [96, 50, 48]. This condition can (and has been) formalized using inseparability. In fact, the non-interference requirement can be given by the condition that $\mathcal{O}_i$ and $\mathcal{O}_1 \cup \mathcal{O}_2 \cup \mathcal{T}_{12}$ are Σ_i inseparable, for $i = 1, 2$.

Knowledge base exchange. This application is a natural extension of *data exchange* [29], where the task is to transform a data instance $\mathcal{D}_1$ structured under a source schema Σ_1 into a data instance $\mathcal{D}_2$ structured under a target schema Σ_2 given a mapping $\mathcal{M}_{12}$ relating Σ_1 and Σ_2. In knowledge base exchange [2], we are interested in translating a KB $\mathcal{K}_1$ in a source signature Σ_1 to a KB $\mathcal{K}_2$ in a target signature Σ_2 according to a mapping given by a TBox $\mathcal{T}_{12}$ that consists of CIs and RIs in $\Sigma_1 \cup \Sigma_2$ defining concept and role names in Σ_2 in terms of concepts and roles in Σ_1. A good solution to this problem can be viewed as a KB $\mathcal{K}_2$ that it is inseparable from $\mathcal{K}_1 \cup \mathcal{T}_{12}$ with respect to a suitable Σ_2-inseparability relation.

Forgetting and uniform interpolation. When adapting an ontology to a new application, it is often useful to eliminate those symbols in its signature that are not relevant for the new application while retaining the semantics of the remaining ones. Another reason for eliminating symbols is predicate hiding, i.e., an ontology is to be published, but some part of it should be concealed from the public because it is confidential [26]. Moreover, one can view the elimination of symbols as an approach to ontology summary: the smaller and more focussed ontology summarizes what the original ontology says about the remaining signature items. The idea of eliminating symbols from a theory has been studied in AI under the name of forgetting a signature Σ [88]. In mathematical logic and modal logic, forgetting has been investigated under the dual notion of uniform interpolation [84, 27, 102, 38, 33, 98]. Under both names, the problem has been studied extensively in DL research [59, 105, 72, 104, 62, 79, 63]. Using inseparability, we can formulate the condition that the result $\mathcal{O}_{\mathsf{forget}}$ of eliminating Σ from $\mathcal{O}$ should not change the semantics of the remaining symbols by demanding that $\mathcal{O}$ and $\mathcal{O}_{\mathsf{forget}}$ are $\mathsf{sig}(\mathcal{O}) \setminus \Sigma$-inseparable for the signature $\mathsf{sig}(\mathcal{O})$ of $\mathcal{O}$.

4 Concept Inseparability

We consider inseparability relations that distinguish TBoxes if and only if they do not entail the same concept inclusions in a selected signature.[6] The resulting *concept inseparability* relations are appropriate for applications that focus on TBox reasoning. We start by defining concept inseparability and the related notions of concept entailment and concept conservative extensions. We give illustrating examples and discuss the relationship between the three notions and their connection to logical equivalence. We then take a detailed look at concept inseparability in $\mathcal{ALC}$ and in $\mathcal{EL}$. In both cases, we first establish a model-theoretic characterization and then show how this characterization can be used to decide concept entailment with the help of automata-theoretic techniques. We also briefly discuss extensions of $\mathcal{ALC}$ and the special case of $\mathcal{EL}$ with acyclic TBoxes.

Definition 2 (concept inseparability, entailment and conservative extension). Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be TBoxes formulated in some DL $\mathcal{L}$, and let Σ be a signature. Then

- the *Σ-concept difference between $\mathcal{T}_1$ and $\mathcal{T}_2$* is the set $\mathsf{cDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ of all Σ-concept inclusions (and role inclusions, if admitted by $\mathcal{L}$) α that are formulated in $\mathcal{L}$ and satisfy $\mathcal{T}_2 \models \alpha$ and $\mathcal{T}_1 \not\models \alpha$;
- $\mathcal{T}_1$ *Σ-concept entails* $\mathcal{T}_2$ if $\mathsf{cDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2) = \emptyset$;
- $\mathcal{T}_1$ and $\mathcal{T}_2$ are *Σ-concept inseparable* if $\mathcal{T}_1$ Σ-concept entails $\mathcal{T}_2$ and vice versa;
- $\mathcal{T}_2$ is a *concept conservative extension of* $\mathcal{T}_1$ if $\mathcal{T}_2 \supseteq \mathcal{T}_1$ and $\mathcal{T}_1$ and $\mathcal{T}_2$ are $\mathsf{sig}(\mathcal{T}_1)$-inseparable.

We illustrate this definition by a number of examples.

Example 3 (concept entailment vs. logical entailment). If $\Sigma \supseteq \mathsf{sig}(\mathcal{T}_1 \cup \mathcal{T}_2)$, then Σ-concept entailment is equivalent to logical entailment, that is, $\mathcal{T}_1$ Σ-concept entails $\mathcal{T}_2$ iff $\mathcal{T}_1 \models \mathcal{T}_2$. We recommend the reader to verify that this is a straightforward consequence of the definitions (it is crucial to observe that, because of our assumption on Σ, the concept inclusions in $\mathcal{T}_2$ qualify as potential members of $\mathsf{cDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$).

Example 4 (definitorial extension). An important way to extend an ontology is to introduce definitions of new concept names. Let $\mathcal{T}_1$ be a TBox, say formulated in $\mathcal{ALC}$, and let $\mathcal{T}_2 = \{A \equiv C\} \cup \mathcal{T}_1$, where A is a fresh concept name. Then

[6] For DLs that admit role inclusions, one additionally considers entailment of these.

$\mathcal{T}_2$ is called a *definitorial extension* of $\mathcal{T}_1$. Clearly, unless $\mathcal{T}_1$ is inconsistent, we have $\mathcal{T}_1 \not\models \mathcal{T}_2$. However, $\mathcal{T}_2$ is a concept-conservative extension of $\mathcal{T}_1$. For the proof, assume that $\mathcal{T}_1 \not\models \alpha$ and $\mathsf{sig}(\alpha) \subseteq \mathsf{sig}(\mathcal{T}_1)$. We show that $\mathcal{T}_2 \not\models \alpha$. There is a model $\mathcal{I}_1$ of $\mathcal{T}_1$ such that $\mathcal{I} \not\models \alpha$. Modify $\mathcal{I}$ by setting $A^{\mathcal{I}} = C^{\mathcal{I}}$. Then, since $A \notin \mathsf{sig}(\mathcal{T}_1)$, the new $\mathcal{I}$ is still a model of $\mathcal{T}_1$ and we still have $\mathcal{I} \not\models \alpha$. Moreover, $\mathcal{I}$ satisfies $A \equiv C$, and thus is a model of $\mathcal{T}_2$. Consequently, $\mathcal{T}_2 \not\models \alpha$.

The notion of concept inseparability depends on the DL in which the separating concept inclusions can be formulated. Note that, in Definition 12, we assume that this DL is the one in which the original TBoxes are formulated. Throughout this paper, we will thus make sure that the DL we work with is always clear from the context. We illustrate the difference that the choice of the 'separating DL' can make by two examples.

Example 5. Consider the $\mathcal{ALC}$ TBoxes

$$\mathcal{T}_1 = \{A \sqsubseteq \exists r.\top\} \quad \text{and} \quad \mathcal{T}_2 = \{A \sqsubseteq \exists r.B \sqcap \exists r.\neg B\}$$

and the signature $\Sigma = \{A, r\}$. If we view $\mathcal{T}_1$ and $\mathcal{T}_2$ as $\mathcal{ALCQ}$ TBoxes and consequently allow concept inclusions formulated in $\mathcal{ALCQ}$ to separate them, then $A \sqsubseteq (\geq 2r.\top) \in \mathsf{cDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$, and so $\mathcal{T}_1$ and $\mathcal{T}_2$ are Σ-concept separable. However, $\mathcal{T}_1$ and $\mathcal{T}_2$ are Σ-concept inseparable when we only allow separation in terms of $\mathcal{ALC}$-concept inclusions. Intuitively, this is the case because, in $\mathcal{ALC}$, one cannot count the number of r-successors of an individual. We will later introduce the model-theoretic machinery required to prove such statements in a formal way.

Example 6. Consider the $\mathcal{EL}$ TBoxes

$$\mathcal{T}_1 = \{\mathsf{Human} \sqsubseteq \exists \mathsf{eats}.\top,\ \mathsf{Plant} \sqsubseteq \exists \mathsf{grows\_in}.\mathsf{Area},\ \mathsf{Vegetarian} \sqsubseteq \mathsf{Healthy}\},$$
$$\mathcal{T}_2 = \mathcal{T}_1 \cup \{\mathsf{Human} \sqsubseteq \exists \mathsf{eats}.\mathsf{Food},\ \mathsf{Food} \sqcap \mathsf{Plant} \sqsubseteq \mathsf{Vegetarian}\}.$$

It can be verified that

$$\mathsf{Human} \sqcap \forall \mathsf{eats}.\mathsf{Plant} \sqsubseteq \exists \mathsf{eats}.\mathsf{Vegetarian}$$

is entailed by $\mathcal{T}_2$ but not by $\mathcal{T}_1$. If we view $\mathcal{T}_1$ and $\mathcal{T}_2$ as $\mathcal{ALC}$ TBoxes, then $\mathcal{T}_2$ is thus not a concept conservative extension of $\mathcal{T}_1$. However, we will show later that if we view $\mathcal{T}_1$ and $\mathcal{T}_2$ as $\mathcal{EL}$ TBoxes, then $\mathcal{T}_2$ is a concept conservative extension of $\mathcal{T}_1$ (i.e., $\mathcal{T}_1$ and $\mathcal{T}_2$ are $\mathsf{sig}(\mathcal{T}_1)$-inseparable in terms of $\mathcal{EL}$-concept inclusions).

As remarked in the introduction, conservative extensions are a special case of both inseparability and entailment. The former is by definition and the latter since $\mathcal{T}_2$ is a concept conservative extension of $\mathcal{T}_1 \subseteq \mathcal{T}_2$ iff $\mathcal{T}_1$ $\mathsf{sig}(\mathcal{T}_1)$-entails $\mathcal{T}_2$. Before turning our attention to specific DLs, we discuss a bit more the relationship between entailment and inseparability. On the one hand, inseparability is defined in terms of entailment and thus inseparability can be decided by two entailment checks. One might wonder about the converse direction, i.e., whether entailment can be reduced in some natural way to inseparability. This question is related to the following robustness condition.

Definition 3 (robustness under joins). A DL $\mathcal{L}$ is *robust under joins* for concept inseparability if, for all $\mathcal{L}$ TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ and signatures Σ with $\mathsf{sig}(\mathcal{T}_1) \cap \mathsf{sig}(\mathcal{T}_2) \subseteq \Sigma$, the following are equivalent:

(i) $\mathcal{T}_1$ Σ-concept entails $\mathcal{T}_2$ in $\mathcal{L}$;
(ii) $\mathcal{T}_1$ and $\mathcal{T}_1 \cup \mathcal{T}_2$ are Σ-concept inseparable in $\mathcal{L}$.

Observe that the implication $(ii) \Rightarrow (i)$ is trivial. The converse holds for many DLs such as $\mathcal{ALC}$, $\mathcal{ALCI}$ and $\mathcal{EL}$; see [55] for details. However, there are also standard DLs such as $\mathcal{ALCH}$ for which robustness under joins fails.

Theorem 1. *If a DL $\mathcal{L}$ is robust under joins for concept inseparability, then concept entailment in $\mathcal{L}$ can be polynomially reduced to concept inseparability in $\mathcal{L}$.*

Proof. Assume that we want to decide whether $\mathcal{T}_1$ Σ-concept entails $\mathcal{T}_2$. By replacing every non-Σ-symbol X shared by $\mathcal{T}_1$ and $\mathcal{T}_2$ with a fresh symbol X_1 in $\mathcal{T}_1$ and a distinct fresh symbol X_2 in $\mathcal{T}_2$, we can achieve that $\Sigma \supseteq \mathsf{sig}(\mathcal{T}_1) \cap \mathsf{sig}(\mathcal{T}_2)$ without changing (non-)Σ-concept entailment of $\mathcal{T}_2$ by $\mathcal{T}_1$. We then have, by robustness under joins, that $\mathcal{T}_1$ Σ-concept entails $\mathcal{T}_2$ iff $\mathcal{T}_1$ and $\mathcal{T}_1 \cup \mathcal{T}_2$ are Σ-concept inseparable. □

For DLs $\mathcal{L}$ that are not robust under joins for concept inseparability (such as $\mathcal{ALCH}$) it has not yet been investigated whether there exist natural polynomial reductions of concept entailment to concept inseparability.

4.1 Concept inseparability for $\mathcal{ALC}$

We first give a model-theoretic characterization of concept entailment in $\mathcal{ALC}$ in terms of bisimulations and then show how this characterization can be used to obtain an algorithm for deciding concept entailment based on automata-theoretic techniques. We also discuss the complexity, which is 2ExpTime-complete, and the size of minimal counterexamples that witness inseparability.

Bisimulations are a central tool for studying the expressive power of $\mathcal{ALC}$ and of modal logics; see for example [41, 68]. By a *pointed interpretation* we mean a pair $(\mathcal{I}, d)$, where $\mathcal{I}$ is an interpretation and $d \in \Delta^{\mathcal{I}}$.

Definition 4 (Σ-bisimulation). Let Σ be a finite signature and $(\mathcal{I}_1, d_1)$ and $(\mathcal{I}_2, d_2)$ pointed interpretations. A relation $S \subseteq \Delta^{\mathcal{I}_1} \times \Delta^{\mathcal{I}_2}$ is a *Σ-bisimulation* between $(\mathcal{I}_1, d_1)$ and $(\mathcal{I}_2, d_2)$ if $(d_1, d_2) \in S$ and, for all $(d, d') \in S$, the following conditions are satisfied:

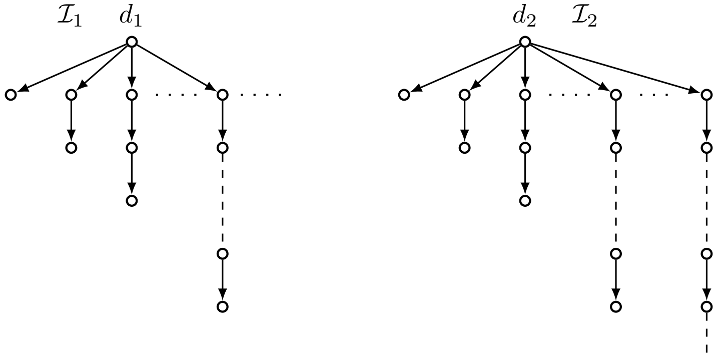
(base) $d \in A^{\mathcal{I}_1}$ iff $d' \in A^{\mathcal{I}_2}$, for all $A \in \Sigma \cap \mathsf{N_C}$;
(zig) if $(d, e) \in r^{\mathcal{I}_1}$, then there exists $e' \in \Delta^{\mathcal{I}_2}$ such that $(d', e') \in r^{\mathcal{I}_2}$ and $(e, e') \in S$, for all $r \in \Sigma \cap \mathsf{N_R}$;
(zag) if $(d', e') \in r^{\mathcal{I}_2}$, then there exists $e \in \Delta^{\mathcal{I}_1}$ such that $(d, e) \in r^{\mathcal{I}_1}$ and $(e, e') \in S$, for all $r \in \Sigma \cap \mathsf{N_R}$.

We say that $(\mathcal{I}_1, d_1)$ and $(\mathcal{I}_2, d_2)$ are *Σ-bisimilar* and write $(\mathcal{I}_1, d_1) \sim_{\Sigma}^{\mathsf{bisim}} (\mathcal{I}_2, d_2)$ if there exists a Σ-bisimulation between them.

We now recall the main connection between bisimulations and $\mathcal{ALC}$. Say that $(\mathcal{I}_1, d_1)$ and $(\mathcal{I}_2, d_2)$ are *$\mathcal{ALC}_\Sigma$-equivalent*, in symbols $(\mathcal{I}_1, d_1) \equiv_{\Sigma}^{\mathcal{ALC}} (\mathcal{I}_2, d_2)$, in case $d_1 \in C^{\mathcal{I}_1}$ iff $d_2 \in C^{\mathcal{I}_2}$ for all Σ-concepts C in $\mathcal{ALC}$. An interpretation $\mathcal{I}$ is of *finite outdegree* if the set $\{d' \mid (d, d') \in \bigcup_{r \in \mathsf{N_R}} r^{\mathcal{I}}\}$ is finite, for any $d \in \Delta^{\mathcal{I}}$.

Theorem 2. *Let $(\mathcal{I}_1, d_1)$ and $(\mathcal{I}_2, d_2)$ be pointed interpretations and Σ a signature. Then $(\mathcal{I}_1, d_1) \sim_{\Sigma}^{\mathsf{bisim}} (\mathcal{I}_2, d_2)$ implies $(\mathcal{I}_1, d_1) \equiv_{\Sigma}^{\mathcal{ALC}} (\mathcal{I}_2, d_2)$. The converse holds if $\mathcal{I}_1$ and $\mathcal{I}_2$ are of finite outdegree.*

Example 7. The following classical example shows that without the condition of finite outdegree, the converse direction does not hold.



Here, $(\mathcal{I}_1, d_1)$ is a pointed interpretation with an r-chain of length n starting from d_1, for each $n \geq 1$. $(\mathcal{I}_2, d_2)$ coincides with $(\mathcal{I}_1, d_1)$ except that it also contains an infinite r-chain starting from d_2. Let $\Sigma = \{r\}$. It can be proved that $(\mathcal{I}_1, d_1) \equiv_{\Sigma}^{\mathcal{ALC}} (\mathcal{I}_2, d_2)$. However, $(\mathcal{I}_1, d_1) \not\sim_{\Sigma}^{\mathsf{bisim}} (\mathcal{I}_2, d_2)$ due to the infinite chain in $(\mathcal{I}_2, d_2)$.

As a first application of Theorem 2, we note that $\mathcal{ALC}$ cannot distinguish between an interpretation and its *unraveling* into a tree. An interpretation $\mathcal{I}$ is called a *tree interpretation* if $r^{\mathcal{I}} \cap s^{\mathcal{I}} = \emptyset$ for any $r \neq s$ and the directed graph $(\Delta^{\mathcal{I}}, \bigcup_{r \in \mathsf{N_R}} r^{\mathcal{I}})$ is a (possibly infinite) tree. The root of $\mathcal{I}$ is denoted by $\rho^{\mathcal{I}}$. By the unraveling technique [41], one can show that every pointed interpretation $(\mathcal{I}, d)$ is Σ-bisimilar to a pointed tree interpretation $(\mathcal{I}^*, \rho^{\mathcal{I}^*})$, for any signature Σ. Indeed, suppose $(\mathcal{I}, d)$ is given. The domain $\Delta^{\mathcal{I}^*}$ of $\mathcal{I}^*$ is the set of words $w = d_0 r_0 d_1 \cdots r_n d_n$ such that $d_0 = d$ and $(d_i, d_{i+1}) \in r_i^{\mathcal{I}}$ for all $i < n$ and roles names r_i. We set $\mathsf{tail}(d_0 r_0 d_1 \cdots r_n d_n) = d_n$ and define the interpretation $A^{\mathcal{I}^*}$ and $r^{\mathcal{I}^*}$ of concept names A and role names r by setting:

- $w \in A^{\mathcal{I}^*}$ if $\mathsf{tail}(w) \in A^{\mathcal{I}^*}$;
- $(w, w') \in r^{\mathcal{I}^*}$ if $w' = wrd'$.

The following lemma can be proved by a straightforward induction.

Lemma 1. *The relation $S = \{(w, \mathsf{tail}(w)) \mid w \in \Delta^{\mathcal{I}^*}\}$ is a Σ-bisimulation between $(\mathcal{I}^*, \rho^{\mathcal{I}^*})$ and $(\mathcal{I}, d)$, for any signature Σ.*

We now characterize concept entailment (and thus also concept inseparability and concept conservative extensions) in $\mathcal{ALC}$ using bisimulations, following [72].

Theorem 3. *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be $\mathcal{ALC}$ TBoxes and Σ a signature. Then $\mathcal{T}_1$ Σ-concept entails $\mathcal{T}_2$ iff, for any model $\mathcal{I}_1$ of $\mathcal{T}_1$ and any $d_1 \in \Delta^{\mathcal{I}_1}$, there exist a model $\mathcal{I}_2$ of $\mathcal{T}_2$ and $d_2 \in \Delta^{\mathcal{I}_2}$ such that $(\mathcal{I}_1, d_1) \sim^{\mathsf{bisim}}_{\Sigma} (\mathcal{I}_2, d_2)$.*

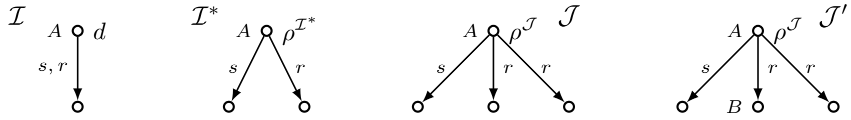
For $\mathcal{I}_1$ of finite outdegree, one can prove this result directly by employing compactness arguments and Theorem 2. For the general case, we refer to [72]. We illustrate Theorem 3 by sketching a proof of the statement from Example 5 (in a slightly more general form).

Example 8. Consider the $\mathcal{ALC}$ TBoxes

$$\mathcal{T}_1 = \{A \sqsubseteq \exists r.\top\} \cup \mathcal{T} \quad \text{and} \quad \mathcal{T}_2 = \{A \sqsubseteq \exists r.B \sqcap \exists r.\neg B\} \cup \mathcal{T},$$

where $\mathcal{T}$ is an $\mathcal{ALC}$ TBox and $B \notin \Sigma = \{A, r\} \cup \mathsf{sig}(\mathcal{T})$. We use Theorem 3 and Lemma 1 to show that $\mathcal{T}_1$ Σ-concept entails $\mathcal{T}_2$. Suppose $\mathcal{I}$ is a model of $\mathcal{T}_1$ and $d \in \Delta^{\mathcal{I}}$. Using tree unraveling, we construct a tree model $\mathcal{I}^*$ of $\mathcal{T}_1$ with $(\mathcal{I}, d) \sim^{\mathsf{bisim}}_{\Sigma} (\mathcal{I}^*, \rho^{\mathcal{I}^*})$. As bisimilations and $\mathcal{ALC}$ TBoxes are oblivious to duplication of successors, we find a tree model $\mathcal{J}$ of $\mathcal{T}_1$ such that $e \in A^{\mathcal{J}}$ implies $\#\{d \mid (e, d) \in r^{\mathcal{J}}\} \geq 2$ for all $e \in \Delta^{\mathcal{J}}$ and $(\mathcal{I}^*, \rho^{\mathcal{I}^*}) \sim^{\mathsf{bisim}}_{\Sigma} (\mathcal{J}, \rho^{\mathcal{J}})$. By reinterpreting $B \notin \Sigma$, we can find $\mathcal{J}'$ that coincides with $\mathcal{J}$ except that now we ensure that $e \in A^{\mathcal{J}}$ implies $e \in (\exists r.B \sqcap \exists r.\neg B)^{\mathcal{J}'}$ for all $e \in \Delta^{\mathcal{J}}$. But then $\mathcal{J}'$ is a model of $\mathcal{T}_2$ and $(\mathcal{I}, d) \sim^{\mathsf{bisim}}_{\Sigma} (\mathcal{J}', \rho^{\mathcal{J}})$, as required.

Below, we illustrate possible interpretations $\mathcal{I}^*$, $\mathcal{J}$ and $\mathcal{J}'$ satisfying the above conditions, for a given interpretation $\mathcal{I}$.



Theorem 3 is a useful starting point for constructing decision procedures for concept entailment in $\mathcal{ALC}$ and related problems. This can be done from first principles as in [36, 70]. Here we present an approach that uses tree automata. We use amorphous alternating parity tree automata [106], which actually run on unrestricted interpretations rather than on trees. They still belong to the family of *tree* automata as they are in the tradition of more classical forms of such automata and cannot distinguish between an interpretation and its unraveling into a tree (which indicates a connection to bisimulations).

Definition 5 (APTA). An (*amorphous*) *alternating parity tree automaton* (or APTA for short) is a tuple $\mathcal{A} = (Q, \Sigma_N, \Sigma_E, q_0, \delta, \Omega)$, where Q is a finite set of *states*, $\Sigma_N \subseteq \mathsf{N_C}$ is the finite *node alphabet*, $\Sigma_E \subseteq \mathsf{N_R}$ is the finite *edge alphabet*, $q_0 \in Q$ is the *initial state*, $\delta : Q \to \mathsf{mov}(\mathcal{A})$ is the transition function with $\mathsf{mov}(\mathcal{A}) = \{\mathsf{true}, \mathsf{false}, A, \neg A, q, q \wedge q', q \vee q', \langle r \rangle q, [r]q \mid A \in \Sigma_N, q, q' \in Q, r \in \Sigma_E\}$ the set of *moves* of the automaton, and $\Omega : Q \to \mathbb{N}$ is the *priority function*.

Intuitively, the move q means that the automaton sends a copy of itself in state q to the element of the interpretation that it is currently processing, $\langle r\rangle q$ means that a copy in state q is sent to an r-successor of the current element, and $[r]q$ means that a copy in state q is sent to every r-successor.

It will be convenient to use unrestricted modal logic formulas in negation normal form when specifying the transition function of APTAs. The more restricted form required by Definition 5 can then be attained by introducing intermediate states. We next introduce the semantics of APTAs.

In what follows, a *Σ-labelled tree* is a pair (T, ℓ) with T a tree and $\ell : T \to \Sigma$ a node labelling function. A *path* π in a tree T is a subset of T such that $\varepsilon \in \pi$ and for each $x \in \pi$ that is not a leaf in T, π contains one child of x.

Definition 6 (run). Let $(\mathcal{I}, d_0)$ be a pointed $\Sigma_N \cup \Sigma_E$-interpretation and let $\mathfrak{A} = (Q, \Sigma_N, \Sigma_E, q_0, \delta, \Omega)$ be an APTA. A *run* of $\mathfrak{A}$ on $(\mathcal{I}, d_0)$ is a $Q \times \Delta^{\mathcal{I}}$-labelled tree (T, ℓ) such that $\ell(\varepsilon) = (q_0, d_0)$ and for every $x \in T$ with $\ell(x) = (q, d)$:

- $\delta(q) \neq \mathsf{false}$;
- if $\delta(q) = A$ ($\delta(q) = \neg A$), then $d \in A^{\mathcal{I}}$ ($d \notin A^{\mathcal{I}}$);
- if $\delta(q) = q' \wedge q''$, then there are children y, y' of x with $\ell(y) = (q', d)$ and $\ell(y') = (q'', d)$;
- if $\delta(q) = q' \vee q''$, then there is a child y of x such that $\ell(y) = (q', d)$ or $\ell(y') = (q'', d)$;
- if $\delta(q) = \langle r\rangle q'$, then there is a $(d, d') \in r^{\mathcal{I}}$ and a child y of x such that $\ell(y) = (q', d')$;
- if $\delta(q) = [r]q'$ and $(d, d') \in r^{\mathcal{I}}$, then there is a child y of x with $\ell(y) = (q', d')$.

A run (T, ℓ) is *accepting* if, for every path π of T, the maximal $i \in \mathbb{N}$ with $\{x \in \pi \mid \ell(x) = (q, d) \text{ with } \Omega(q) = i\}$ infinite is even. We use $L(\mathfrak{A})$ to denote the language accepted by $\mathfrak{A}$, i.e., the set of pointed $\Sigma_N \cup \Sigma_E$-interpretations $(\mathcal{I}, d)$ such that there is an accepting run of $\mathfrak{A}$ on $(\mathcal{I}, d)$.

APTAs can easily be complemented in polynomial time in the same way as other alternating tree automata, and for all APTAs $\mathfrak{A}_1$ and $\mathfrak{A}_2$, one can construct in polynomial time an APTA that accepts $L(\mathfrak{A}_1) \cap L(\mathfrak{A}_2)$. The emptiness problem for APTAs is ExpTime-complete [106].

We now describe how APTAs can be used to decide concept entailment in $\mathcal{ALC}$. Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be $\mathcal{ALC}$ TBoxes and Σ a signature. By Theorem 3, $\mathcal{T}_1$ does not Σ-concept entail $\mathcal{T}_2$ iff there is a model $\mathcal{I}_1$ of $\mathcal{T}_1$ and a $d_1 \in \Delta^{\mathcal{I}_1}$ such that $(\mathcal{I}_1, d_1) \not\sim_{\Sigma}^{\mathsf{bisim}} (\mathcal{I}_2, d_2)$ for all models $\mathcal{I}_2$ of $\mathcal{T}_2$ and $d_2 \in \Delta^{\mathcal{I}_2}$. We first observe that this still holds when we restrict ourselves to *rooted interpretations*, that is, to pointed interpretations $(\mathcal{I}_i, d_i)$ such that every $e \in \Delta^{\mathcal{I}_i}$ is reachable from d_i by some sequence of role names. In fact, whenever $(\mathcal{I}_1, d_1) \not\sim_{\Sigma}^{\mathsf{bisim}} (\mathcal{I}_2, d_2)$, then also $(\mathcal{I}_1^r, d_1) \not\sim_{\Sigma}^{\mathsf{bisim}} (\mathcal{I}_2^r, d_2)$ where $\mathcal{I}_i^r$ is the restriction of $\mathcal{I}_i$ to the elements reachable from d_i. Moreover, if $\mathcal{I}_1$ and $\mathcal{I}_2$ are models of $\mathcal{T}_1$ and $\mathcal{T}_2$, respectively, then the same is true for $\mathcal{I}_1^r$ and $\mathcal{I}_2^r$. Rootedness is important because APTAs can obviously not speak about unreachable parts of a pointed interpretation. We now construct two APTAs $\mathfrak{A}_1$ and $\mathfrak{A}_2$ such that for all rooted $\mathsf{sig}(\mathcal{T}_1)$-interpretations $(\mathcal{I}, d)$,

1. $(\mathcal{I}, d) \in L(\mathfrak{A}_1)$ iff $\mathcal{I} \models \mathcal{T}_1$;
2. $(\mathcal{I}, d) \in L(\mathfrak{A}_2)$ iff there exist a model $\mathcal{J}$ of $\mathcal{T}_2$ and an $e \in \Delta^{\mathcal{J}}$ such that $(\mathcal{I}, d) \sim_{\Sigma}^{\mathsf{bisim}} (\mathcal{J}, e)$.

Defining $\mathfrak{A}$ as $\mathfrak{A}_1 \cap \overline{\mathfrak{A}_2}$, we then have $L(\mathfrak{A}) = \emptyset$ iff $\mathcal{T}_1$ Σ-concept entails $\mathcal{T}_2$. It is easy to construct the automaton $\mathfrak{A}_1$. We only illustrate the idea by an example. Assume that $\mathcal{T}_1 = \{A \sqsubseteq \neg\forall r.B\}$. We first rewrite $\mathcal{T}_1$ into the equivalent TBox $\{\top \sqsubseteq \neg A \sqcup \exists r.\neg B\}$ and then use an APTA with only state q_0 and

$$\delta(q_0) = \bigwedge_{s \in \mathsf{N_R}} [s]q_0 \wedge (\neg A \vee \langle r\rangle \neg B).$$

The acceptance condition is trivial, that is, $\Omega(q_0) = 0$. The construction of $\mathfrak{A}_2$ is more interesting. We require the notion of a type, which occurs in many constructions for $\mathcal{ALC}$. Let $\mathsf{cl}(\mathcal{T}_2)$ denote the set of concepts used in $\mathcal{T}_2$, closed under subconcepts and single negation. A type t is a set $t \subseteq \mathsf{cl}(\mathcal{T}_2)$ such that, for some model $\mathcal{I}$ of $\mathcal{T}_2$ and some $d \in \Delta^{\mathcal{I}}$, we have $t = \{C \in \mathsf{cl}(\mathcal{T}_2) \mid d \in C^{\mathcal{I}}\}$. Let $\mathsf{TP}(\mathcal{T}_2)$ denote the set of all types for $\mathcal{T}_2$. For $t, t' \in \mathsf{TP}(\mathcal{T}_2)$ and a role name r, we write $t \rightsquigarrow_r t'$ if (i) $\forall r.C \in t$ implies $C \in t'$ and (ii) $C \in t'$ implies $\exists r.C \in t$ whenever $\exists r.C \in \mathsf{cl}(\mathcal{T})$. Now we define $\mathfrak{A}_2$ to have state set $Q = \mathsf{TP}(\mathcal{T}_2) \uplus \{q_0\}$ and the following transitions:

$$\begin{aligned}
\delta(q_0) &= \bigvee \mathsf{TP}(\mathcal{T}_2), \\
\delta(t) &= \bigwedge_{A \in t \cap \mathsf{N_C} \cap \Sigma} A \wedge \bigwedge_{A \in (\mathsf{N_C} \cap \Sigma) \setminus t} \neg A \\
&\quad \wedge \bigwedge_{r \in \Sigma \cap \mathsf{N_R}} [r] \bigvee \{t' \in \mathsf{TP}(\mathcal{T}_2) \mid t \rightsquigarrow_r t'\} \\
&\quad \wedge \bigwedge_{\exists r.C \in t, r \in \Sigma} \langle r\rangle \bigvee \{t' \in \mathsf{TP}(\mathcal{T}_2) \mid t \rightsquigarrow_r t',\ C \in t'\}.
\end{aligned}$$

Here, the empty conjunction represents true and the empty disjunction represents false. The acceptance condition is again trivial, but note that this might change with complementation. The idea is that $\mathfrak{A}_2$ (partially) guesses a model $\mathcal{J}$ that is Σ-bisimilar to the input interpretation $\mathcal{I}$, represented as types. Note that $\mathfrak{A}_2$ verifies only the Σ-part of $\mathcal{J}$ on $\mathcal{I}$, and that it might label the same element with different types (which can then only differ in their non-Σ-parts). A detailed proof that the above automaton works as expected is provided in [72]. In summary, we obtain the upper bound in the following theorem.

Theorem 4. *In $\mathcal{ALC}$, concept entailment, concept inseparability, and concept conservative extensions are 2*Exp*T*ime*-complete.*

The sketched APTA-based decision procedure actually yields an upper bound that is slightly stronger than what is stated in Theorem 4: the algorithm for concept entailment (and concept conservative extensions) actually runs in time $2^{p(|\mathcal{T}_1| \cdot 2^{|\mathcal{T}_2|})}$ for some polynomial $p()$ and is thus only single exponential in $|\mathcal{T}_1|$.

For simplicity, in the remainder of the paper we will typically not explicitly report on such fine-grained upper bounds that distinguish between different inputs.

The lower bound stated in Theorem 4 is proved (for concept conservative extensions) in [36] using a rather intricate reduction of the word problem of exponentially space bounded alternating Turing machines (ATMs). An interesting issue that is closely related to computational hardness is to analyze the size of the smallest concept inclusions that witness non-Σ-concept entailment of a TBox $\mathcal{T}_2$ by a TBox $\mathcal{T}_1$, that is, of the members of $\mathsf{cDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$. It is shown in [36] for the case of concept conservative extensions in $\mathcal{ALC}$ (and thus also for concept entailment) that smallest witness inclusions can be triple exponential in size, but not larger. An example that shows why witness inclusions can get large is given in Section 4.3.

4.2 Concept inseparability for extensions of $\mathcal{ALC}$

We briefly discuss results on concept inseparability for extensions of $\mathcal{ALC}$ and give pointers to the literature.

In principle, the machinery and results that we have presented for $\mathcal{ALC}$ can be adapted to many extensions of $\mathcal{ALC}$, for example, with number restrictions, inverse roles, and role inclusions. To achieve this, the notion of bisimulation has to be adapted to match the expressive power of the considered DL and the automata construction has to be modified. In particular, amorphous automata as used above are tightly linked to the expressive power of $\mathcal{ALC}$ and have to be replaced by traditional alternating tree automata (running on trees with fixed outdegree) which requires a slightly more technical automaton construction.

As an illustration, we only give some brief examples. To obtain an analogue of Theorem 2 for $\mathcal{ALCI}$, one needs to extend bisimulations that additionally respect successors reachable by an inverse role; to obtain such a result for $\mathcal{ALCQ}$, we need bisimilations that respect the number of successors [41, 68, 57]. Corresponding versions of Theorem 3 can then be proved using techniques from [68, 57].

Example 9. Consider the $\mathcal{ALCQ}$ TBoxes

$$\mathcal{T}_1 = \{A \sqsubseteq {\geq}\, 2\, r.\top\} \cup \mathcal{T} \quad \text{and} \quad \mathcal{T}_2 = \{A \sqsubseteq \exists r.B \sqcap \exists r.\neg B\} \cup \mathcal{T},$$

where $\mathcal{T}$ is an $\mathcal{ALCQ}$ TBox that does not use the concept name B. Suppose $\Sigma = \{A, r\} \cup \mathsf{sig}(\mathcal{T})$. Then $\mathcal{T}_1$ and $\mathcal{T}_2$ are Σ-concept inseparable in $\mathcal{ALCQ}$. Formally, this can be shown using the characterizations from [57].

The above approach has not been fully developed in the literature. However, using more elementary methods, the following complexity result has been established in [70].

Theorem 5. *In $\mathcal{ALCQI}$, concept entailment, concept inseparability, and concept conservative extensions are* 2ExpTime*-complete.*

It is also shown in [70] that, in $\mathcal{ALCQI}$, smallest counterexamples are still triple exponential, and that further adding nominals to $\mathcal{ALCQI}$ results in undecidability.

Theorem 6. *In $\mathcal{ALCQIO}$, concept entailment, concept inseparability, and concept conservative extensions are undecidable.*

For a number of prominent extensions of $\mathcal{ALC}$, concept inseparability has not yet been investigated in much detail. This particularly concerns extensions with transitive roles [45]. We note that it is not straightforward to lift the above techniques to DLs with transitive roles; see [37] where conservative extensions in modal logics with transitive frames are studied and [33] in which modal logics with bisimulation quantifiers (which are implicit in Theorem 3) are studied, including cases with transitive frame classes. As illustrated in Section 8, extensions of $\mathcal{ALC}$ with the universal role are also an interesting subject to study.

4.3 Concept inseparability for $\mathcal{EL}$

We again start with model-theoretic characterizations and then proceed to decision procedures, complexity, and the length of counterexamples. In contrast to $\mathcal{ALC}$, we use simulations, which intuitively are 'half a bisimulation', much like $\mathcal{EL}$ is 'half of $\mathcal{ALC}$'. The precise definition is as follows.

Definition 7 (Σ-simulation). Let Σ be a finite signature and $(\mathcal{I}_1, d_1)$, $(\mathcal{I}_2, d_2)$ pointed interpretations. A relation $S \subseteq \Delta^{\mathcal{I}_1} \times \Delta^{\mathcal{I}_2}$ is a *Σ-simulation* from $(\mathcal{I}_1, d_1)$ to $(\mathcal{I}_2, d_2)$ if $(d_1, d_2) \in S$ and, for all $(d, d') \in S$, the following conditions are satisfied:

(base$^\ell$) if $d \in A^{\mathcal{I}_1}$, then $d' \in A^{\mathcal{I}_2}$, for all $A \in \Sigma \cap \mathsf{N_C}$;
(zig) if $(d, e) \in r^{\mathcal{I}_1}$, then there exists $e' \in \Delta^{\mathcal{I}_2}$ such that $(d', e') \in r^{\mathcal{I}_2}$ and $(e, e') \in S$, for all $r \in \Sigma \cap \mathsf{N_R}$.

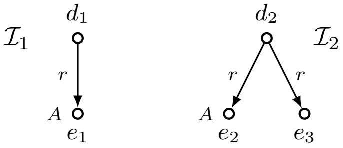
We say that $(\mathcal{I}_2, d_2)$ *Σ-simulates* $(\mathcal{I}_1, d_1)$ and write $(\mathcal{I}_1, d_1) \leq^{\mathsf{sim}}_{\Sigma} (\mathcal{I}_2, d_2)$ if there exist a Σ-simulation from $(\mathcal{I}_1, d_1)$ to $(\mathcal{I}_2, d_2)$. We say that $(\mathcal{I}_1, d_1)$ and $(\mathcal{I}_2, d_2)$ are *Σ-equisimilar*, in symbols $(\mathcal{I}_1, d_1) \sim^{\mathsf{esim}}_{\Sigma} (\mathcal{I}_2, d_2)$, if both $(\mathcal{I}_1, d_1) \leq^{\mathsf{sim}}_{\Sigma} (\mathcal{I}_2, d_2)$ and $(\mathcal{I}_2, d_2) \leq^{\mathsf{sim}}_{\Sigma} (\mathcal{I}_1, d_1)$.

A pointed interpretation $(\mathcal{I}_1, d_1)$ is *$\mathcal{EL}_\Sigma$-contained in* $(\mathcal{I}_2, d_2)$, in symbols $(\mathcal{I}_1, d_1) \leq^{\mathcal{EL}}_{\Sigma} (\mathcal{I}_2, d_2)$, if $d_1 \in C^{\mathcal{I}_1}$ implies $d_2 \in C^{\mathcal{I}_2}$, for all $\mathcal{EL}_\Sigma$-concepts C. We call pointed interpretations $(\mathcal{I}_1, d_1)$ and $(\mathcal{I}_2, d_2)$ *$\mathcal{EL}_\Sigma$-equivalent*, in symbols $(\mathcal{I}_1, d_1) \equiv^{\mathcal{EL}}_{\Sigma} (\mathcal{I}_2, d_2)$, in case $(\mathcal{I}_1, d_1) \leq^{\mathcal{EL}}_{\Sigma} (\mathcal{I}_2, d_2)$ and $(\mathcal{I}_2, d_2) \leq^{\mathcal{EL}}_{\Sigma} (\mathcal{I}_1, d_1)$. The following was shown in [71, 69].

Theorem 7. *Let $(\mathcal{I}_1, d_1)$ and $(\mathcal{I}_2, d_2)$ be pointed interpretations and Σ a signature. Then $(\mathcal{I}_1, d_1) \leq^{\mathsf{sim}}_{\Sigma} (\mathcal{I}_2, d_2)$ implies $(\mathcal{I}_1, d_1) \leq^{\mathcal{EL}}_{\Sigma} (\mathcal{I}_2, d_2)$. The converse holds if $\mathcal{I}_1$ and $\mathcal{I}_2$ are of finite outdegree.*

The interpretations given in Example 7 can be used to show that the converse direction in Theorem 7 does not hold in general (since $(\mathcal{I}_2, d_2) \not\leq^{\mathsf{sim}}_{\Sigma} (\mathcal{I}_1, d_1)$). It is instructive to see pointed interpretations that are equisimilar but not bisimilar.

Example 10. Consider the interpretations $\mathcal{I}_1 = (\{d_1, e_1\}, A^{\mathcal{I}_1} = \{e_1\}, r^{\mathcal{I}_1} = \{(d_1, e_1)\})$ and $\mathcal{I}_2 = (\{d_2, e_2, e_3\}, A^{\mathcal{I}_2} = \{e_2\}, r^{\mathcal{I}_2} = \{(d_2, e_2), (d_2, e_3)\})$ and let $\Sigma = \{r, A\}$. Then $(\mathcal{I}_1, d_1)$ and $(\mathcal{I}_2, d_2)$ are Σ-equisimilar but not Σ-bisimilar.



Similar to Theorem 3, Σ-equisimilarity can be used to give a model-theoretic characterization of concept entailment (and thus also concept inseparability and concept conservative extensions) in $\mathcal{EL}$ [69].

Theorem 8. *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be $\mathcal{EL}$ TBoxes and Σ a signature. Then $\mathcal{T}_1$ Σ-concept entails $\mathcal{T}_2$ iff, for any model $\mathcal{I}_1$ of $\mathcal{T}_1$ and any $d_1 \in \Delta^{\mathcal{I}_1}$, there exist a model $\mathcal{I}_2$ of $\mathcal{T}_2$ and $d_2 \in \Delta^{\mathcal{I}_2}$ such that $(\mathcal{I}_1, d_1) \sim_{\Sigma}^{\mathsf{esim}} (\mathcal{I}_2, d_2)$.*

We illustrate Theorem 8 by proving that the TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ from Example 6 are Σ-concept inseparable in $\mathcal{EL}$.

Example 11. Recall that $\Sigma = \mathsf{sig}(\mathcal{T}_1)$ and

$$\mathcal{T}_1 = \{\mathsf{Human} \sqsubseteq \exists \mathsf{eats}.\top,\ \mathsf{Plant} \sqsubseteq \exists \mathsf{grows\_in}.\mathsf{Area},\ \mathsf{Vegetarian} \sqsubseteq \mathsf{Healthy}\},$$
$$\mathcal{T}_2 = \mathcal{T}_1 \cup \{\mathsf{Human} \sqsubseteq \exists \mathsf{eats}.\mathsf{Food},\ \mathsf{Food} \sqcap \mathsf{Plant} \sqsubseteq \mathsf{Vegetarian}\}.$$

Let $\mathcal{I}$ be a model of $\mathcal{T}_1$ and $d \in \Delta^{\mathcal{I}}$. We may assume that $\mathsf{Food}^{\mathcal{I}} = \emptyset$. Define $\mathcal{I}'$ by adding, for every $e \in \mathsf{Human}^{\mathcal{I}}$, a fresh individual $\mathsf{new}(e)$ to $\Delta^{\mathcal{I}}$ with $(e, \mathsf{new}(e)) \in \mathsf{eats}^{\mathcal{I}'}$ and $\mathsf{new}(e) \in \mathsf{Food}^{\mathcal{I}'}$. Clearly, $\mathcal{I}'$ is a model of $\mathcal{T}_2$. We show that $(\mathcal{I}, d)$ and $(\mathcal{I}', d)$ are Σ-equisimilar. The identity $\{(e, e) \mid e \in \Delta^{\mathcal{I}}\}$ is obviously a Σ-simulation from $(\mathcal{I}, d)$ to $(\mathcal{I}', d)$. Conversely, pick for each $e \in \mathsf{Human}^{\mathcal{I}'}$ an $\mathsf{old}(e) \in \Delta^{\mathcal{I}}$ with $(e, \mathsf{old}(e)) \in \mathsf{eats}^{\mathcal{I}}$, which must exist by the first CI of $\mathcal{T}_1$. It can be verified that

$$S = \{(e, e) \mid e \in \Delta^{\mathcal{I}}\} \cup \{(\mathsf{new}(e), \mathsf{old}(e)) \mid e \in \Delta^{\mathcal{I}}\}$$

is a Σ-simulation from $(\mathcal{I}', d)$ to $(\mathcal{I}, d)$. Note that $(\mathcal{I}, d)$ and $(\mathcal{I}', d)$ are not guaranteed to be Σ-bisimilar.

As in the $\mathcal{ALC}$ case, Theorem 8 gives rise to a decision procedure for concept entailment based on tree automata. However, we can now get the complexity down to ExpTime.[7] To achieve this, we define the automaton $\mathfrak{A}_2$ in a more careful way than for $\mathcal{ALC}$, while we do not touch the construction of $\mathfrak{A}_1$. Let $\mathsf{sub}(\mathcal{T}_2)$ denote the set of concepts that occur in $\mathcal{T}_2$, closed under subconcepts. For any $C \in \mathsf{sub}(\mathcal{T}_2)$, we use $\mathsf{con}_{\mathcal{T}}(C)$ to denote the set of concepts $D \in \mathsf{sub}(\mathcal{T}_2)$ such that $\mathcal{T} \models C \sqsubseteq D$. We define the APTA based on the set of states

$$Q = \{q_0\} \uplus \{q_C, \overline{q}_C \mid C \in \mathsf{sub}(\mathcal{T}_2)\},$$

[7] An alternative elementary proof is given in [71].

where q_0 is the starting state. The transitions are as follows:

$$\begin{aligned}
\delta(q_0) &= \bigwedge_{C \in \mathsf{sub}(\mathcal{T}_2)} (q_C \vee \overline{q}_C) \wedge \bigwedge_{r \in \Sigma} [r] q_0, \\
\delta(q_A) &= A \wedge \bigwedge_{C \in \mathsf{con}_{\mathcal{T}}(A)} q_C \text{ and } \delta(\overline{q}_A) = \neg A \text{ for all } A \in \mathsf{sub}(\mathcal{T}_2) \cap \mathsf{N_C} \cap \Sigma, \\
\delta(q_{C \sqcap D}) &= q_C \wedge q_D \wedge \bigwedge_{E \in \mathsf{con}_{\mathcal{T}}(C \sqcap D)} q_E \text{ and} \\
\delta(\overline{q}_{C \sqcap D}) &= \overline{q}_C \vee \overline{q}_D \text{ for all } C \sqcap D \in \mathsf{sub}(\mathcal{T}_2), \\
\delta(q_{\exists r.C}) &= \langle r \rangle q_C \wedge \bigwedge_{D \in \mathsf{con}_{\mathcal{T}}(\exists r.C)} q_D \text{ and} \\
\delta(\overline{q}_{\exists r.C}) &= [r] \overline{q}_C \text{ for all } \exists r.C \in \mathsf{sub}(\mathcal{T}_2) \text{ with } r \in \Sigma, \\
\delta(q_\top) &= \bigwedge_{C \in \mathsf{con}_{\mathcal{T}}(\top)} q_C \text{ and } \delta(\overline{q}_\top) = \mathsf{false}.
\end{aligned}$$

Observe that, in each case, the transition for $\overline{q}_C$ is the dual of the transition for q_C, except that the latter has an additional conjunction pertaining to $\mathsf{con}_{\mathcal{T}}$. As before, we set $\Omega(q) = 0$ for all $q \in Q$. An essential difference between the above APTA $\mathfrak{A}_2$ and the one that we had constructed for $\mathcal{ALC}$ is that the latter had to look at sets of subconcepts (in the form of a type) while the automaton above always considers only a single subconcept at the time. A proof that the above automaton works as expected can be extracted from [69].

Theorem 9. *In $\mathcal{EL}$, concept entailment, concept inseparability, and concept conservative extensions are* ExpTime*-complete.*

The lower bound in Theorem 9 is proved (for concept conservative extension) in [71] using a reduction of the word problem of polynomially space bounded ATMs. It can be extracted from the proofs in [71] that smallest concept inclusions that witness failure of concept entailment (or concept conservative extensions) are at most double exponentially large, measured in the size of the input TBoxes.[8] The following example shows a case where they are also at least double exponentially large.

Example 12. For each $n \geq 1$, we give TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ whose size is polynomial in n and such that $\mathcal{T}_2$ is not a concept conservative extension of $\mathcal{T}_1$, but the elements of $\mathsf{cDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ are of size at least 2^{2^n} for $\Sigma = \mathsf{sig}(\mathcal{T}_1)$. It is instructive to start with the definition of $\mathcal{T}_2$, which is as follows:

$$\begin{aligned}
A &\sqsubseteq \overline{X}_0 \sqcap \cdots \sqcap \overline{X}_{n-1}, \\
\textstyle\bigsqcap_{\sigma \in \{r,s\}} \exists \sigma.(\overline{X}_i \sqcap X_0 \sqcap \cdots \sqcap X_{i-1}) &\sqsubseteq X_i, && \text{for } i < n, \\
\textstyle\bigsqcap_{\sigma \in \{r,s\}} \exists \sigma.(X_i \sqcap X_0 \sqcap \cdots \sqcap X_{i-1}) &\sqsubseteq \overline{X}_i, && \text{for } i < n, \\
\textstyle\bigsqcap_{\sigma \in \{r,s\}} \exists \sigma.(\overline{X_i} \sqcap \overline{X}_j) &\sqsubseteq \overline{X_i}, && \text{for } j < i < n, \\
\textstyle\bigsqcap_{\sigma \in \{r,s\}} \exists \sigma.(X_i \sqcap \overline{X}_j) &\sqsubseteq X_i, && \text{for } j < i < n, \\
X_0 \sqcap \cdots \sqcap X_{n-1} &\sqsubseteq B.
\end{aligned}$$

[8] This should not be confused with the size of uniform interpolants, which can even be triple exponential in $\mathcal{EL}$ [78].

The concept names $X_0, \dots, X_{n-1}$ and $\overline{X}_0, \dots, \overline{X}_{n-1}$ are used to represent a binary counter: if X_i is true, then the i-th bit is positive and if $\overline{X}_i$ is true, then it is negative. These concept names will not be used in $\mathcal{T}_1$ and thus cannot occur in $\mathsf{cDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ for the signature Σ of $\mathcal{T}_1$. Observe that Lines 2-5 implement incrementation of the counter. We are interested in consequences of $\mathcal{T}_2$ that are of the form $C_{2^n} \sqsubseteq B$, where

$$C_0 = A, \qquad\qquad C_i = \exists r.C_{i-1} \sqcap \exists s.C_{i-1},$$

which we would like to be the smallest elements of $\mathsf{cDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$. Clearly, C_{2^n} is of size at least 2^{2^n}. Ideally, we would like to employ a trivial TBox $\mathcal{T}_1$ that uses only signature $\Sigma = \{A, B, r, s\}$ and has no interesting consequences (only tautologies). If we do exactly this, though, there are some undesired (single exponentially) 'small' CIs in $\mathsf{cDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$, in particular $C'_n \sqsubseteq B$, where

$$C'_0 = A, \qquad\qquad C'_i = A \sqcap \exists r.C_{i-1} \sqcap \exists s.C_{i-1}.$$

Intuitively, the multiple use of A messes up our counter, making bits both true and false at the same time and resulting in all concept names X_i to become true already after travelling n steps along r. We thus have to achieve that these CIs are already consequences of $\mathcal{T}_1$. To this end, we define $\mathcal{T}_1$ as

$$\exists\sigma.A \sqsubseteq A', \qquad A' \sqcap A \sqsubseteq B', \qquad \exists\sigma.B' \sqsubseteq B', \qquad B' \sqsubseteq B$$

where σ ranges over $\{r, s\}$, and include these concept assertions also in $\mathcal{T}_2$ to achieve $\mathcal{T}_1 \subseteq \mathcal{T}_2$ as required for conservative extensions.

4.4 Concept inseparability for acyclic $\mathcal{EL}$ TBoxes

We show that concept inseparability for *acyclic* $\mathcal{EL}$ TBoxes can be decided in polynomial time and discuss interesting applications to versioning and the logical diff of TBoxes. We remark that TBoxes used in practice are often acyclic, and that, in fact, many biomedical ontologies such as SNOMED CT are acyclic $\mathcal{EL}$ TBoxes or mild extensions thereof.

Concept inseparability of acyclic $\mathcal{EL}$ TBoxes is still far from being a trivial problem. For example, it can be shown that smallest counterexamples from $\mathsf{cDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ can be exponential in size [53]. However, acyclic $\mathcal{EL}$ TBoxes enjoy the pleasant property that if $\mathsf{cDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ is non-empty, then it must contain a concept inclusion of the form $C \sqsubseteq A$ or $A \sqsubseteq C$, with A a concept name. This is a consequence of the following result, established in [53].

Theorem 10. *Suppose $\mathcal{T}_1$ and $\mathcal{T}_2$ are acyclic $\mathcal{EL}$ TBoxes and Σ a signature. If $C \sqsubseteq D \in \mathsf{cDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$, then there exist subconcepts C' of C and D' of D such that $C' \sqsubseteq D' \in \mathsf{cDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$, and C' or D' is a concept name.*

Theorem 10 implies that every logical difference between $\mathcal{T}_1$ and $\mathcal{T}_2$ is associated with a concept name from Σ (that must occur in $\mathcal{T}_2$). This opens up an interesting perspective for representing the logical difference between TBoxes since, in contrast to $\mathsf{cDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$, the set of all concept names A that are associated with a logical difference $C \sqsubseteq A$ or $A \sqsubseteq C$ is finite. One can thus summarize for the user the logical difference between two TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ by presenting her with the list of all such concept names A.

Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be acyclic $\mathcal{EL}$ TBoxes and Σ a signature. We define the set of *left-hand* Σ-concept difference witnesses $\mathsf{cWtn}^{\mathsf{lhs}}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ (or *right-hand* Σ-concept difference witnesses $\mathsf{cWtn}^{\mathsf{rhs}}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$) as the set of all $A \in \Sigma \cap \mathsf{N_C}$ such that there exists a concept C with $A \sqsubseteq C \in \mathsf{cDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ (or $C \sqsubseteq A \in \mathsf{cDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$, respectively). Note that, by Theorem 10, $\mathcal{T}_1$ Σ-concept entails $\mathcal{T}_2$ iff $\mathsf{cWtn}^{\mathsf{lhs}}_\Sigma(\mathcal{T}_1, \mathcal{T}_2) = \mathsf{cWtn}^{\mathsf{rhs}}_\Sigma(\mathcal{T}_1, \mathcal{T}_2) = \emptyset$. In the following, we explain how both sets can be computed in polynomial time. The constructions are from [53].

The tractability of computing $\mathsf{cWtn}^{\mathsf{lhs}}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ follows from Theorem 7 and the fact that $\mathcal{EL}$ has canonical models. More specifically, for every $\mathcal{EL}$ TBox $\mathcal{T}$ and $\mathcal{EL}$ concept C one can construct in polynomial time a canonical pointed interpretation $(\mathcal{I}_{\mathcal{T},C}, d)$ such that, for any $\mathcal{EL}$ concept D, we have $d \in D^{\mathcal{I}_{\mathcal{T},C}}$ iff $\mathcal{T} \models C \sqsubseteq D$. Then Theorem 7 yields for any $A \in \Sigma$ that

$$A \in \mathsf{cWtn}^{\mathsf{lhs}}_\Sigma(\mathcal{T}_1, \mathcal{T}_2) \quad \Longleftrightarrow \quad (\mathcal{I}_2, d_2) \not\leq^{\mathsf{sim}}_\Sigma (\mathcal{I}_1, d_1)$$

where $(\mathcal{I}_i, d_i)$ are canonical pointed interpretations for $\mathcal{T}_i$ and A, $i = 1, 2$. Since the existence of a simulation between polynomial size pointed interpretations can be decided in polynomial time [23], we have proved the following result.

Theorem 11. *For $\mathcal{EL}$ TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ and a signature Σ, $\mathsf{cWtn}^{\mathsf{lhs}}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ can be computed in polynomial time.*

We now consider $\mathsf{cWtn}^{\mathsf{rhs}}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$, that is, Σ-CIs of the form $C \sqsubseteq A$. To check, for a concept name $A \in \Sigma$, whether $A \in \mathsf{cWtn}^{\mathsf{rhs}}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$, ideally we would like to compute all concepts C such that $\mathcal{T}_1 \not\models C \sqsubseteq A$ and then check whether $\mathcal{T}_2 \models C \sqsubseteq A$. Unfortunately, there are infinitely many such concepts C. Note that if $\mathcal{T}_2 \models C \sqsubseteq A$ and C' is more specific than C in the sense that $\models C' \sqsubseteq C$, then $\mathcal{T}_2 \models C' \sqsubseteq A$. If there is a *most specific* concept C_A among all C with $\mathcal{T}_1 \not\models C \sqsubseteq A$, it thus suffices to compute this C_A and check whether $\mathcal{T}_2 \models C_A \sqsubseteq A$. Intuitively, though, such a C_A is only guaranteed to exist when we admit infinitary concepts. The solution is to represent C_A not as a concept, but as a TBox. We only demonstrate this approach by an example and refer the interested reader to [53] for further details.

Example 13. (a) Suppose that $\mathcal{T}_1 = \{A \equiv \exists r.A_1\}$, $\mathcal{T}_2 = \{A \equiv \exists r.A_2\}$ and $\Sigma = \{A, A_1, A_2, r\}$. A concept C_A such that $\mathcal{T}_1 \not\models C_A \sqsubseteq A$ should have neither A nor $\exists r.A_1$ as top level conjuncts. This can be captured by the CIs

$$X_A \sqsubseteq A_1 \sqcap A_2 \sqcap \exists r.(A \sqcap A_2 \sqcap \exists r.X_\Sigma), \tag{1}$$

$$X_\Sigma \sqsubseteq A \sqcap A_1 \sqcap A_2 \sqcap \exists r.X_\Sigma, \tag{2}$$

where X_A and X_Σ are fresh concept names and X_A represents the most specific concept C_A with $\mathcal{T}_1 \not\models C_A \sqsubseteq A$. We have $\mathcal{T}_2 \cup \{(1), (2)\} \models X_A \sqsubseteq A$ and thus $A \in \mathsf{cWtn}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$.

(b) Consider next $\mathcal{T}_1 = \{A \equiv \exists r.A_1 \sqcap \exists r.A_2\}$, $\mathcal{T}_2 = \{A \equiv \exists r.A_2\}$ and $\Sigma = \{A, A_1, A_2, r\}$. A concept C_A with $\mathcal{T}_1 \not\models C_A \sqsubseteq A$ should not have both $\exists r.A_1$ and $\exists r.A_2$ as top level conjuncts. Thus the most specific C_A should contain exactly one of these top level conjuncts, which gives rise to a choice. We use the CIs

$$X_A^1 \sqsubseteq A_1 \sqcap A_2 \sqcap \exists r.(A \sqcap A_2 \sqcap \exists r.X_\Sigma), \tag{3}$$

$$X_A^2 \sqsubseteq A_1 \sqcap A_2 \sqcap \exists r.(A \sqcap A_1 \sqcap \exists r.X_\Sigma), \tag{4}$$

where, intuitively, the disjunction of X_A^1 and X_A^2 represents the most specific C_A. We have $\mathcal{T}_2 \cup \{(3), (2)\} \models X_A^1 \sqsubseteq A$ and thus $A \in \mathsf{cWtn}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$.

The following result is proved by generalizing the examples given above.

Theorem 12. *For $\mathcal{EL}$ TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ and signatures Σ, $\mathsf{cWtn}_\Sigma^{\mathsf{rhs}}(\mathcal{T}_1, \mathcal{T}_2)$ can be computed in polynomial time.*

The results stated above can be generalized to extensions of acyclic $\mathcal{EL}$ with role inclusions and domain and range restrictions and have been implemented in the CEX tool for computing logical difference [53].

An alternative approach to computing right-hand Σ-concept difference witnesses based on checking for the existence of a simulations between polynomial size hypergraphs has been introduced in [67]. It has recently been extended [32] to the case of unrestricted $\mathcal{EL}$ TBoxes; the hypergraphs then become exponential in the size of the input.

5 Model Inseparability

We consider inseparability relations according to which two TBoxes are indistinguishable w.r.t. a signature Σ in case their models coincide when restricted to Σ. A central observation is that two TBoxes are Σ-model inseparable iff they cannot be distinguished by entailment of a second-order (SO) sentence in Σ. As a consequence, model inseparability implies concept inseparability for any DL $\mathcal{L}$ and is thus language independent and very robust. It is particularly useful when a user is not committed to a certain DL or is interested in more than just terminological reasoning.

We start this section with introducing model inseparability and the related notions of model entailment and model conservative extensions. We then look at the relationship between these notions and also compare model inseparability to concept inseparability. We next discuss complexity. It turns out that model inseparability is undecidable for almost all DLs, including $\mathcal{EL}$, with the exception of some *DL-Lite* dialects. Interestingly, by restricting the signature Σ to be a set concept names, one can often restore decidability. We then move to model inseparability in the case in which one TBox is empty, which is of particular interest for applications in ontology reuse and module extraction. While this restricted case is still undecidable in $\mathcal{EL}$, it is decidable for acyclic $\mathcal{EL}$ TBoxes.

We close the section by discussing approximations of model inseparability that play an important role in module extraction.

Two interpretation $\mathcal{I}$ and $\mathcal{J}$ *coincide for* a signature Σ, written $\mathcal{I} =_\Sigma \mathcal{J}$, if $\Delta^\mathcal{I} = \Delta^\mathcal{J}$ and $X^\mathcal{I} = X^\mathcal{J}$ for all $X \in \Sigma$. Our central definitions are now as follows.

Definition 8 (model inseparability, entailment and conservative extensions). Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be TBoxes and let Σ be a signature. Then

- the *Σ-model difference* between $\mathcal{T}_1$ and $\mathcal{T}_2$ is the set $\mathsf{mDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ of all models $\mathcal{I}$ of $\mathcal{T}_1$ such that there does not exist a model $\mathcal{J}$ of $\mathcal{T}_2$ with $\mathcal{J} =_\Sigma \mathcal{I}$;
- $\mathcal{T}_1$ *Σ-model entails* $\mathcal{T}_2$ if $\mathsf{mDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2) = \emptyset$;
- $\mathcal{T}_1$ and $\mathcal{T}_2$ are *Σ-model inseparable* if $\mathcal{T}_1$ Σ-model entails $\mathcal{T}_2$ and vice versa;
- $\mathcal{T}_2$ is a *model conservative extension of* $\mathcal{T}_1$ if $\mathcal{T}_2 \supseteq \mathcal{T}_1$ and $\mathcal{T}_1$ and $\mathcal{T}_2$ are $\mathsf{sig}(\mathcal{T}_1)$-model inseparable.

Similarly to concept entailment (Example 3), model entailment coincides with logical entailment when $\Sigma \supseteq \mathsf{sig}(\mathcal{T}_1 \cup \mathcal{T}_2)$. We again recommend to the reader to verify this to become acquainted with the definitions. Also, one can show as in the proof from Example 4 that definitorial extensions are always model conservative extensions.

Regarding the relationship between concept inseparability and model inseparability, we note that the latter implies the former. The proof of the following result goes through for any DL $\mathcal{L}$ that enjoys a coincidence lemma (that is, for any DL, and even when $\mathcal{L}$ is the set of all second-order sentences).

Theorem 13. *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be TBoxes formulated in some DL $\mathcal{L}$ and Σ a signature such that $\mathcal{T}_1$ Σ-model entails $\mathcal{T}_2$. Then $\mathcal{T}_1$ Σ-concept entails $\mathcal{T}_2$.*

Proof. Suppose $\mathcal{T}_1$ Σ-model entails $\mathcal{T}_2$, and let α be a Σ-inclusion in $\mathcal{L}$ such that $\mathcal{T}_2 \models \alpha$. We have to show that $\mathcal{T}_1 \models \alpha$. Let $\mathcal{I}$ be a model of $\mathcal{T}_1$. There is a model $\mathcal{J}$ of $\mathcal{T}_2$ such that $\mathcal{J} =_\Sigma \mathcal{I}$. Then $\mathcal{J} \models \alpha$, and so $\mathcal{I} \models \alpha$ since $\mathsf{sig}(\alpha) \subseteq \Sigma$. □

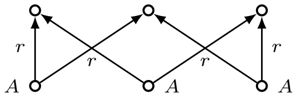
As noted, Theorem 13 even holds when $\mathcal{L}$ is the set of all SO-sentences. Thus, if $\mathcal{T}_1$ Σ-model entails $\mathcal{T}_2$ then, for every SO-sentence φ in the signature Σ, $\mathcal{T}_2 \models \varphi$ implies $\mathcal{T}_1 \models \varphi$. It is proved in [55] that, in fact, the latter exactly characterizes Σ-model entailment.

The following example shows that concept inseparability in $\mathcal{ALCQ}$ does not imply model inseparability (similar examples can be given for any DL and even for full first-order logic [55]).

Example 14. Consider the $\mathcal{ALCQ}$ TBoxes and signature from Example 9:

$$\mathcal{T}_1 = \{A \sqsubseteq\ \geq 2\, r.\top\} \qquad \mathcal{T}_2 = \{A \sqsubseteq \exists r.B \sqcap \exists r.\neg B\} \qquad \Sigma = \{A, r\}.$$

We have noted in Example 5 that $\mathcal{T}_1$ and $\mathcal{T}_2$ are Σ-concept inseparable. However, it is easy to see that the following interpretation is in $\mathsf{mDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$.



We note that the relationship between model-based notions of conservative extension and language-dependent notions of conservative extensions was also extensively discussed in the literature on software specification [17, 99, 100, 86, 76].

We now consider the relationship between model entailment and model inseparability. As in the concept case, model inseparability is defined in terms of model entailment and can be decided by two model entailment checks. Conversely, model entailment can be polynomially reduced to model inseparability (in constast to concept inseparability, where this depends on the DL under consideration).

Lemma 2. *In any DL $\mathcal{L}$, model entailment can be polynomially reduced to model inseparability.*

Proof. Assume that we want to decide whether $\mathcal{T}_1$ Σ-model entails $\mathcal{T}_2$ holds. By replacing every non-Σ-symbol X shared by $\mathcal{T}_1$ and $\mathcal{T}_2$ with a fresh symbol X_1 in $\mathcal{T}_1$ and a distinct fresh symbol X_2 in $\mathcal{T}_2$, we can achieve that $\Sigma \supseteq \mathsf{sig}(\mathcal{T}_1) \cap \mathsf{sig}(\mathcal{T}_2)$ without changing the original (non-)Σ-model entailment of $\mathcal{T}_2$ by $\mathcal{T}_1$. We then have that $\mathcal{T}_1$ Σ-model entails $\mathcal{T}_2$ iff $\mathcal{T}_1$ and $\mathcal{T}_1 \cup \mathcal{T}_2$ are Σ-model inseparable. □

The proof of Lemma 2 shows that any DL $\mathcal{L}$ is *robust under joins for model inseparability*, defined analogously to robustness under joins for concept inseparability; see Definition 3.

5.1 Undecidability of model inseparability

Model-inseparability is computationally much harder than concept inseparability. In fact, it is undecidable already for $\mathcal{EL}$ TBoxes [56]. Here, we give a short and transparent proof showing that model conservative extensions are undecidable in $\mathcal{ALC}$. The proof is by reduction of the following undecidable $\mathbb{N} \times \mathbb{N}$ *tiling problem* [8, 89, 13]: given a finite set $\mathfrak{T}$ of *tile types* T, each with four colors *left*(T), *right*(T), *up*(T) and *down*(T), decide whether $\mathfrak{T}$ *tiles* the grid $\mathbb{N} \times \mathbb{N}$ in the sense that there exists a function (called a *tiling*) τ from $\mathbb{N} \times \mathbb{N}$ to $\mathfrak{T}$ such that

- $\mathit{up}(\tau(i,j)) = \mathit{down}(\tau(i,j+1))$ and
- $\mathit{right}(\tau(i,j)) = \mathit{left}(\tau(i+1,j))$.

If we think of a tile as a physical 1×1-square with a color on each of its four edges, then a tiling τ of $\mathbb{N} \times \mathbb{N}$ is just a way of placing tiles, each of a type from $\mathfrak{T}$, to cover the $\mathbb{N} \times \mathbb{N}$ grid, with no rotation of the tiles allowed and such that the colors on adjacent edges are identical.

Theorem 14. *In $\mathcal{ALC}$, model conservative extensions are undecidable.*

Proof. Given a set $\mathfrak{T}$ of tile types, we regard each $T \in \mathfrak{T}$ as a concept name and let x and y be role names. Let $\mathcal{T}_1$ be the TBox with the following CIs:

$$\begin{aligned}
&\top \sqsubseteq \bigsqcup_{T \in \mathfrak{T}} T,\\
&T \sqcap T' \sqsubseteq \bot, \quad \text{for } T \neq T',\\
&T \sqcap \exists x.T' \sqsubseteq \bot, \quad \text{for } \mathsf{right}(T) \neq \mathsf{left}(T'),\\
&T \sqcap \exists y.T' \sqsubseteq \bot, \quad \text{for } \mathsf{up}(T) \neq \mathsf{down}(T'),\\
&\top \sqsubseteq \exists x.\top \sqcap \exists y.\top.
\end{aligned}$$

Let $\mathcal{T}_2 = \mathcal{T}_1 \cup \mathcal{T}$, where $\mathcal{T}$ consists of a single CI:

$$\top \sqsubseteq \exists u.(\exists x.B \sqcap \exists x.\neg B) \sqcup \exists u.(\exists y.B \sqcap \exists y.\neg B) \sqcup \exists u.(\exists x.\exists y.B \sqcap \exists y.\exists x.\neg B),$$

where u is a fresh role name and B is a fresh concept name. Let $\Sigma = \mathsf{sig}(\mathcal{T}_1)$. One can show that $\mathcal{T}$ can be satisfied in a model $\mathcal{J} =_\Sigma \mathcal{I}$ iff in $\mathcal{I}$ either x is not functional or y is not functional or $x \circ y \neq y \circ x$. It is not hard to see then that $\mathfrak{T}$ tiles $\mathbb{N} \times \mathbb{N}$ iff $\mathcal{T}_1$ and $\mathcal{T}_2$ are not Σ-model inseparable. □

The only standard DLs for which model inseparability is known to be decidable are certain *DL-Lite* dialects. In fact, it is shown in [61] that Σ-model entailment between TBoxes in the extensions of *DL-Lite*$_{core}$ with Boolean operators and unqualified number restrictions is decidable. The computational complexity remains open and for the extension *DL-Lite*$^{\mathcal{H}}_{core}$ of *DL-Lite*$_{core}$ with role hierarchies, even decidability is open. The decidability proof given in [61] is by reduction to the two-sorted first-order theory of Boolean algebras (BA) combined with Presburger arithmetic (PA) for representing cardinalities of sets. The decidability of this theory, called BAPA, has been first proved in [31]. Here we do not go into the decidability proof, but confine ourselves to giving an instructive example which shows that *uncountable* models have to be considered when deciding model entailment in *DL-Lite*$_{core}$ extended with unqualified number restrictions [61].

Example 15. The TBox $\mathcal{T}_1$ states, using auxiliary role names r and s, that the extension of the concept name B is infinite:

$$\begin{aligned}
\mathcal{T}_1 = \{&\top \sqsubseteq \exists r.\top,\ \exists r^-.\top \sqsubseteq \exists s.\top,\ \exists s^-.\top \sqsubseteq B,\\
&B \sqsubseteq \exists s.\top,\ (\geq 2\, s^-.\top) \sqsubseteq \bot,\ \exists r^-.\top \sqcap \exists s^-.\top \sqsubseteq \bot\}.
\end{aligned}$$

The TBox $\mathcal{T}_2$ states that p is an injective function from A to B:

$$\mathcal{T}_2 = \{A \equiv \exists p.\top,\ \ \exists p^-.\top \sqsubseteq B,\ \ (\geq 2\, p.\top) \sqsubseteq \bot,\ \ (\geq 2\, p^-.\top) \sqsubseteq \bot\}.$$

Let $\Sigma = \{A, B\}$. There exists an uncountable model $\mathcal{I}$ of $\mathcal{T}_1$ with uncountable $A^{\mathcal{I}}$ and at most countable $B^{\mathcal{I}}$. Thus, there is no injection from $A^{\mathcal{I}}$ to $B^{\mathcal{I}}$, and so $\mathcal{I} \in \mathsf{mDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ and $\mathcal{T}_1$ does not Σ-model entail $\mathcal{T}_2$. Observe, however, that if $\mathcal{I}$ is a countably infinite model of $\mathcal{T}_1$, then there is always an injection from $A^{\mathcal{I}}$

to $B^{\mathcal{I}}$. Thus, in this case there exists a model $\mathcal{I}'$ of $\mathcal{T}_2$ with $\mathcal{I}' =_{\Sigma} \mathcal{I}$. It follows that uncountable models of $\mathcal{T}_1$ are needed to prove that $\mathcal{T}_1$ does not Σ-model entail $\mathcal{T}_2$.

An interesting way to make Σ-model inseparability decidable is to require that Σ contains only concept names. We show that, in this case, one can use the standard filtration technique from modal logic to show that there always exists a counterexample to Σ-model inseparability of at most exponential size (in sharp contrast to Example 15).

Lemma 3. *Suppose $\mathcal{T}_1$ and $\mathcal{T}_2$ are $\mathcal{ALC}$ TBoxes and Σ contains concept names only. If $\mathsf{mDiff}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2) \neq \emptyset$, then there is an interpretation $\mathcal{I}$ in $\mathsf{mDiff}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2)$ such that $|\Delta^{\mathcal{I}}| \leq 2^{|\mathcal{T}_1|+|\mathcal{T}_2|}$.*

Proof. Assume $\mathcal{I} \in \mathsf{mDiff}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2)$. Define an equivalence relation $\sim \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ by setting $d \sim d'$ iff, for all $C \in \mathsf{sub}(\mathcal{T}_1 \cup \mathcal{T}_2)$, we have $d \in C^{\mathcal{I}_1}$ iff $d' \in C^{\mathcal{I}_2}$. Let $[d] = \{d' \in \Delta^{\mathcal{I}} \mid d' \sim d\}$. Define an interpretation $\mathcal{I}'$ by taking

$$\begin{aligned}
\Delta^{\mathcal{I}'} &= \{[d] \mid d \in \Delta^{\mathcal{I}}\},\\
A^{\mathcal{I}'} &= \{[d] \mid d \in A^{\mathcal{I}}\} \text{ for all } A \in \mathsf{sub}(\mathcal{T}_1),\\
r^{\mathcal{I}'} &= \{([d],[d']) \mid \exists e \in [d] \exists e' \in [d'] (e,e') \in r^{\mathcal{I}}\} \text{ for all role names } r.
\end{aligned}$$

It is not difficult to show that $d \in C^{\mathcal{I}}$ iff $[d] \in C^{\mathcal{I}'}$ for all $d \in \Delta^{\mathcal{I}}$ and $C \in \mathsf{sub}(\mathcal{T}_1)$. Thus $\mathcal{I}'$ is a model of $\mathcal{T}_1$. We now show that there does not exist a model $\mathcal{J}'$ of $\mathcal{T}_2$ with $\mathcal{I}' =_{\Sigma} \mathcal{J}'$. For a proof by contradiction, assume that such a $\mathcal{J}'$ exists. We define a model $\mathcal{J}$ of $\mathcal{T}_2$ with $\mathcal{J} =_{\Sigma} \mathcal{I}$, and thus derive a contradiction to the assumption that $\mathcal{I} \in \mathsf{mDiff}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2)$. To this end, let $A^{\mathcal{J}} = A^{\mathcal{I}}$ for all $A \in \Sigma$ and set

$$\begin{aligned}
A^{\mathcal{J}} &= \{d \mid [d] \in A^{\mathcal{J}}\} \text{ for all } A \notin \Sigma,\\
r^{\mathcal{J}} &= \{(d,d') \mid ([d],[d']) \in r^{\mathcal{J}}\} \text{ for all role names } r.
\end{aligned}$$

Note that the role names (which are all not in Σ), are interpreted in a 'maximal' way. It can be proved that $d \in C^{\mathcal{J}}$ iff $[d] \in C^{\mathcal{J}'}$ for all $d \in \Delta^{\mathcal{I}}$ and $C \in \mathsf{sub}(\mathcal{T}_2)$. Thus $\mathcal{J}$ is a model of $\mathcal{T}_2$ and we have derived a contradiction. □

Using the bounded model property established in Lemma 3, one can prove a coNExp$^{\text{NP}}$ upper bound for model inseparability. A matching lower bound and several extensions of this result are proved in [56].

Theorem 15. *In $\mathcal{ALC}$, Σ-model inseparability is* coNExp$^{\text{NP}}$*-complete when Σ is restricted to sets of concept names.*

Proof. We sketch the proof of the upper bound. It is sufficient to show that one can check in NExp$^{\text{NP}}$ whether $\mathsf{mDiff}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2) \neq \emptyset$. By Lemma 3, one can do this by guessing a model $\mathcal{I}$ of $\mathcal{T}_1$ of size at most $2^{|\mathcal{T}_1|+|\mathcal{T}_2|}$ and then calling an oracle to verify that there is no model $\mathcal{J}$ of $\mathcal{T}_2$ with $\mathcal{J} =_{\Sigma} \mathcal{I}$. The oracle runs in NP since we can give it the guessed $\mathcal{I}$ as an input, thus we are asking for a model of $\mathcal{T}_2$ of size polynomial in the size of the oracle input.

The lower bound is proved in [56] by a reduction of satisfiability in circumscribed $\mathcal{ALC}$ KBs, which is known to be coNExp$^{\text{NP}}$-hard. □

In [56], Theorem 15 is generalized to $\mathcal{ALCI}$. We conjecture that it can be further extended to most standard DLs that admit the finite model property. For DLs without the finite model property such as $\mathcal{ALCQI}$, we expect that BAPA-based techniques, as used for circumscription in [12], can be employed to obtain an analog of Theorem 15.

5.2 Model inseparability from the empty TBox

We now consider model inseparability in the case where one TBox is empty. To motivate this important case, consider the application of ontology reuse, where one wants to import a TBox $\mathcal{T}_{\mathsf{im}}$ into a TBox $\mathcal{T}$ that is currently being developed. Recall that the result of importing $\mathcal{T}_{\mathsf{im}}$ in $\mathcal{T}$ is the union $\mathcal{T} \cup \mathcal{T}_{\mathsf{im}}$ and that, when importing $\mathcal{T}_{\mathsf{im}}$ into $\mathcal{T}$, the TBox $\mathcal{T}$ is not supposed to interfere with the modeling of the symbols from $\mathcal{T}_{\mathsf{im}}$. We can formalize this requirement by demanding that

- $\mathcal{T} \cup \mathcal{T}_{\mathsf{im}}$ and $\mathcal{T}_{\mathsf{im}}$ are Σ-model inseparable for $\Sigma = \mathsf{sig}(\mathcal{T}_{\mathsf{im}})$.

In this scenario, one has to be prepared for the imported TBox $\mathcal{T}_{\mathsf{im}}$ to be revised. Thus, one would like to design the importing TBox $\mathcal{T}$ such that *any* TBox $\mathcal{T}_{\mathsf{im}}$ can be imported into $\mathcal{T}$ without undesired interaction as long as the signature of $\mathcal{T}_{\mathsf{im}}$ is not changed. Intuitively, $\mathcal{T}$ provides a *safe interface for importing ontologies* that only share symbols from some fixed signature Σ with $\mathcal{T}$. This idea led to the definition of safety for a signature in [25]:

Definition 9. Let $\mathcal{T}$ be an $\mathcal{L}$ TBox. We say that $\mathcal{T}$ is *safe for a signature Σ under model inseparability* if $\mathcal{T} \cup \mathcal{T}_{\mathsf{im}}$ is $\mathsf{sig}(\mathcal{T}_{\mathsf{im}})$-model inseparable from $\mathcal{T}_{\mathsf{im}}$ for all $\mathcal{L}$ TBoxes $\mathcal{T}_{\mathsf{im}}$ with $\mathsf{sig}(\mathcal{T}) \cap \mathsf{sig}(\mathcal{T}_{\mathsf{im}}) \subseteq \Sigma$.

As one quantifies over all TBoxes $\mathcal{T}_{\mathsf{im}}$ in Definition 9, safety for a signature seems hard to deal with algorithmically. Fortunately, it turns out that the quantification can be avoided. This is related to the following robustness property.[9]

Definition 10. A DL $\mathcal{L}$ is said to be *robust under replacement for model inseparability* if, for all $\mathcal{L}$ TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ and signatures Σ, the following condition is satisfied: if $\mathcal{T}_1$ and $\mathcal{T}_2$ are Σ-model inseparable, then $\mathcal{T}_1 \cup \mathcal{T}$ and $\mathcal{T}_2 \cup \mathcal{T}$ are Σ-model inseparable for all $\mathcal{L}$ TBoxes $\mathcal{T}$ with $\mathsf{sig}(\mathcal{T}) \cap \mathsf{sig}(\mathcal{T}_1 \cup \mathcal{T}_2) \subseteq \Sigma$.

The following has been observed in [25]. It again applies to any standard DL, and in fact even to second-order logic.

Theorem 16. *In any DL $\mathcal{L}$, model inseparability is robust under replacement.*

[9] Similar robustness properties and notions of equivalence have been discussed in logic programming, we refer the reader to [75, 65, 28] and references therein. We will discuss this robustness property further in Section 8.

Using robustness under replacement, it can be proved that safety for a signature is nothing but inseparability from the empty TBox, in this way eliminating the quantification over TBoxes used in the original definition. This has first been observed in [25]. The connection to robustness under replacement is from [56].

Theorem 17. *A TBox $\mathcal{T}$ is safe for a signature Σ under model-inseparability iff $\mathcal{T}$ is Σ-model inseparable from the empty TBox.*

Proof. Assume first that $\mathcal{T}$ is not Σ-model inseparable from $\emptyset$. Then $\mathcal{T} \cup \mathcal{T}_{im}$ is not Σ-model inseparable from $\mathcal{T}_{im}$, where $\mathcal{T}_{im}$ is the trivial Σ-TBox $\mathcal{T}_{im} = \{A \sqsubseteq A \mid A \in \Sigma \cap \mathsf{N_C}\} \cup \{\exists r.\top \sqsubseteq \top \mid r \in \Sigma \cap \mathsf{N_R}\}$. Hence $\mathcal{T}$ is not safe for Σ. Now assume $\mathcal{T}$ is Σ-model inseparable from $\emptyset$ and let $\mathcal{T}_{im}$ be a TBox such that $\mathsf{sig}(\mathcal{T}) \cap \mathsf{sig}(\mathcal{T}_{im}) \subseteq \Sigma$. Then it follows from robustness under replacement that $\mathcal{T} \cup \mathcal{T}_{im}$ is $\mathsf{sig}(\mathcal{T}_{im})$-model inseparable from $\mathcal{T}_{im}$. □

By Theorem 17, deciding safety of a TBox $\mathcal{T}$ for a signature Σ under model inseparability amounts to checking Σ-model inseparability from the empty TBox. We thus consider the latter problem as an important special case of model inseparability. Unfortunately, even in $\mathcal{EL}$, model inseparability from the empty TBox is undecidable [56].

Theorem 18. *In $\mathcal{EL}$, model inseparability from the empty TBox is undecidable.*

We now consider acyclic $\mathcal{EL}$ TBoxes as an important special case. As we have mentioned before, many large-scale TBoxes are in fact acyclic $\mathcal{EL}$ TBoxes or mild extensions thereof. Interestingly, model inseparability of acyclic TBoxes from the empty TBox can be decided in polynomial time [56]. The approach is based on a characterization of model inseparability from the empty TBox in terms of certain syntactic and semantic *dependencies*. The following example shows two cases of how an acyclic $\mathcal{EL}$ TBox can fail to be model inseparable from the empty TBox. These two cases will then give rise to two types of syntactic dependencies.

Example 16. (a) Let $\mathcal{T} = \{A \sqsubseteq \exists r.B, B \sqsubseteq \exists s.E\}$ and $\Sigma = \{A, s\}$. Then $\mathcal{T}$ is not Σ-model inseparable from the empty TBox: for the interpretation $\mathcal{I}$ with $\Delta^{\mathcal{I}} = \{d\}$, $A^{\mathcal{I}} = \{d\}$, and $s^{\mathcal{I}} = \emptyset$, there is no model $\mathcal{J}$ of $\mathcal{T}$ with $\mathcal{J} =_{\Sigma} \mathcal{I}$.

(b) Let $\mathcal{T} = \{A_1 \sqsubseteq \exists r.B_1, A_2 \sqsubseteq \exists r.B_2, A \equiv B_1 \sqcap B_2\}$ and $\Sigma = \{A_1, A_2, A\}$. Then $\mathcal{T}$ is not Σ-model inseparable from the empty TBox: for the interpretation $\mathcal{I}$ with $\Delta^{\mathcal{I}} = \{d\}$, $A_1^{\mathcal{I}} = A_2^{\mathcal{I}} = \{d\}$, and $A^{\mathcal{I}} = \emptyset$, there is no model $\mathcal{J}$ of $\mathcal{T}$ with $\mathcal{J} =_{\Sigma} \mathcal{I}$.

Intuitively, in part (a) of Example 16, the reason for separability from the empty TBox is that we can start with a Σ-concept name that occurs on some left-hand side (which is A) and then deduce from it that another Σ-symbol (which is s) must be non-empty. Part (b) is of a slightly different nature. We start with a set of Σ-concept names (which is $\{A_1, A_2\}$) and from that deduce a set of concepts that implies another Σ-concept (which is A) via a concept definition, right-to-left. It turns out that it is convenient to distinguish between these two cases also in general. We first introduce some notation. For an acyclic TBox $\mathcal{T}$, let

- $\mathsf{lhs}(\mathcal{T})$ denote the set of concept names A such that there is some CI $A \equiv C$ or $A \sqsubseteq C$ in $\mathcal{T}$;
- $\mathsf{def}(\mathcal{T})$ denote the set of concept names A such that there is a definition $A \equiv C$ in $\mathcal{T}$;
- $\mathsf{depend}^{\equiv}_{\mathcal{T}}(A)$ be defined exactly as $\mathsf{depend}_{\mathcal{T}}(A)$ in Section 2, except that only concept definitions $A \equiv C$ are considered while concept inclusions $A \sqsubseteq C$ are disregarded.

Definition 11. Let $\mathcal{T}$ be an acyclic $\mathcal{EL}$ TBox, Σ a signature, and $A \in \Sigma$. We say that

- *A has a direct Σ-dependency in $\mathcal{T}$* if $\mathsf{depend}_{\mathcal{T}}(A) \cap \Sigma \neq \emptyset$;
- *A has an indirect Σ-dependency in $\mathcal{T}$* if $A \in \mathsf{def}(\mathcal{T}) \cap \Sigma$ and there are $A_1, \ldots, A_n \in \mathsf{lhs}(\mathcal{T}) \cap \Sigma$ such that $A \notin \{A_1, \ldots, A_n\}$ and

$$\mathsf{depend}^{\equiv}_{\mathcal{T}}(A) \setminus \mathsf{def}(\mathcal{T}) \subseteq \bigcup_{1 \leq i \leq n} \mathsf{depend}_{\mathcal{T}}(A_i).$$

We say that $\mathcal{T}$ *contains an (in)direct Σ-dependency* if there is an $A \in \Sigma$ that has an (in)direct Σ-dependency in $\mathcal{T}$.

It is proved in [56] that, for every acyclic $\mathcal{EL}$ TBox $\mathcal{T}$ and signature Σ, $\mathcal{T}$ is Σ-model inseparable from the empty TBox iff $\mathcal{T}$ has neither direct nor indirect Σ-dependencies. It can be decided in PTime in a straightforward way whether a given $\mathcal{EL}$ TBox contains a direct Σ-dependency. For indirect Σ-dependencies, this is less obvious since we start with a set of concept names from $\mathsf{lhs}(\mathcal{T}) \cap \Sigma$. Fortunately, it can be shown that if a concept name $A \in \Sigma$ has an indirect Σ-dependency in $\mathcal{T}$ induced by concept names $A_1, \ldots, A_n \in \mathsf{lhs}(\mathcal{T}) \cap \Sigma$, then A has an indirect Σ-dependency in $\mathcal{T}$ induced by the set of concept names $(\mathsf{lhs}(\mathcal{T}) \cap \Sigma) \setminus \{A\}$. We thus only need to consider the latter set.

Theorem 19. *In $\mathcal{EL}$, model inseparability of acyclic TBoxes from the empty TBox is in* PTime.

Also in [56], Theorem 19 is extended from $\mathcal{EL}$ to $\mathcal{ELI}$, and it is shown that, in $\mathcal{ALC}$ and $\mathcal{ALCI}$, model inseparability from the empty TBox is Π^p_2-complete for acyclic TBoxes.

5.3 Locality-based approximations

We have seen in the previous section that model inseparability from the empty TBox is of great practical value in the context of ontology reuse, that it is undecidable even in $\mathcal{EL}$, and that decidability can (sometimes) be regained by restricting TBoxes to be acyclic. In the non-acyclic case, one option is to resort to approximations from above. This leads to the (semantic) notion of $\emptyset$-locality and its syntactic companion $\bot$-locality. We discuss the former in this section and the latter in Section 8.

A TBox $\mathcal{T}$ is called *$\emptyset$-local w.r.t. a signature Σ* if, for every interpretation $\mathcal{I}$, there exists a model $\mathcal{J}$ of $\mathcal{T}$ such that $\mathcal{I} =_\Sigma \mathcal{J}$ and $A^\mathcal{J} = r^\mathcal{J} = \emptyset$, for all $A \in \mathsf{N_C} \setminus \Sigma$ and $r \in \mathsf{N_R} \setminus \Sigma$; in other words, every interpretation of Σ-symbols can be trivially extended to a model of $\mathcal{T}$ by interpreting non-Σ symbols as the empty set. Note that, if $\mathcal{T}$ is $\emptyset$-local w.r.t. Σ, then it is Σ-model inseparable from the empty TBox and thus, by Theorem 17, safe for Σ under model inseparability. The following example shows that the converse does not hold.

Example 17. Let $\mathcal{T} = \{A \sqsubseteq B\}$ and $\Sigma = \{A\}$. Then $\mathcal{T}$ is Σ-model inseparable from $\emptyset$, but $\mathcal{T}$ is not $\emptyset$-local w.r.t. Σ.

In contrast to model inseparability, $\emptyset$-locality is decidable also in $\mathcal{ALC}$ and beyond, and is computationally not harder than standard reasoning tasks such as satisfiability. The next procedure for checking $\emptyset$-locality was given in [44].

Theorem 20. *Let $\mathcal{T}$ be an $\mathcal{ALCQI}$ TBox and Σ a signature. Suppose $\mathcal{T}{\restriction}_{\Sigma=\emptyset}$ is obtained from $\mathcal{T}$ by replacing all concepts of the form A, $\exists r.C$, $\exists r^-.C$, $(\geq n\, r.C)$ and $(\geq n\, r^-.C)$ with $\bot$ whenever $A \notin \Sigma$ and $r \notin \Sigma$. Then $\mathcal{T}$ is $\emptyset$-local w.r.t. Σ iff $\mathcal{T}{\restriction}_{\Sigma=\emptyset}$ is logically equivalent to the empty TBox.*

While Theorem 20 is stated here for $\mathcal{ALCQI}$—the most expressive DL considered in this paper—the original result in [42] is more general and applies to $\mathcal{SHOIQ}$ knowledge bases. There is also a dual notion of Δ-locality [25], in which non-Σ symbols are interpreted as the entire domain and which can also be reduced to logical equivalence.

We also remark that, unlike model inseparability from the empty TBox, model inseparability cannot easily be reduced to logical equivalence in the style of Theorem 20.

Example 18. Let $\mathcal{T} = \{A \sqsubseteq B \sqcup C\}$, $\mathcal{T}' = \{A \sqsubseteq B\}$ and $\Sigma = \{A, B\}$. Then the TBoxes $\mathcal{T}{\restriction}_{\Sigma=\emptyset}$ and $\mathcal{T}'{\restriction}_{\Sigma=\emptyset}$ are logically equivalent, yet $\mathcal{T}$ is not Σ-model inseparable from $\mathcal{T}'$.

$\emptyset$-locality and its syntactic companion $\bot$-locality are prominently used in ontology modularization [43, 25, 42, 93]. A subset $\mathcal{M}$ of $\mathcal{T}$ is called a *$\emptyset$-local Σ-module* of $\mathcal{T}$ if $\mathcal{T} \setminus \mathcal{M}$ is $\emptyset$-local w.r.t. Σ. It can be shown that every $\emptyset$-local Σ-module $\mathcal{M}$ of $\mathcal{T}$ is self-contained (that is, $\mathcal{M}$ is Σ-model inseparable from $\mathcal{T}$) and depleting (that is, $\mathcal{T} \setminus \mathcal{M}$ is $\Sigma \cup \mathsf{sig}(\mathcal{M})$-model inseparable from the empty TBox). In addition, $\emptyset$-local modules are also *subsumer-preserving*, that is, for every $A \in \Sigma \cap \mathsf{N_C}$ and $B \in \mathsf{N_C}$, if $\mathcal{T} \models A \sqsubseteq B$ then $\mathcal{M} \models A \sqsubseteq B$. This property is particular useful in modular reasoning [91, 90, 92].

A $\emptyset$-local module of a given ontology $\mathcal{T}$ for a given signature Σ can be computed in a straightforward way as follows. Starting with $\mathcal{M} = \emptyset$, iteratively add to $\mathcal{M}$ every $\alpha \in \mathcal{T}$ such that $\alpha{\restriction}_{\Sigma \cup \mathsf{sig}(\mathcal{M})=\emptyset}$ is not a tautology until $\mathcal{T} \setminus \mathcal{M}$ is $\emptyset$-local w.r.t. $\Sigma \cup \mathsf{sig}(\mathcal{M})$. The resulting module might be larger than necessary because this procedure actually generates a $\emptyset$-local $\Sigma \cup \mathsf{sig}(\mathcal{M})$-module rather than only a Σ-module and because $\emptyset$-locality overapproximates model inseparability, but in most practical cases results in reasonably small modules [43].

6 Query Inseparability for KBs

In this section, we consider inseparability of KBs rather than of TBoxes. One main application of KBs is to provide access to the data stored in their ABox by means of database-style queries, also taking into account the knowledge from the TBox to compute more complete answers. This approach to querying data is known as *ontology-mediated querying* [11], and it is a core part of the *ontology-based data access* (OBDA) paradigm [85]. In many applications of KBs, a reasonable notion of inseparability is thus the one where both KBs are required to give the same answers to all relevant queries that a user might pose. Of course, such an inseparability relation depends on the class of relevant queries and on the signature that we are allowed to use in the query. We will consider the two most important query languages, which are conjunctive queries (CQs) and unions thereof (UCQs), and their rooted fragments, rCQs and rUCQs.

We start the section by introducing query inseparability of KBs and related notions of query entailment and query conservative extensions. We then discuss the connection to the logical equivalence of KBs, how the choice of a query language impacts query inseparability, and the relation between query entailment and query inseparability. Next, we give model-theoretic characterizations of query inseparability which are based on model classes that are complete for query answering and on (partial or full) homomorphisms. We then move to decidability and complexity, starting with $\mathcal{ALC}$ and then proceeding to *DL-Lite*, $\mathcal{EL}$, and Horn-$\mathcal{ALC}$. In the case of $\mathcal{ALC}$, inseparability in terms of CQs turns out to be undecidable while inseparability in terms of UCQs is decidable in 2ExpTime (and the same is true for the rooted versions of these query languages). In the mentioned Horn DLs, CQ inseparability coincides with UCQ inseparability and is decidable, with the complexity ranging from PTime for $\mathcal{EL}$ via ExpTime for *DL-Lite*$^{\mathcal{H}}_{core}$, *DL-Lite*$^{\mathcal{H}}_{horn}$, and Horn-$\mathcal{ALC}$ to 2ExpTime for Horn-$\mathcal{ALCI}$.

Definition 12 (query inseparability, entailment and conservative extensions). Let $\mathcal{K}_1$ and $\mathcal{K}_2$ be KBs, Σ a signature, and $\mathcal{Q}$ a class of queries. Then

- the Σ-$\mathcal{Q}$ *difference* between $\mathcal{K}_1$ and $\mathcal{K}_2$ is the set $\mathsf{qDiff}^{\mathcal{Q}}_{\Sigma}(\mathcal{K}_1, \mathcal{K}_2)$ of all $\boldsymbol{q}(\boldsymbol{a})$ such that $\boldsymbol{q}(\boldsymbol{x}) \in \mathcal{Q}_{\Sigma}$, $\boldsymbol{a} \subseteq \mathsf{ind}(\mathcal{A}_2)$, $\mathcal{K}_2 \models \boldsymbol{q}(\boldsymbol{a})$ and $\mathcal{K}_1 \not\models \boldsymbol{q}(\boldsymbol{a})$;
- $\mathcal{K}_1$ Σ-$\mathcal{Q}$ *entails* $\mathcal{K}_2$ if $\mathsf{qDiff}^{\mathcal{Q}}_{\Sigma}(\mathcal{K}_1, \mathcal{K}_2) = \emptyset$;
- $\mathcal{K}_1$ and $\mathcal{K}_2$ are Σ-$\mathcal{Q}$ *inseparable* if $\mathcal{K}_1$ Σ-$\mathcal{Q}$ entails $\mathcal{K}_2$ and vice versa;
- $\mathcal{K}_2$ is a $\mathcal{Q}$-*conservative extension of* $\mathcal{K}_1$ if $\mathcal{K}_2 \supseteq \mathcal{K}_1$, and $\mathcal{K}_1$ and $\mathcal{K}_2$ are $\mathsf{sig}(\mathcal{K}_1)$-$\mathcal{Q}$ inseparable.

If $\boldsymbol{q}(\boldsymbol{a}) \in \mathsf{qDiff}^{\mathcal{Q}}_{\Sigma}(\mathcal{K}_1, \mathcal{K}_2)$, then we say that $\boldsymbol{q}(\boldsymbol{a})$ Σ-$\mathcal{Q}$ *separates* $\mathcal{K}_1$ and $\mathcal{K}_2$.

Note that slight variations of the definition of query inseparability are possible; for example, one can allow signatures to also contain individual names and then consider only query answers that consist of these names [14].

Query inseparability is a coarser relationship between KBs than logical equivalence even when $\Sigma \supseteq \mathsf{sig}(\mathcal{K}_1) \cup \mathsf{sig}(\mathcal{K}_2)$. Recall that this is in sharp contrast to concept and model inseparability, for which we observed that they coincide with logic equivalence under analogous assumptions on Σ.

Example 19 (query inseparability and logical equivalence). Let $\mathcal{K}_i = (\mathcal{T}_i, \mathcal{A}_i)$, $i = 1, 2$, where $\mathcal{A}_1 = \{A(c)\}$, $\mathcal{T}_1 = \{A \sqsubseteq B\}$, $\mathcal{A}_2 = \{A(c), B(c)\}$, and $\mathcal{T}_2 = \emptyset$. Then $\mathcal{K}_1$ and $\mathcal{K}_2$ are Σ-UCQ inseparable for any signature Σ but clearly $\mathcal{K}_1$ and $\mathcal{K}_2$ are not logically equivalent.

This example shows that there are drastic logical differences between KBs that cannot be detected by UCQs. This means that, when we aim to replace a KB with a query inseparable one, we have significant freedom to modify the KB. In the example above, we went from a KB with a non-empty TBox to a KB with an empty TBox, which should be easier to deal with when queries have to be answered efficiently.

We now compare the notions of $\mathcal{Q}$ inseparability induced by different choices of the query language $\mathcal{Q}$. A first observation is that, for Horn DLs such as $\mathcal{EL}$, there is no difference between UCQ inseparability and CQ inseparability. The same applies to rCQs and rUCQs. This follows from the fact that KBs formulated in a Horn DL have a universal model, that is, a single model that gives the same answers to queries as the KB itself—see Section 6.1 for more details.[10]

Theorem 21. *Let $\mathcal{K}_1$ and $\mathcal{K}_2$ be KBs formulated in a Horn DL, and let Σ be a signature. Then*

(i) $\mathcal{K}_1$ Σ-UCQ entails $\mathcal{K}_2$ iff $\mathcal{K}_1$ Σ-CQ entails $\mathcal{K}_2$;
(ii) $\mathcal{K}_1$ Σ-rUCQ entails $\mathcal{K}_2$ iff $\mathcal{K}_1$ Σ-rCQ entails $\mathcal{K}_2$.

The equivalences above do not hold for DLs that are not Horn, as shown by the following example:

Example 20. Let $\mathcal{K}_i = (\mathcal{T}_i, \mathcal{A})$, for $i = 1, 2$, be the $\mathcal{ALC}$ KBs where $\mathcal{T}_1 = \emptyset$, $\mathcal{T}_2 = \{A \sqsubseteq B_1 \sqcup B_2\}$, and $\mathcal{A} = \{A(c)\}$. Let $\Sigma = \{A, B_1, B_2\}$. Then $\mathcal{K}_1$ Σ-CQ entails $\mathcal{K}_2$, but the UCQ (actually rUCQ) $\boldsymbol{q}(x) = B_1(x) \vee B_2(x)$ shows that $\mathcal{K}_1$ does not Σ-UCQ entail $\mathcal{K}_2$.

As in the case of concept and model inseparability (of TBoxes), it is instructive to consider the connection between query entailment and query inseparability. As before, query inseparability is defined in terms of query entailment. The converse direction is harder to analyze. Recall that, for concept and model inseparability, we employed robustness under joins to reduce entailment to inseparability. Robustness under joins is defined as follows for Σ-$\mathcal{Q}$-inseparability: if $\Sigma \supseteq \mathsf{sig}(\mathcal{K}_1) \cap \mathsf{sig}(\mathcal{K}_2)$, then $\mathcal{K}_1$ Σ-$\mathcal{Q}$ entails $\mathcal{K}_2$ iff $\mathcal{K}_1$ and $\mathcal{K}_1 \cup \mathcal{K}_2$ are Σ-$\mathcal{Q}$ inseparable. Unfortunately, this property does not hold.

Example 21. Let $\mathcal{K}_i = (\mathcal{T}_i, \mathcal{A})$, $i = 1, 2$, be Horn-$\mathcal{ALC}$ KBs with

$$\mathcal{T}_1 = \{A \sqsubseteq \exists r.B \sqcap \exists r.\neg B\}, \quad \mathcal{T}_2 = \{A \sqsubseteq \exists r.B \sqcap \forall r.B\}, \quad \mathcal{A} = \{A(c)\}.$$

Let $\Sigma = \{A, B, r\}$. Then, for any class of queries $\mathcal{Q}$ introduced above, $\mathcal{K}_1$ Σ-$\mathcal{Q}$ entails $\mathcal{K}_2$ but $\mathcal{K}_1$ and $\mathcal{K}_1 \cup \mathcal{K}_2$ are not Σ-$\mathcal{Q}$ inseparable since $\mathcal{K}_1 \cup \mathcal{K}_2$ is not satisfiable.

[10] In fact, when we say 'Horn DL', we mean a DL in which every KB has a universal model.

Robustness under joins has not yet been studied systematically for query inseparability. While Example 21 shows that query inseparability does not enjoy robustness under joins in Horn-$\mathcal{ALC}$, it is open whether the same is true in $\mathcal{EL}$ and the *DL-Lite* family. Interestingly, there is a (non-trivial) polynomial time reduction of query entailment to query inseparability that works for many Horn DLs and does not rely on robustness under joins [14].

Theorem 22. *Σ-CQ entailment of KBs is polynomially reducible to Σ-CQ inseparability of KBs for any Horn DL containing $\mathcal{EL}$ or DL-Lite$^{\mathcal{H}}_{core}$, and contained in Horn-$\mathcal{ALCHI}$.*

6.1 Model-theoretic criteria for query inseparability

We now provide model-theoretic characterizations of query inseparability. Recall that query inseparability is defined in terms of certain answers and that, given a KB $\mathcal{K}$ and a query $\boldsymbol{q}(\boldsymbol{x})$, a tuple $\boldsymbol{a} \subseteq \mathsf{ind}(\mathcal{K})$ is a certain answer to $\boldsymbol{q}(\boldsymbol{x})$ over $\mathcal{K}$ iff, for every model $\mathcal{I}$ of $\mathcal{K}$, we have $\mathcal{I} \models \boldsymbol{q}(\boldsymbol{a})$. It is well-known that, in many cases, it is actually not necessary to consider *all* models $\mathcal{I}$ of $\mathcal{K}$ to compute certain answers. We say that a class $\boldsymbol{M}$ of models of $\mathcal{K}$ is *complete for $\mathcal{K}$ and a class $\mathcal{Q}$ of queries* if, for every $\boldsymbol{q}(\boldsymbol{x}) \in \mathcal{Q}$, we have $\mathcal{K} \models \boldsymbol{q}(\boldsymbol{a})$ iff $\mathcal{I} \models \boldsymbol{q}(\boldsymbol{a})$ for all $\mathcal{I} \in \boldsymbol{M}$.

In the following, we give some important examples of model classes for which KBs are complete.

Example 22. Given an $\mathcal{ALC}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, we denote by $\boldsymbol{M}^b_{tree}(\mathcal{K})$ the class of all models $\mathcal{I}$ of $\mathcal{K}$ that can be constructed by choosing, for each $a \in \mathsf{ind}(\mathcal{A})$, a tree interpretation $\mathcal{I}_a$ (see Section 4.1) of outdegree bounded by $|\mathcal{T}|$ and with root a, taking their disjoint union, and then adding the pair (a, b) to $r^{\mathcal{I}}$ whenever $r(a, b) \in \mathcal{A}$. It is known that $\boldsymbol{M}^b_{tree}(\mathcal{K})$ is complete for $\mathcal{K}$ and UCQs, and thus for any class of queries considered in this paper [39]. If $\mathcal{K}$ is formulated in Horn-$\mathcal{ALC}$ or in $\mathcal{EL}$, then there is even a single model $\mathcal{C}_{\mathcal{K}}$ in $\boldsymbol{M}^b_{tree}(\mathcal{K})$ such that $\{\mathcal{C}_{\mathcal{K}}\}$ is complete for $\mathcal{K}$ and UCQs, the *universal* (or *canonical*) *model* of $\mathcal{K}$ [73].

If the KB is formulated in an extension of $\mathcal{ALC}$, the class of models needs to be adapted appropriately. The only such extension we are going to consider is $\mathcal{ALCHI}$ and its fragment *DL-Lite*$^{\mathcal{H}}_{core}$. In this case, one needs a more liberal definition of tree interpretation where role edges can point both downwards and upwards and multi edges are allowed. We refer to the resulting class of models as $\boldsymbol{M}^b_{utree}$.

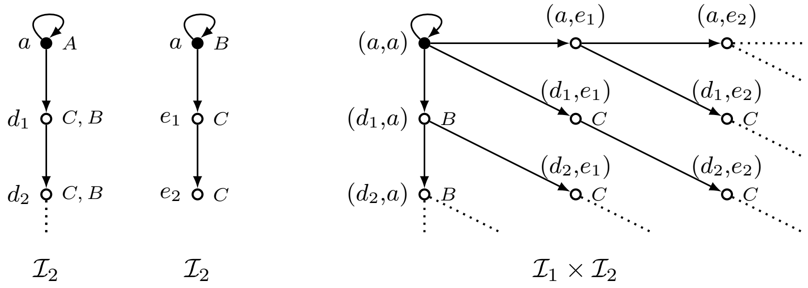
It is well-known from model theory [21] that, for any CQ $\boldsymbol{q}(\boldsymbol{x})$ and any tuple $\boldsymbol{a} \subseteq \mathsf{ind}(\mathcal{K})$, we have $\mathcal{I} \models \boldsymbol{q}(\boldsymbol{a})$ for all $\mathcal{I} \in \boldsymbol{M}$ iff $\prod \boldsymbol{M} \models \boldsymbol{q}(\boldsymbol{a})$, where $\prod \boldsymbol{M}$ is the *direct product* of interpretations in $\boldsymbol{M}$. More precisely, if $\boldsymbol{M} = \{\mathcal{I}_i \mid i \in I\}$, for some set I, then $\prod \boldsymbol{M} = (\Delta^{\prod \boldsymbol{M}}, \cdot^{\prod \boldsymbol{M}})$, where

- $\Delta^{\prod \boldsymbol{M}} = \prod_{i \in I} \Delta^{\mathcal{I}_i}$ is the Cartesian product of the $\Delta^{\mathcal{I}_i}$;
- $a^{\prod \boldsymbol{M}} = (a^{\mathcal{I}_i})_{i \in I}$, for any individual name a;
- $A^{\prod \boldsymbol{M}} = \{(d_i)_{i \in I} \mid d_i \in A^{\mathcal{I}_i} \text{ for all } i \in I\}$, for any concept name A;

– $r^{\prod \boldsymbol{M}} = \{(d_i, e_i)_{i \in I} \mid (d_i, e_i) \in r^{\mathcal{I}_i} \text{ for all } i \in I\}$, for any role name r.

It is to be noted that in general $\prod \boldsymbol{M}$ is *not* a model of $\mathcal{K}$, even if every interpretation in $\boldsymbol{M}$ is.

Example 23. Two interpretations $\mathcal{I}_1$ and $\mathcal{I}_2$ are shown below together with their direct product $\mathcal{I}_1 \times \mathcal{I}_2$ (all the arrows are assumed to be labelled with r):



Now, consider the CQ $\boldsymbol{q}_1(x) = \exists y, z\, (r(x, y) \wedge r(y, z) \wedge B(y) \wedge C(z))$. We clearly have $\mathcal{I}_1 \models \boldsymbol{q}_1(a)$, $\mathcal{I}_2 \models \boldsymbol{q}_1(a)$, and $\mathcal{I}_1 \times \mathcal{I}_2 \models \boldsymbol{q}_1(a)$. On the other hand, for the Boolean CQ $\boldsymbol{q}_2 = \exists x, y, z\ (r(x, y) \wedge r(y, z) \wedge C(y) \wedge B(z))$, we have $\mathcal{I}_1 \models \boldsymbol{q}_2$ but $\mathcal{I}_2 \not\models \boldsymbol{q}_2$, and so $\mathcal{I}_1 \times \mathcal{I}_2 \not\models \boldsymbol{q}_2$.

Another well-known model-theoretic notion that we need for our characterizations is that of homomorphism. Let $\mathcal{I}_1$ and $\mathcal{I}_2$ be interpretations, and Σ a signature. A function $h \colon \Delta^{\mathcal{I}_2} \to \Delta^{\mathcal{I}_1}$ is a *Σ-homomorphism from $\mathcal{I}_2$ to $\mathcal{I}_1$* if

– $h(a^{\mathcal{I}_2}) = a^{\mathcal{I}_1}$ for all $a \in \mathsf{N_I}$ interpreted by $\mathcal{I}_2$,
– $d \in A^{\mathcal{I}_2}$ implies $h(d) \in A^{\mathcal{I}_1}$ for all $d \in \Delta^{\mathcal{I}_2}$ and Σ-concept names A,
– $(d, e) \in r^{\mathcal{I}_2}$ implies $(h(d), h(e)) \in r^{\mathcal{I}_1}$ for all $d, e \in \Delta^{\mathcal{I}_2}$ and Σ-role names r.

It is readily seen that if $\mathcal{I}_2 \models \boldsymbol{q}(\boldsymbol{a})$, for a Σ-CQ $\boldsymbol{q}(\boldsymbol{x})$, and there is a Σ-homomorphism from $\mathcal{I}_2$ to $\mathcal{I}_1$, then $\mathcal{I}_1 \models \boldsymbol{q}(\boldsymbol{a})$. Furthermore, if we regard $\boldsymbol{q}(\boldsymbol{a})$ as an interpretation whose domain consists of the elements in $\boldsymbol{a}$ (substituted for the answer variables) and of the quantified variables in $\boldsymbol{q}(\boldsymbol{x})$, and whose interpretation function is given by its atoms, then $\mathcal{I}_2 \models \boldsymbol{q}(\boldsymbol{a})$ iff there exists a Σ-homomorphism from $\boldsymbol{q}(\boldsymbol{a})$ to $\mathcal{I}_2$.

To give model-theoretic criteria for CQ entailment and UCQ entailment, we actually start with partial Σ-homomorphisms, which we replace by full homomorphisms in a second step. Let n be a natural number. We say that $\mathcal{I}_2$ is *$n\Sigma$-homomorphically embeddable into $\mathcal{I}_1$* if, for any subinterpretation $\mathcal{I}_2'$ of $\mathcal{I}_2$ with $|\Delta^{\mathcal{I}_2'}| \leq n$, there is a Σ-homomorphism from $\mathcal{I}_2'$ to $\mathcal{I}_1$.[11] If $\mathcal{I}_2$ is $n\Sigma$-homomorphically embeddable into $\mathcal{I}_1$ for any $n > 0$, then we say that $\mathcal{I}_2$ is *finitely Σ-homomorphically embeddable into $\mathcal{I}_1$*.

Theorem 23. *Let $\mathcal{K}_1$ and $\mathcal{K}_2$ be KBs, Σ a signature, and $\boldsymbol{M}_i^{\mathcal{Q}}$ a class of interpretations that is complete for $\mathcal{K}_i$ and the class of queries $\mathcal{Q}$, for $i = 1, 2$ and $\mathcal{Q} \in \{CQ, UCQs\}$. Then*

[11] $\mathcal{I}_2'$ is a *subinterpretation* of $\mathcal{I}_2$ if $\Delta^{\mathcal{I}_2'} \subseteq \Delta^{\mathcal{I}_2}$, $A^{\mathcal{I}_2'} = A^{\mathcal{I}_2} \cap \Delta^{\mathcal{I}_2'}$ and $r^{\mathcal{I}_2'} = r^{\mathcal{I}_2} \cap (\Delta^{\mathcal{I}_2'} \times \Delta^{\mathcal{I}_2'})$, for all concept names A and role names r.

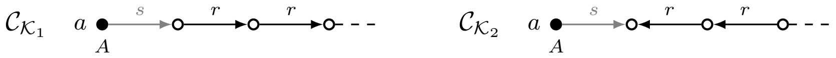
- *(i) $\mathcal{K}_1$ Σ-UCQ entails $\mathcal{K}_2$ iff, for any $n > 0$ and $\mathcal{I}_1 \in \boldsymbol{M}_1^{UCQ}$, there exists $\mathcal{I}_2 \in \boldsymbol{M}_2^{UCQ}$ that is $n\Sigma$-homomorphically embeddable into $\mathcal{I}_1$.*
- *(ii) $\mathcal{K}_1$ Σ-CQ entails $\mathcal{K}_2$ iff $\prod \boldsymbol{M}_2^{CQ}$ is finitely Σ-homomorphically embeddable into $\prod \boldsymbol{M}_1^{CQ}$.*

As finite Σ-homomorphic embeddability is harder to deal with algorithmically than full Σ-homomorphic embeddability, it would be convenient to replace finite Σ-homomorphic embeddability with Σ-homomorphic embeddability in Theorem 23. We first observe that this is not possible in general:

Example 24. Let $\mathcal{K}_i = (\mathcal{T}_i, \mathcal{A})$, $i = 1, 2$, be *DL-Lite$_{core}$* KBs where $\mathcal{A} = \{A(c)\}$, and

$$\mathcal{T}_1 = \{A \sqsubseteq \exists s.\top,\ \exists s^-.\top \sqsubseteq \exists r.\top,\ \exists r^-.\top \sqsubseteq \exists r.\top\},$$
$$\mathcal{T}_2 = \{A \sqsubseteq \exists s.\top,\ \exists s^-.\top \sqsubseteq \exists r^-.\top,\ \exists r.\top \sqsubseteq \exists r^-.\top\}.$$

Let $\Sigma = \{A, r\}$. Recall that the class of models $\{\mathcal{C}_{\mathcal{K}_i}\}$ is complete for $\mathcal{K}_i$ and UCQs, where $\mathcal{C}_{\mathcal{K}_i}$ is the canonical model of $\mathcal{K}_i$:



The KBs $\mathcal{K}_1$ and $\mathcal{K}_2$ are Σ-UCQ inseparable, but $\mathcal{C}_{\mathcal{K}_2}$ is not Σ-homomorphically embeddable into $\mathcal{C}_{\mathcal{K}_1}$.

The example above uses inverse roles and it turns out that these are indeed needed to construct counterexamples against the version of Theorem 23 where finite homomorphic embeddability is replaced with full embeddability. The following result showcases this. It concentrates on *Horn-$\mathcal{ALC}$* and on $\mathcal{ALC}$, which do not admit inverse roles, and establishes characterizations of query entailment based on full homomorphic embeddings.

Theorem 24.

- *(i) Let $\mathcal{K}_1$ and $\mathcal{K}_2$ be Horn-$\mathcal{ALC}$ KBs. Then $\mathcal{K}_1$ Σ-CQ entails $\mathcal{K}_2$ iff $\mathcal{C}_{\mathcal{K}_2}$ is Σ-homomorphically embeddable into $\mathcal{C}_{\mathcal{K}_1}$.*
- *(ii) Let $\mathcal{K}_1$ and $\mathcal{K}_2$ be $\mathcal{ALC}$ KBs. Then $\mathcal{K}_1$ Σ-UCQ entails $\mathcal{K}_2$ iff, for every $\mathcal{I}_1 \in \boldsymbol{M}^b_{tree}(\mathcal{K}_1)$, there exists $\mathcal{I}_2 \in \boldsymbol{M}^b_{tree}(\mathcal{K}_2)$ such that $\mathcal{I}_2$ is Σ-homomorphically embeddable into $\mathcal{I}_1$.*

Claim (i) of Theorem 24 is proved in [14] using a game-theoretic characterization (which we discuss below). The proof of (ii) is given in [16]. One first proves using an automata-theoretic argument that one can work without loss of generality with models in $\boldsymbol{M}^b_{tree}(\mathcal{K}_1)$ in which the tree interpretations $\mathcal{I}_a$ attached to the ABox individuals a are regular. Second, since nodes in $\mathcal{I}_a$ are related to their children using role names only (as opposed to inverse roles), Σ-homomorphisms on tree interpretations correspond to Σ-simulations (see Sections 4.3 and 6.3). Finally, using this observation one can construct the required Σ-homomorphism as the union of finite Σ-homomorphisms on finite initial parts of the tree interpretations $\mathcal{I}_a$.

Note that Theorem 24 omits the case of $\mathcal{ALC}$ KBs and CQ entailment, for which we are not aware of a characterization in terms of full homomorphic embeddability.

Another interesting aspect of Example 24 is that the canonical model of $\mathcal{K}_2$ contains elements that are not reachable along a path of Σ-roles. In fact, just like inverse roles, this is a crucial feature for the example to work. We illustrate this by considering *rooted* UCQs (rUCQs). Recall that in an rUCQ, every variables has to be connected to an answer variable. For answering a Σ-rUCQ, Σ-disconnected parts of models such as in Example 24 can essentially be ignored since the query cannot 'see' them. As a consequence, we can sometimes replace finite homomorphic embeddability with full homomorphic embeddability. We give an example characterization to illustrate this. Call an interpretation $\mathcal{I}$ *Σ-connected* if, for every $u \in \Delta^{\mathcal{I}}$, there is a path $r_1^{\mathcal{I}}(a, u_1), \ldots, r_n^{\mathcal{I}}(u_n, u)$ with an individual a and $r_i \in \Sigma$. An interpretation $\mathcal{I}_2$ is *con-Σ-homomorphically embeddable into* $\mathcal{I}_1$ if the maximal Σ-connected subinterpretation $\mathcal{I}_2'$ of $\mathcal{I}_2$ is Σ-homomorphically embeddable into $\mathcal{I}_1$.

Theorem 25. *Let $\mathcal{K}_1$ and $\mathcal{K}_2$ be $\mathcal{ALCHI}$ KBs and Σ a signature. Then $\mathcal{K}_1$ Σ-rUCQ entails $\mathcal{K}_2$ iff for any $\mathcal{I}_1 \in \boldsymbol{M}^b_{utree}(\mathcal{K}_1)$, there exists $\mathcal{I}_2 \in \boldsymbol{M}^b_{utree}(\mathcal{K}_2)$ that is con-Σ-homomorphically embeddable into $\mathcal{I}_1$.*

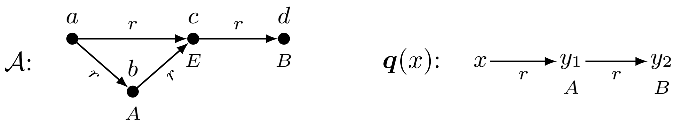
Theorem 25 is proved for $\mathcal{ALC}$ in [15]. The extension to $\mathcal{ALCHI}$ is straightforward. The model-theoretic criteria given above are a good starting point for designing decision procedures for query inseparability. But can they be checked effectively? We first consider this question for $\mathcal{ALC}$ and then move to Horn DLs.

6.2 Query inseparability of $\mathcal{ALC}$ KBs

We begin with CQ entailment and inseparability in $\mathcal{ALC}$ and show that both problems are undecidable even for very restricted classes of KBs. The same is true for rCQs. We then show that, in contrast to the CQ case, UCQ inseparability in $\mathcal{ALC}$ is decidable in 2ExpTime.

The following example illustrates the notion of CQ-inseparability of $\mathcal{ALC}$ KBs.

Example 25. Suppose $\mathcal{T}_1 = \emptyset$, $\mathcal{T}_2 = \{E \sqsubseteq A \sqcup B\}$, $\mathcal{A}$ looks like on the left-hand side of the picture below, and $\Sigma = \{r, A, B\}$. Then we can separate $\mathcal{K}_2 = (\mathcal{T}_2, \mathcal{A})$ from $\mathcal{K}_1 = (\mathcal{T}_1, \mathcal{A})$ by the Σ-CQ $\boldsymbol{q}(x)$ shown on the right-hand side of the picture since clearly $(\mathcal{T}_1, \mathcal{A}) \not\models \boldsymbol{q}(a)$, whereas $(\mathcal{T}_2, \mathcal{A}) \models \boldsymbol{q}(a)$. To see the latter, we first observe that, in any model $\mathcal{I}$ of $\mathcal{K}_2$, we have (i) $c \in A^{\mathcal{I}}$ or (ii) $c \in B^{\mathcal{I}}$. In case (i), $\mathcal{I} \models \boldsymbol{q}(a)$ because of the path $r(a, c), r(c, d)$; and if (ii) holds, then $\mathcal{I} \models \boldsymbol{q}(a)$ because of the path $r(a, b), r(b, c)$ (cf. [94, Example 4.2.5]).

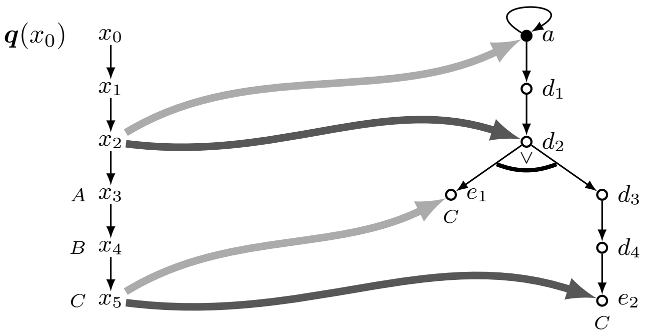


Theorem 26. *Let $\mathcal{Q} \in \{CQ, rCQ\}$.*

(i) Σ-$\mathcal{Q}$ entailment of an $\mathcal{ALC}$ KB by an $\mathcal{EL}$ KB is undecidable.
(ii) Σ-$\mathcal{Q}$ inseparability of an $\mathcal{ALC}$ and an $\mathcal{EL}$ KBs is undecidable.

The proof of this theorem given in [15] uses a reduction of an undecidable tiling problem. As usual in encodings of tilings, it is not hard to synchronize tile colours along one dimension. The following example gives a hint of how this can be achieved in the second dimension.

Example 26. Suppose a KB $\mathcal{K}$ has the two models $\mathcal{I}_i$, $i = 1, 2$, that are formed by the points on the path between a and e_i on the right-hand side of the picture below (this can be easily achieved using an inclusion of the form $D \sqsubseteq D_1 \sqcup D_2$), with a being an ABox point with a loop and the e_i being the only instances of a concept C. Let $\boldsymbol{q}$ be the CQ on the left-hand side of the picture. Then we can have $\mathcal{K} \models \boldsymbol{q}(a)$ only if $d_1, d_3 \in A^{\mathcal{I}_2}$ and $d_2, d_4 \in B^{\mathcal{I}_2}$, with the fat black and grey arrows indicating homomorphisms from $\boldsymbol{q}$ to the $\mathcal{I}_i$ (the grey one sends x_0–x_2 to a using the ABox loop). This trick can be used to pass the tile colours from one row to another.



As we saw in Example 20, UCQs distinguish between more KBs than CQs, that is, UCQ inseparability is a different and in fact more fine-grained notion than CQ inseparability. This has the remarkable effect that decidability is regained [16].

Theorem 27. *In $\mathcal{ALC}$, Σ-$\mathcal{Q}$ entailment and Σ-$\mathcal{Q}$ inseparability of KBs are 2*ExpTime*-complete, for $\mathcal{Q} \in \{UCQ, rUCQ\}$.*

The proof of the upper bound in Theorem 27 uses tree automata and relies on the characterization of Theorem 24 (ii) [16]. In principle, the automata construction is similar to the one given in the proof sketch of Theorem 4. The main differences between the two constructions is that we have to replace bisimulations with homomorphisms. Since homomorphisms preserve only positive and existential information, we can actually drop some parts of the automaton construction. On the other hand, homomorphisms require us to consider also parts of the model $\mathcal{I}_2$ (see Theorem 24) that are not reachable along Σ-roles from an

ABox individual, which requires a bit of extra effort. A more technical issue is that the presence of an ABox seems to not go together so well with amorphous automata and thus one resorts to more traditional tree automata along with a suitable encoding of the ABox and of the model $\mathcal{I}_1$ as a labeled tree. The lower bound is proved by an ATM reduction.

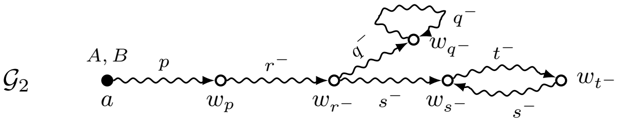
6.3 Query inseparability of KBs in Horn DLs

We first consider DLs without inverse roles and then DLs that admit inverse roles. In both cases, we sketch decision procedures that are based on games played on canonical models $C_{\mathcal{K}}$ as mentioned in Example 22. It is well known from logic programming and databases [1] that such models can be constructed by the *chase procedure*. We illustrate the chase (in a somewhat different but equivalent form) by the following example.

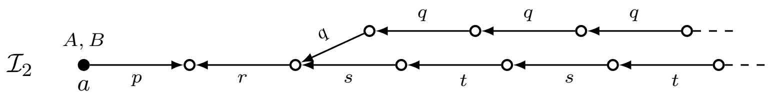
Example 27. Consider the *DL-Lite*$^{\mathcal{H}}_{core}$ KB $\mathcal{K}_2 = (\mathcal{T}_2, \mathcal{A}_2)$ with $\mathcal{A}_2 = \{A(a)\}$ and

$$\begin{aligned}\mathcal{T}_2 = \{A \sqsubseteq B,\ A \sqsubseteq \exists p.\top,\ \exists p^-.\top \sqsubseteq \exists r^-.\top,\ \exists r.\top \sqsubseteq \exists q^-.\top,\ \exists q.\top \sqsubseteq \exists q^-.\top,\\ \exists r.\top \sqsubseteq \exists s^-.\top,\ \exists s.\top \sqsubseteq \exists t^-.\top,\ \exists t.\top \sqsubseteq \exists s^-.\top,\ t^- \sqcap s \sqsubseteq \bot\}.\end{aligned}$$

We first construct a 'closure' of the ABox $\mathcal{A}_2$ under the inclusions in $\mathcal{T}_2$. For instance, to satisfy $A \sqsubseteq \exists p.\top$, we introduce a *witness* w_p *for* p and draw an arrow $\rightsquigarrow$ from a to w_p indicating that $p(a, w_p)$ holds. The inclusion $\exists p^-.\top \sqsubseteq \exists r^-.\top$ requires a witness w_{r^-} for r^- and the arrow $w_p \rightsquigarrow w_{r^-}$. Having reached the witness w_{q^-} for q^- and applying $\exists q.\top \sqsubseteq \exists q^-.\top$ to it, we 'reuse' w_{q^-} and simply draw a loop $w_{q^-} \rightsquigarrow w_{q^-}$. The resulting finite interpretation $\mathcal{G}_2$ shown below is called the *generating structure for* $\mathcal{K}_2$:



Note that $\mathcal{G}_2$ is *not* a model of $\mathcal{K}_2$ because $(w_{s^-}, w_{t^-}) \in (t^-)^{\mathcal{G}_2} \cap s^{\mathcal{G}_2}$. We can obtain a model of $\mathcal{K}_2$ by *unravelling* the witness part of the generating structure $\mathcal{G}_2$ into an infinite tree (in general, forest). The resulting interpretation $\mathcal{I}_2$ shown below is a canonical model of $\mathcal{K}_2$.



The generating structure underlying the canonical model $\mathcal{C}_{\mathcal{K}}$ of a Horn KB $\mathcal{K}$ defined above will be denoted by $\mathcal{G}_{\mathcal{K}}$. By Theorem 24, if $\mathcal{K}_1$ and $\mathcal{K}_2$ are KBs formulated in a Horn DL, then $\mathcal{K}_1$ Σ-CQ entails $\mathcal{K}_2$ iff $\mathcal{C}_{\mathcal{K}_2}$ is $n\Sigma$-homomorphically embeddable into $\mathcal{C}_{\mathcal{K}_2}$ for any $n > 0$.

In what follows, we require the following upper bounds on the size of generating structures for Horn KBs [14]:

Theorem 28.

(*i*) *The generating structure for any consistent* Horn-$\mathcal{ALCHI}$ *KB* $(\mathcal{T}, \mathcal{A})$ *can be constructed in time* $|\mathcal{A}| \cdot 2^{p(|\mathcal{T}|)}$, *where* p *is some fixed polynomial*;

(*ii*) *The generating structure for any consistent KB* $(\mathcal{T}, \mathcal{A})$ *formulated in a DL from the* $\mathcal{EL}$ *or DL-Lite family can be constructed in time* $|\mathcal{A}| \cdot p(|\mathcal{T}|)$, *where* p *is some fixed polynomial.*

We now show that checking whether a canonical model is $n\Sigma$-homomorphically embeddable into another canonical model can be established by playing games on their underlying generating structures. For more details, the reader is referred to [14].

Suppose $\mathcal{K}_1$ and $\mathcal{K}_2$ are (consistent) Horn KBs, $\mathcal{C}_1$ and $\mathcal{C}_2$ are their canonical models, and Σ a signature. First, we reformulate the definition of $n\Sigma$-homomorphic embedding in game-theoretic terms. The states of our game are of the form $(\pi \mapsto \sigma)$, where $\pi \in \Delta^{\mathcal{C}_2}$ and $\sigma \in \Delta^{\mathcal{C}_1}$. Intuitively, $(\pi \mapsto \sigma)$ means that 'π is to be Σ-homomorphically mapped to σ'. The game is played by player 1 and player 2 starting from some initial state $(\pi_0 \mapsto \sigma_0)$. The aim of player 1 is to demonstrate that there exists a Σ-homomorphism from (a finite subinterpretation of) $\mathcal{C}_2$ into $\mathcal{C}_1$ with π_0 mapped to σ_0, while player 2 wants to show that there is no such homomorphism. In each round $i > 0$ of the game, player 2 challenges player 1 with some $\pi_i \in \Delta^{\mathcal{C}_2}$ that is related to π_{i-1} by some Σ-role. Player 1, in turn, has to respond with some $\sigma_i \in \Delta^{\mathcal{C}_1}$ such that the already constructed partial Σ-homomorphism can be extended with $(\pi_i \mapsto \sigma_i)$, in particular:

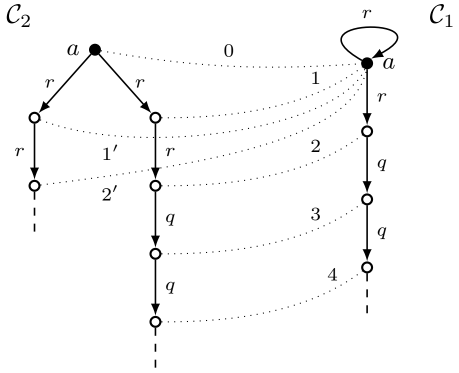
- $\pi_i \in A^{\mathcal{C}_2}$ implies $\sigma_i \in A^{\mathcal{C}_1}$, for any Σ-concept name A, and
- $(\pi_{i-1}, \pi_i) \in r^{\mathcal{C}_2}$ implies $(\sigma_{i-1}, \sigma_i) \in r^{\mathcal{C}_1}$, for any Σ-role r.

A play of length n starting from a state $\mathfrak{s}_0$ is any sequence $\mathfrak{s}_0, \dots, \mathfrak{s}_n$ of states obtained as described above. For any ordinal $\lambda \leq \omega$, we say that player 1 has a λ-*winning strategy* in the game starting from $\mathfrak{s}_0$ if, for any play $\mathfrak{s}_0, \dots, \mathfrak{s}_n$ with $n < \lambda$ that is played according to this strategy, player 1 has a response to any challenge of player 2 in the final state $\mathfrak{s}_n$.

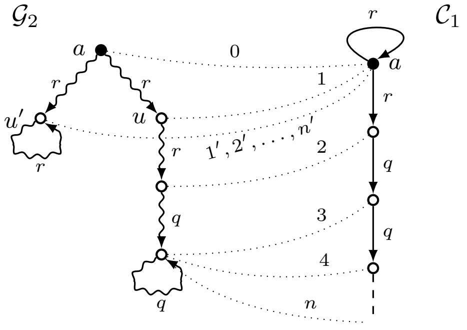
It is easy to see that if, for any $\pi_0 \in \Delta^{\mathcal{C}_2}$, there is $\sigma_0 \in \Delta^{\mathcal{C}_1}$ such that player 1 has an ω-winning strategy in this game starting from $(\pi_0 \to \sigma_0)$, then there is a Σ-homomorphism from $\mathcal{C}_2$ into $\mathcal{C}_1$, and the other way round. That $\mathcal{C}_2$ is *finitely* Σ-homomorphically embeddable into $\mathcal{C}_1$ is equivalent to the following condition:

- for any $\pi_0 \in \Delta^{\mathcal{C}_2}$ and any $n < \omega$, there exists $\sigma_0 \in \Delta^{\mathcal{C}_1}$ such that player 1 has an n-winning strategy in this game starting from $(\pi_0 \to \sigma_0)$.

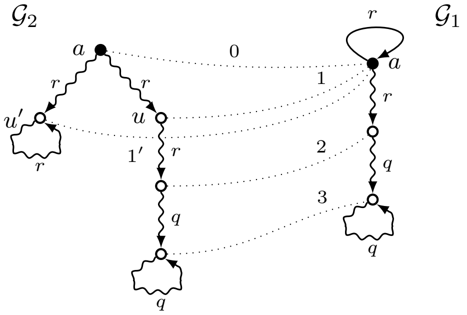
Example 28. Suppose $\mathcal{C}_1$ and $\mathcal{C}_2$ look like in the picture below. An ω-winning strategy for player 1 starting from $(a \mapsto a)$ is shown by the dotted lines with the rounds of the game indicated by the numbers on the lines.



Note, however, that the game-theoretic criterion formulated above does not immediately yield any algorithm to decide finite homomorphic embeddability because both $\mathcal{C}_2$ and $\mathcal{C}_1$ can be infinite. It is readily seen that the canonical model $\mathcal{C}_2$ in the game can be replaced by the underlying generating structure $\mathcal{G}_2$, in which player 2 can only make challenges indicated by the generating relation $\rightsquigarrow$. The picture below illustrates the game played on $\mathcal{G}_2$ and $\mathcal{C}_1$ from Example 28.



If the KBs are formulated in a Horn DL that does not allow *inverse roles*, then $\mathcal{C}_1$ can also be replaced with its generating structure $\mathcal{G}_1$ as illustrated by the picture below:



Reachability or simulation games on finite graphs such as the one discussed above have been extensively investigated in game theory [77, 22]. In particular, it follows that checking the existence of n-winning strategies, for all $n < \omega$, can be done in polynomial time in the number of states and the number of the available challenges. Together with Theorem 28, this gives the upper bounds in the following theorem. Claim (i) was first observed in [71], while (ii) and the results on data complexity are from [14].

Theorem 29. *Σ-CQ entailment and Σ-CQ inseparability of KBs are*

(i) *in* PTime *for $\mathcal{EL}$;*
(ii) ExpTime*-complete for Horn-$\mathcal{ALC}$.*

Both problems are in PTime *for data complexity for both $\mathcal{EL}$ and Horn-$\mathcal{ALC}$.*

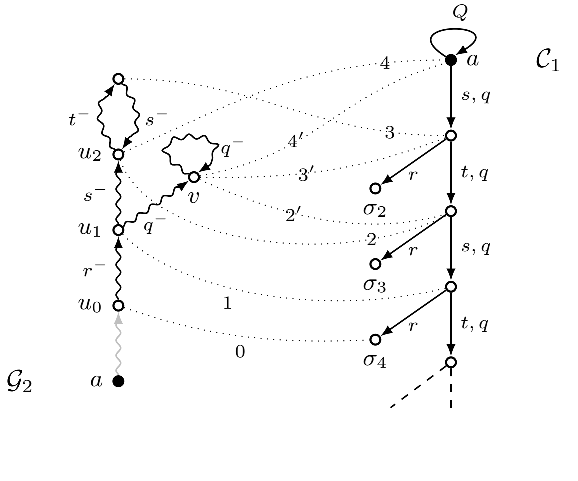
Here, by 'data complexity' we mean that only the ABoxes of the two involved KBs are regarded as input, while the TBoxes are fixed. Analogously to data complexity in query answering, the rationale behind this setup is that, in data-centric applications, ABoxes tend to be huge compared to the TBoxes and thus it can result in more realistic complexities to assume that the latter are actually of constant size. The lower bound in Theorem 29 is proved by reduction of the word problem of polynomially space-bounded ATMs. We remind the reader at this point that, in all DLs studied in this section, CQ entailment coincides with UCQ entailment, and likewise for inseparability.

If inverse roles are available, then replacing canonical models with their generating structures in games often becomes less straightforward. We explain the issues using an example in *DL-Lite*$^{\mathcal{H}}_{core}$, where inverse roles interact in a problematic way with role inclusions. Similar effects can be observed in Horn-$\mathcal{ALCI}$, though, despite the fact that no role inclusions are available there.

Example 29. Consider the *DL-Lite*$^{\mathcal{H}}_{core}$ KBs $\mathcal{K}_1 = (\mathcal{T}_1, \{Q(a, a)\})$ with

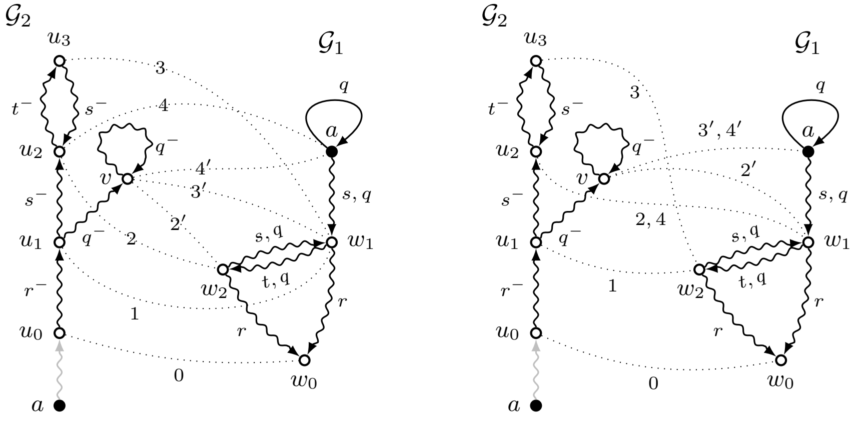
$$\mathcal{T}_1 = \{\ A \sqsubseteq \exists s.\top,\ \exists s^-.\top \sqsubseteq \exists t.\top,\ \exists t^-.\top \sqsubseteq \exists s.\top,\ s \sqsubseteq q,\ t \sqsubseteq q,\ \exists q^-.\top \sqsubseteq \exists r.\top\ \}$$

and $\mathcal{K}_2$ from Example 27. Let $\Sigma = \{q, r, s, t\}$. The generating structure $\mathcal{G}_2$ for $\mathcal{K}_2$ and the canonical model $\mathcal{C}_1$ for $\mathcal{K}_1$, as well as a 4-winning strategy for player 1 in the game over $\mathcal{G}_2$ and $\mathcal{C}_1$ starting from the state (u_0, σ_4) are shown in the picture below:



(In fact, for any $n > 0$, player 1 has an n-winning strategy starting from any $(u_0 \mapsto \sigma_m)$ provided that m is even and $m \geq n$.)

This game over $\mathcal{G}_2$ and $\mathcal{C}_1$ has its obvious counterparts over $\mathcal{G}_2$ and $\mathcal{G}_1$; one of them is shown on the left-hand side of the picture below. It is to be noted, however, that—unlike Example 28—the responses of player 1 are in the reverse direction of the $\rightsquigarrow$-arrows (which is possible because of the inverse roles).



On the other hand, such reverse responses may create paths in $\mathcal{G}_1$ that do not have any real counterparts in $\mathcal{C}_1$, and so do not give rise to Σ-homomorphisms we need. An example is shown on the right-hand side of the picture above, where u_3 is mapped to w_2 and v to a in round 3, which is impossible to reproduce in the *tree-shaped* $\mathcal{C}_1$.

One way to ensure that, in the 'backwards game' over $\mathcal{G}_2$ and $\mathcal{G}_1$, all the challenges made by player 1 in any given state are responded by the same element of $\mathcal{G}_1$, is to use states of the form $(\Xi \mapsto w)$, where Ξ is the *set* of elements of $\mathcal{G}_2$ to be mapped to an element w of $\mathcal{G}_1$. In our example above, we can use the state $(\{u_2, v\} \mapsto w_2)$, where the only challenge of player 2 is the set of $\rightsquigarrow$- successors of u_2 and v marked by Σ-roles, that is, $\Xi' = \{u_3, v\}$, to which player 1 responds with $(\Xi' \mapsto w_1)$.

By allowing more complex states, we increase their number and, as a consequence, the complexity of deciding finite Σ-homomorphic embeddability. The proof of the following theorem can be found in [14]:

Theorem 30. *Σ-CQ entailment and inseparability of KBs are*

(i) ExpTime*-complete for DL-Lite$^{\mathcal{H}}_{horn}$ and DL-Lite$^{\mathcal{H}}_{core}$;*
(ii) 2ExpTime*-complete for Horn-$\mathcal{ALCI}$ and Horn-$\mathcal{ALCHI}$.*

For all of these DLs, both problems are in PTime *for data complexity.*

The lower bounds are once again proved using alternating Turing machines. We remark that CQ entailment and inseparability are in PTime in *DL-Lite$_{core}$* and *DL-Lite$_{horn}$*. In *DL-Lite*, it is thus the combination of inverse roles and role hierarchies that causes hardness.

7 Query Inseparability of TBoxes

Query inseparability of KBs, as studied in the previous section, presupposes that the data (in the form of an ABox) is known to the user, as is the case for example in KB exchange [2]. In many query answering applications, though, the data is either not known during the TBox design or it changes so frequently that query inseparability w.r.t. one fixed data set is not a sufficiently robust notion. In such cases, one wants to decide query inseparability of *TBoxes* $\mathcal{T}_1$ and $\mathcal{T}_2$, defined by requiring that, for *all* ABoxes $\mathcal{A}$, the KBs $(\mathcal{T}_1, \mathcal{A})$ and $(\mathcal{T}_2, \mathcal{A})$ are query inseparable. To increase flexibility, we can also specify a signature of the ABoxes that we are considering, and we do not require that it coincides with the signature of the queries. In a sense, this change in the setup brings us closer to the material from Sections 4 and 5, where also inseparability of TBoxes is studied. As in the KB case, the main classes of queries that we consider are CQs and UCQs as well as rCQs and rUCQs. To relate concept inseparability and query inseparability of TBoxes, we additionally consider a class of tree-shaped CQs.

The structure of this section is as follows. We start by discussing the impact that the choice of query language has on query inseparability of TBoxes. We then relate query inseparability of TBoxes to logical equivalence, concept inseparability, and model inseparability. For Horn DLs, query inseparability and concept inseparability are very closely related, while this is not the case for DLs with disjunction. Finally, we consider the decidability and complexity of query inseparability of TBoxes. Undecidability of CQ inseparability of $\mathcal{ALC}$ KBs transfers to the TBox case, and the same is true of upper complexity bounds in *DL-Lite*. New techniques are needed to establish decidability and complexity results for other Horn DLs such as $\mathcal{EL}$ and Horn-$\mathcal{ALC}$. A main observation underlying our algorithms is that it is sufficient to consider tree-shaped ABoxes when searching for witnesses of query separability of TBoxes.

Definition 13 (query inseparability, entailment and conservative extensions). Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be TBoxes, $\Theta = (\Sigma_1, \Sigma_2)$ a pair of signatures, and $\mathcal{Q}$ a class of queries. Then

- the Θ-$\mathcal{Q}$ *difference* between $\mathcal{T}_1$ and $\mathcal{T}_2$ is the set $\mathsf{qDiff}^{\mathcal{Q}}_{\Theta}(\mathcal{T}_1, \mathcal{T}_2)$ of all pairs $(\mathcal{A}, \boldsymbol{q}(\boldsymbol{a}))$ such that $\mathcal{A}$ is a Σ_1-ABox and $\boldsymbol{q}(\boldsymbol{a}) \in \mathsf{qDiff}^{\mathcal{Q}}_{\Sigma_2}(\mathcal{K}_1, \mathcal{K}_2)$, where $\mathcal{K}_i = (\mathcal{T}_i, \mathcal{A})$ for $i = 1, 2$;
- $\mathcal{T}_1$ Θ-$\mathcal{Q}$ *entails* $\mathcal{T}_2$ if $\mathsf{qDiff}^{\mathcal{Q}}_{\Theta}(\mathcal{T}_1, \mathcal{T}_2) = \emptyset$;
- $\mathcal{T}_1$ and $\mathcal{T}_2$ are Θ-$\mathcal{Q}$ *inseparable* if $\mathcal{T}_1$ Θ-$\mathcal{Q}$ entails $\mathcal{T}_2$ and vice versa;
- $\mathcal{T}_2$ is a $\mathcal{Q}$ *conservative extension of* $\mathcal{T}_1$ iff $\mathcal{T}_2 \supseteq \mathcal{T}_1$ and $\mathcal{T}_1$ and $\mathcal{T}_2$ are Θ-$\mathcal{Q}$ inseparable for $\Sigma_1 = \Sigma_2 = \mathsf{sig}(\mathcal{T}_1)$.

If $(\mathcal{A}, \boldsymbol{q}(\boldsymbol{a})) \in \mathsf{qDiff}^{\mathcal{Q}}_{\Theta}(\mathcal{T}_1, \mathcal{T}_2)$, we say that $(\mathcal{A}, \boldsymbol{q}(\boldsymbol{a}))$ Θ-$\mathcal{Q}$ *separates* $\mathcal{T}_1$ and $\mathcal{T}_2$.

Note that Definition 13 does not require the separating ABoxes to be satisfiable with $\mathcal{T}_1$ and $\mathcal{T}_2$. Thus, a potential source of separability is that there is an ABox with which one of the TBoxes is satisfiable while the other is not.

One could also define a (more brave) version of query inseparability where only ABoxes are considered that are satisfiable with $\mathcal{T}_1$ and $\mathcal{T}_2$. We will discuss this further in Section 7.2.

We now analyze the impact that the choice of query language has on query inseparability. To this end, we introduce a class of tree-shaped CQs that is closely related to $\mathcal{EL}$-concepts. Every $\mathcal{EL}$-concept C corresponds to a tree-shaped rCQ $\boldsymbol{q}_C(x)$ such that, for any interpretation $\mathcal{I}$ and $d \in \Delta^{\mathcal{I}}$, we have $d \in C^{\mathcal{I}}$ iff $\mathcal{I} \models \boldsymbol{q}_C(d)$. We denote $\boldsymbol{q}_C(x)$ by $C(x)$ and the Boolean CQ $\exists x\, \boldsymbol{q}_C(x)$ by $\exists x\, C(x)$. We use $\mathcal{Q}_{\mathcal{EL}}$ to denote the class of all queries of the former kind and $\mathcal{Q}_{\mathcal{EL}^u}$ for the class of queries of any of these two kinds. In the following theorem, the equivalence of (*iii*) with the other two conditions is of particular interest. Informally it says that, in $\mathcal{EL}$ and Horn-$\mathcal{ALC}$, tree-shaped queries are always sufficient to separate TBoxes.

Theorem 31. *Let $\mathcal{L}$ be a Horn DL, $\mathcal{T}_1$ and $\mathcal{T}_2$ TBoxes formulated in $\mathcal{L}$, and $\Theta = (\Sigma_1, \Sigma_2)$ a pair of signatures. Then the following conditions are equivalent:*

(*i*) *$\mathcal{T}_1$ Θ-UCQ entails $\mathcal{T}_2$;*
(*ii*) *$\mathcal{T}_1$ Θ-CQ entails $\mathcal{T}_2$.*

If $\mathcal{L}$ is $\mathcal{EL}$ or Horn-$\mathcal{ALC}$, then these conditions are also equivalent to

(*iii*) *$\mathcal{T}_1$ Θ-$\mathcal{Q}_{\mathcal{EL}^u}$ entails $\mathcal{T}_2$.*

The same is true when UCQs are replaced with rUCQs, CQs with rCQs, and $\mathcal{Q}_{\mathcal{EL}^u}$ with $\mathcal{Q}_{\mathcal{EL}}$ (simultaneously).

Proof. The first equivalence follows directly from the fact that KBs in Horn DLs are complete w.r.t. a single model (Example 22). We sketch the proof that Θ-$\mathcal{Q}_{\mathcal{EL}^u}$ entailment implies Θ-CQ entailment in $\mathcal{EL}$ and Horn-$\mathcal{ALC}$. Assume that there is a Σ_1-ABox $\mathcal{A}$ such that $\mathcal{K}_1$ does not Σ_2-CQ entail $\mathcal{K}_2$ for $\mathcal{K}_1 = (\mathcal{T}_1, \mathcal{A})$ and $\mathcal{K}_2 = (\mathcal{T}_2, \mathcal{A})$. Then $\mathcal{C}_{\mathcal{K}_2}$ is not finitely Σ-homomorphically embeddable into $\mathcal{C}_{\mathcal{K}_1}$ (see Theorem 23). We thus find a finite subinterpretation $\mathcal{I}$ of an interpretation $\mathcal{I}_a$ in $\mathcal{C}_{\mathcal{K}_2}$ (see Example 22) that is not Σ-homomorphically embeddable into $\mathcal{C}_{\mathcal{K}_1}$. We can regard the Σ-reduct of $\mathcal{I}$ as a Σ-query in $\mathcal{Q}_{\mathcal{EL}^u}$ which takes the form $C(x)$ if $\mathcal{I}$ contains a and $\exists x C(x)$ otherwise. This query witnesses that $\mathcal{K}_1$ does not Θ-$\mathcal{Q}_{\mathcal{EL}^u}$ entail $\mathcal{K}_2$, as required. □

For Horn DLs other than $\mathcal{EL}$ and Horn-$\mathcal{ALC}$, the equivalence between (ii) and (iii) in Theorem 31 does often not hold in exactly the stated form. The reason is that additional constructors such as inverse roles and role inclusions require us to work with slightly different types of canonical models; see Example 22. However, the equivalence then holds for appropriate extensions of $\mathcal{Q}_{\mathcal{EL}^u}$, for example, by replacing $\mathcal{EL}$ concepts with $\mathcal{ELI}$ concepts when moving from Horn-$\mathcal{ALC}$ to Horn-$\mathcal{ALCI}$.

Theorem 31 does not hold for non-Horn DLs. We have already observed in Example 20 that UCQ entailment and CQ entailment of KBs do not coincide in $\mathcal{ALC}$. Since the example uses the same ABox in both KBs, it also applies to the inseparability of TBoxes. In the following example, we prove that the equivalence between CQ inseparability and $\mathcal{Q}_{\mathcal{EL}^u}$ inseparability fails in $\mathcal{ALC}$, too. The proof can actually be strengthened to show that, in $\mathcal{ALC}$, CQ inseparability is a stronger notion than inseparability by *acyclic* CQs (which generalize $\mathcal{Q}_{\mathcal{EL}^u}$ by allowing multiple answer variables and edges in trees that are directed upwards).

Example 30. Let $\Sigma_1 = \{r\}$, $\Sigma_2 = \{r, A\}$, and $\Theta = (\Sigma_1, \Sigma_2)$. We construct an $\mathcal{ALC}$ TBox $\mathcal{T}_1$ as follows:

- to ensure that, for any Σ_1-ABox $\mathcal{A}$, the KB $(\mathcal{T}_1, \mathcal{A})$ is satisfiable iff $\mathcal{A}$ (viewed as an undirected graph with edges $\{\{a, b\} \mid r(a, b) \in \mathcal{A}\}$) is two-colorable, we take the CIs
$$B \sqsubseteq \forall r.\neg B, \quad \neg B \sqsubseteq \forall r.B;$$
- to ensure that, for any Σ_1-ABox $\mathcal{A}$, any model in $\boldsymbol{M}^b_{tree}(\mathcal{T}_1, \mathcal{A})$ has an infinite r-chain of nodes labeled with the concept name A whose root is not reachable from an ABox individual along Σ_2-roles we add the CIs
$$\exists r.\top \sqsubseteq \exists s.B', \quad B' \sqsubseteq A \sqcap \exists r.B'.$$

$\mathcal{T}_2$ is the extension of $\mathcal{T}_1$ with the CI

$$\exists r.\top \sqsubseteq A \sqcup \forall r.A.$$

Thus, models of $(\mathcal{T}_2, A)$ extend models of $(\mathcal{T}_1, \mathcal{A})$ by labeling certain individuals in $\mathcal{A}$ with A. The non-$\mathcal{A}$ part is not modified as we can assume that its elements are already labeled with A. Observe that $\mathcal{T}_1$ and $\mathcal{T}_2$ can be distinguished by the ABox $\mathcal{A} = \{r(a, b), r(b, a)\}$ and the CQ $\boldsymbol{q} = \exists x, y\, (A(x) \wedge r(x, y) \wedge r(y, x))$. Indeed, $a \in A^{\mathcal{I}}$ or $b \in A^{\mathcal{I}}$ holds in every model $\mathcal{I}$ of $(\mathcal{T}_2, \mathcal{A})$ but this is not the case for $(\mathcal{T}_1, \mathcal{A})$. We now argue that, for every Σ_1 ABox $\mathcal{A}$ and every Σ_2 concept C in $\mathcal{EL}$, $(\mathcal{T}_2, \mathcal{A}) \models \exists x\, C(x)$ implies $(\mathcal{T}_1, \mathcal{A}) \models \exists x\, C(x)$ and $(\mathcal{T}_2, \mathcal{A}) \models C(a)$ implies $(\mathcal{T}_1, \mathcal{A}) \models C(a)$. Assume that $\mathcal{A}$ and C are given. As any model in $\boldsymbol{M}^b_{tree}(\mathcal{T}_1, \mathcal{A})$ has infinite r-chains labeled with A, we have $(\mathcal{T}_1, \mathcal{A}) \models \exists x\, C(x)$ for any Σ_2-concept C in $\mathcal{EL}$. Thus, we only have to consider the case $(\mathcal{T}_2, \mathcal{A}) \models C(a)$. If C does not contain A, then clearly $\mathcal{A} \models C(a)$, and so $(\mathcal{T}_1, \mathcal{A}) \models C(a)$, as required. If $\mathcal{A}$ is not 2-colorable, we also have $(\mathcal{T}_1, \mathcal{A}) \models C(a)$, as required. Otherwise C contains A and $\mathcal{A}$ is 2-colorable. But then it is easy to see that $(\mathcal{T}_2, \mathcal{A}) \not\models C(a)$ and we have derived a contradiction.

7.1 Relation to other notions of inseparability

We now consider the relationship between query inseparability, model inseparability, and logical equivalence. Clearly, Σ-model inseparability entails Θ-$\mathcal{Q}$ inseparability for $\Theta = (\Sigma, \Sigma)$ and any class $\mathcal{Q}$ of queries. The same is true for logical equivalence, where we can even choose Θ freely. The converse direction is more interesting.

An ABox $\mathcal{A}$ is said to be *tree-shaped* if the directed graph $(\mathsf{ind}(\mathcal{A}), \{(a, b) \mid r(a, b) \in \mathcal{A}\})$ is a tree and $r(a, b) \in \mathcal{A}$ implies $s(a, b) \notin \mathcal{A}$ for any $a, b \in \mathsf{ind}(\mathcal{A})$

and $s \neq r$. We call $\mathcal{A}$ *undirected tree-shaped* (or *utree-shaped*) if the undirected graph $(\mathsf{ind}(\mathcal{A}), \{\{a,b\} \mid r(a,b) \in \mathcal{A}\})$ is a tree and $r(a,b) \in \mathcal{A}$ implies $s(a,b) \notin \mathcal{A}$ for any $a, b \in \mathsf{ind}(\mathcal{A})$ and $s \neq r$. Observe that every $\mathcal{EL}$ concept C corresponds to a tree-shaped ABox $\mathcal{A}_C$ and, conversely, every tree-shaped ABox $\mathcal{A}$ corresponds to an $\mathcal{EL}$-concept $C_{\mathcal{A}}$. In particular, for any TBox $\mathcal{T}$ and $\mathcal{EL}$ concept D, we have $\mathcal{T} \models C \sqsubseteq D$ iff $(\mathcal{T}, \mathcal{A}_C) \models D(\rho_C)$, ρ_C the root of $\mathcal{A}_C$.

Theorem 32. *Let $\mathcal{L} \in \{\textit{DL-Lite}^{\mathcal{H}}_{core}, \mathcal{EL}\}$ and let $\Theta = (\Sigma_1, \Sigma_2)$ be a pair of signatures such that $\Sigma_i \supseteq \mathsf{sig}(\mathcal{T}_1) \cup \mathsf{sig}(\mathcal{T}_2)$ for $i \in \{1, 2\}$. Then the following conditions are equivalent:*

(i) $\mathcal{T}_1$ and $\mathcal{T}_2$ are logically equivalent;
(ii) $\mathcal{T}_1$ and $\mathcal{T}_2$ are Θ-rCQ inseparable.

Proof. We show $(ii) \Rightarrow (i)$ for $\mathcal{EL}$, the proof for $\textit{DL-Lite}^{\mathcal{H}}_{core}$ is similar and omitted. Assume $\mathcal{T}_1$ and $\mathcal{T}_2$ are $\mathcal{EL}$ TBoxes that are not logically equivalent. Then there is $C \sqsubseteq D \in \mathcal{T}_2$ such that $\mathcal{T}_1 \not\models C \sqsubseteq D$ (or vice versa). We regard C as the tree-shaped Σ_1-ABox $\mathcal{A}_C$ with root ρ_C and D as the Σ_2-rCQ $D(x)$. Then $(\mathcal{T}_2, \mathcal{A}_C) \models D(\rho_C)$ but $(\mathcal{T}_1, \mathcal{A}_C) \not\models D(\rho_C)$. Thus $\mathcal{T}_1$ and $\mathcal{T}_2$ are Θ-rCQ separable. □

Of course, Theorem 32 fails when the restriction of Θ is dropped. The following example shows that, even with this restriction, Theorem 32 does not hold for Horn-$\mathcal{ALC}$.

Example 31. Consider the Horn-$\mathcal{ALC}$ TBoxes

$$\mathcal{T}_1 = \{A \sqsubseteq \exists r.\neg A\} \quad \text{and} \quad \mathcal{T}_2 = \{A \sqsubseteq \exists r.\top\}.$$

Clearly, $\mathcal{T}_1$ and $\mathcal{T}_2$ are not logically equivalent. However, it is easy to see that they are Θ-UCQ inseparable for any Θ.

We now relate query inseparability and concept inseparability. In $\mathcal{ALC}$, these notions are incomparable. It already follows from Example 31 that UCQ inseparability does not imply concept inseparability. The following example shows that the converse implication does not hold either.

Example 32. Consider the $\mathcal{ALC}$ TBoxes $\mathcal{T}_1 = \emptyset$ and

$$\mathcal{T}_2 = \{B \sqcap \exists r.B \sqsubseteq A, \neg B \sqcap \exists r.\neg B \sqsubseteq A\}.$$

Using Theorem 3, one can show that $\mathcal{T}_1$ and $\mathcal{T}_2$ are Σ-concept inseparable, for $\Sigma = \{A, r\}$. However, $\mathcal{T}_1$ and $\mathcal{T}_2$ are not Θ-CQ inseparable for any $\Theta = (\Sigma_1, \Sigma_2)$ with $r \in \Sigma_1$ and $A \in \Sigma_2$ since for the ABox $\mathcal{A} = \{r(a,a)\}$ we have $(\mathcal{T}_1, \mathcal{A}) \not\models A(a)$ and $(\mathcal{T}_2, \mathcal{A}) \models A(a)$.

In Horn DLs, in contrast, concept inseparability and query inseparability are closely related. To explain why this is the case, consider $\mathcal{EL}$ as a paradigmatic example. Since $\mathcal{EL}$ concepts are positive and existential, an $\mathcal{EL}$ concept inclusion $C \sqsubseteq D$ which shows that two TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ are not concept inseparable is almost the same as a witness $(\mathcal{A}, \boldsymbol{q}(\boldsymbol{a}))$ that query separates $\mathcal{T}_1$ and $\mathcal{T}_2$. In fact, both ABoxes and queries are positive and existential as well, but they need not be tree-shaped. Thus, a first puzzle piece is provided by Theorem 31 which implies that we need to consider only tree-shaped queries $\boldsymbol{q}$. This is complemented by the observation that it also suffices to consider only tree-shaped ABoxes $\mathcal{A}$. The latter is also an important foundation for designing decision procedures for query inseparability in Horn DLs. The following result was first proved in [71] for $\mathcal{EL}$. We state it here also for Horn-$\mathcal{ALC}$ [15] as this will be needed later on.

Theorem 33. *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be Horn-$\mathcal{ALC}$ TBoxes and $\Theta = (\Sigma_1, \Sigma_2)$. Then the following are equivalent:*

(i) *$\mathcal{T}_1$ Θ-CQ entails $\mathcal{T}_2$;*
(ii) *for all utree-shaped Σ_1-ABoxes $\mathcal{A}$ and all $\mathcal{EL}$-concepts C in signature Σ_2:*
  (a) *if $(\mathcal{T}_2, \mathcal{A}) \models C(a)$, then $(\mathcal{T}_1, \mathcal{A}) \models C(a)$ where a is the root of $\mathcal{A}$;*
  (b) *if $(\mathcal{T}_2, \mathcal{A}) \models \exists x\, C(x)$, then $(\mathcal{T}_1, \mathcal{A}) \models \exists x\, C(x)$.*

If $\mathcal{T}_1$ and $\mathcal{T}_2$ are $\mathcal{EL}$ TBoxes, then it is sufficient to consider tree-shaped ABoxes in (ii). The same holds when CQs are replaced with rCQs and (b) is dropped from (ii).

Theorem 33 can be proved by an unraveling argument. It is closely related to the notion of unraveling tolerance from [74]. As explained above, Theorem 33 allows us to prove that concept inseparability and query inseparability are the same notion. Here, we state this result only for $\mathcal{EL}$ [71]. Let Σ-$\mathcal{EL}^u$-concept entailment between $\mathcal{EL}$ TBoxes be defined like Σ-concept entailment between $\mathcal{EL}$ TBoxes, except that in $\mathsf{cDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ we now admit concept inclusions $C \sqsubseteq D$ where C is an $\mathcal{EL}$-concept and D an $\mathcal{EL}^u$-concept.

Theorem 34. *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be $\mathcal{EL}$ TBoxes and $\Theta = (\Sigma, \Sigma)$. Then*

(i) *$\mathcal{T}_1$ Σ-concept entails $\mathcal{T}_2$ iff $\mathcal{T}_1$ Θ-rCQ entails $\mathcal{T}_2$;*
(ii) *$\mathcal{T}_1$ Σ-$\mathcal{EL}^u$-concept entails $\mathcal{T}_2$ iff $\mathcal{T}_1$ Θ-CQ entails $\mathcal{T}_2$.*

7.2 Deciding query inseparability of TBoxes

We now study the decidability and computational complexity of query inseparability. Some results can be obtained by transferring results from Section 6 on query inseparability for KBs (in the $\mathcal{ALC}$ and *DL-Lite* case) or results from Section 4 on concept inseparability of TBoxes (in the $\mathcal{EL}$ case). In other cases, though, this does not seem possible. To obtain results for Horn-$\mathcal{ALC}$, in particular, we need new technical machinery; as before, we proceed by first giving model-theoretic characterizations and then using tree automata.

In *DL-Lite*$_{core}$ and *DL-Lite*$^{\mathcal{H}}_{core}$, there is a straightforward reduction of query inseparability of TBoxes to query inseparability of KBs. Informally, such a reduction is possible since DL-Lite TBoxes are so restricted that they can only perform deductions from a single ABox assertion, but not from multiple ones together.

Theorem 35. *For $\mathcal{Q} \in \{CQ, rCQ\}$, Θ-$\mathcal{Q}$ entailment and Θ-$\mathcal{Q}$ inseparability of TBoxes are*

(i) *in* PTime *for DL-Lite$_{core}$;*
(ii) ExpTime-*complete for DL-Lite$^{\mathcal{H}}_{core}$.*

Proof. Let $\Theta = (\Sigma_1, \Sigma_2)$. Using the fact that every CI in a *DL-Lite$^{\mathcal{H}}_{core}$* TBox has only a single concept of the form A or $\exists r.\top$ on the left-hand side, one can show that if $\mathcal{T}_1$ does not Θ-$\mathcal{Q}$ entail $\mathcal{T}_2$, then there exists a singleton Σ_1-ABox (containing either a single assertion of the form $A(c)$ or $r(a, b)$) such that $\mathcal{K}_1$ does not Σ_2-$\mathcal{Q}$ entail $\mathcal{K}_2$ for $\mathcal{K}_i = (\mathcal{T}_i, \mathcal{A})$ for $i = 1, 2$. Now the upper bounds follow from Theorem 30. The lower bound proof is a variation of the one establishing Theorem 30. □

The undecidability proof for CQ (and rCQ) entailment and inseparability of $\mathcal{ALC}$ KBs (Theorem 26) can also be lifted to the TBox case; see [15] for details.

Theorem 36. *Let $\mathcal{Q} \in \{CQ, rCQ\}$.*

(i) *Θ-$\mathcal{Q}$ entailment of an $\mathcal{ALC}$ TBox by an $\mathcal{EL}$ TBox is undecidable.*
(ii) *Θ-$\mathcal{Q}$ inseparability of an $\mathcal{ALC}$ and an $\mathcal{EL}$ TBoxes is undecidable.*

In contrast to the KB case, decidability of UCQ entailment and inseparability of $\mathcal{ALC}$ TBoxes remains open, as well as for the rUCQ versions. Note that, for the extension $\mathcal{ALCF}$ of $\mathcal{ALC}$ with functional roles, undecidability of Θ-$\mathcal{Q}$ inseparability can be proved for any class $\mathcal{Q}$ of queries contained in UCQ and containing an atomic query of the form $A(x)$ or $\exists x A(x)$. The proof is by reduction to predicate and query emptiness problems that are shown to be undecidable in [5]. Consider, for example, the class of all CQs. It is undecidable whether for an $\mathcal{ALCF}$ TBox $\mathcal{T}$, a signature Σ, and a concept name $A \notin \Sigma$, there exists a Σ-ABox $\mathcal{A}$ such that $(\mathcal{T}, \mathcal{A})$ is satisfiable and $(\mathcal{T}, \mathcal{A}) \models \exists x A(x)$ [5]. One can easily modify the TBoxes $\mathcal{T}$ constructed in [5] to prove that this problem is still undecidable if Σ-ABoxes $\mathcal{A}$ such that the KB $(\mathcal{T}, \mathcal{A})$ is not satisfiable are admitted. Now observe that there exists a Σ-ABox $\mathcal{A}$ with $(\mathcal{T}, \mathcal{A}) \models \exists x A(x)$ iff $\mathcal{T}$ is not Θ-$\mathcal{CQ}$ inseparable from the empty TBox for $\Theta = (\Sigma, \{A\})$.

We now consider CQ inseparability in $\mathcal{EL}$. From Theorem 34 and Theorem 9, we obtain ExpTime-completeness of Θ-rCQ inseparability when Θ is of the form (Σ, Σ). ExpTime-completeness of Θ-CQ inseparability in this special case was established in [71]. Both results actually generalize to unrestricted signatures Θ.

Theorem 37. *Let $\mathcal{Q} \in \{CQ, rCQ\}$. In $\mathcal{EL}$, Θ-$\mathcal{Q}$-entailment and inseparability of TBoxes are* ExpTime-*complete.*

Theorem 37 has not been formulated in this generality in the literature, so we briefly discuss proofs. In the rooted case, the ExpTime upper bound follows from the same bound for Horn-$\mathcal{ALC}$ which we discuss below. In the non-rooted case, the ExpTime upper bound for the case $\Theta = (\Sigma, \Sigma)$ in [71] is based on Theorem 34 and a direct algorithm for deciding Σ-$\mathcal{EL}^u$-entailment. It is not difficult to extend this algorithm to the general case. Alternatively, one can obtain the same bound by extending the model-theoretic characterization of Σ-concept entailment in $\mathcal{EL}$ given in Theorem 8 to 'Θ-$\mathcal{EL}^u$-concept entailment', where the concept inclusions in $\mathsf{cDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ are of the form $C \sqsubseteq D$ with C an $\mathcal{EL}$ concept in signature Σ_1 and D an $\mathcal{EL}^u$ concept in signature Σ_2. Based on such a characterization, one can then modify the automata construction from the proof of Theorem 9 to obtain an ExpTime upper bound.

We note that, for acyclic $\mathcal{EL}$ TBoxes (and their extensions with role inclusions and domain and range restrictions), Θ-CQ entailment can be decided in polynomial time. This can be proved by a straightforward generalization of the results in [53] where it is assumed that $\Theta = (\Sigma_1, \Sigma_2)$ with $\Sigma_1 = \Sigma_2$. The proof extends the approach sketched in Section 4 for deciding concept inseparability for acyclic $\mathcal{EL}$ TBoxes. A prototype system deciding Θ-CQ inseparability and computing a representation of the logical difference for query inseparability is presented in [54].

We now consider query entailment and inseparability in Horn-$\mathcal{ALC}$, which requires more effort than the cases discussed so far. We will concentrate on CQs and rCQs. To start with, it is convenient to break down our most basic problem, query entailment, into two subproblems:

1. Θ-$\mathcal{Q}$ entailment *over satisfiable ABoxes* is defined in the same way as Θ-$\mathcal{Q}$ entailment except that only ABoxes satisfiable with both $\mathcal{T}_1$ and $\mathcal{T}_2$ can witness inseparability; see the remark after Definition 13.
2. A TBox $\mathcal{T}_1$ *Σ-ABox entails* a TBox $\mathcal{T}_2$, for a signature Σ, if for every Σ-ABox $\mathcal{A}$, unsatisfiability of $(\mathcal{T}_2, \mathcal{A})$ implies unsatisfiability of $(\mathcal{T}_1, \mathcal{A})$.

It is easy to see that a TBox $\mathcal{T}_2$ is Θ-$\mathcal{Q}$-entailed by a TBox $\mathcal{T}_1$, $\Theta = (\Sigma_1, \Sigma_2)$, if $\mathcal{T}_2$ is Θ-$\mathcal{Q}$-entailed by $\mathcal{T}_1$ over satisfiable ABoxes and $\mathcal{T}_2$ is Σ_1-ABox entailed by $\mathcal{T}_1$. For proving decidability and upper complexity bounds, we can thus concentrate on problems 1 and 2 above. ABox entailment, in fact, is reducible in polynomial time and in a straightforward way to the containment problem of ontology-mediated queries with CQs of the form $\exists x\, A(x)$, which is ExpTime-complete in *Horn-$\mathcal{ALC}$* [9]. For deciding query entailment over satisfiable ABoxes, we can find a transparent model-theoretic characterization. The following result from [15] is essentially a consequence of Theorem 24 (i), Theorem 25, and Theorem 33, but additionally establishes a bound on the branching degree of witness ABoxes.

Theorem 38. *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be Horn-$\mathcal{ALC}$ TBoxes and $\Theta = (\Sigma_1, \Sigma_2)$. Then*

(i) $\mathcal{T}_1$ Θ-CQ entails $\mathcal{T}_2$ over satisfiable ABoxes iff, for all utree-shaped Σ_1-ABoxes $\mathcal{A}$ of outdegree $\leq |\mathcal{T}_2|$ and consistent with $\mathcal{T}_1$ and $\mathcal{T}_2$, $\mathcal{C}_{(\mathcal{T}_2,\mathcal{A})}$ is Σ_2-homomorphically embeddable into $\mathcal{C}_{(\mathcal{T}_1,\mathcal{A})}$;

(ii) $\mathcal{T}_1$ Θ-rCQ entails $\mathcal{T}_2$ over satisfiable ABoxes iff, for all utree-shaped Σ_1-ABoxes $\mathcal{A}$ of outdegree $\leq |\mathcal{T}_2|$ and consistent with $\mathcal{T}_1$ and $\mathcal{T}_2$, $\mathcal{C}_{(\mathcal{T}_2,\mathcal{A})}$ is con-Σ_2-homomorphically embeddable into $\mathcal{C}_{(\mathcal{T}_1,\mathcal{A})}$.

Based on Theorem 38, we can derive upper bounds for query inseparability in *Horn-$\mathcal{ALC}$* using tree automata techniques.

Theorem 39. *In Horn-$\mathcal{ALC}$,*

(i) *Θ-rCQ entailment and inseparability of TBoxes is* ExpTime*-complete;*
(ii) *Θ-CQ entailment and inseparability of TBoxes is 2*ExpTime*-complete.*

The automaton constructions are more sophisticated than those used for proving Theorem 27 because the ABox is not fixed. The construction in [15] uses traditional tree automata whose inputs encode a tree-shaped ABox together with (parts of) its tree-shaped canonical models for the TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$. It is actually convenient to first replace Theorem 27 with a more fine-grained characterization that uses simulations instead of homomorphisms and is more operational. Achieving the upper bounds stated in Theorem 39 requires a careful automaton construction using appropriate bookkeeping in the input and mixing alternating with non-deterministic automata. The lower bound is based on an ATM reduction.

Interestingly, the results presented above for query inseparability between *DL-Lite* TBoxes have recently been applied to analyse containment and inseparability for TBoxes with declarative mappings that relate the signature of the data one wants to query to the signature of the TBox that provides the interface for formulating queries [10]. We conjecture that the results we presented for $\mathcal{EL}$ and Horn-$\mathcal{ALC}$ can also be lifted to the extension by declarative mappings.

We note that query inseparability between TBoxes is closely related to program expressiveness [3] and to CQ-equivalence of schema mappings [30, 83]. The latter is concerned with declarative mappings from a source signature Σ_1 to a target signature Σ_2. Such mappings $\mathcal{M}_1$ and $\mathcal{M}_2$ are CQ-equivalent if, for any data instance in Σ_1, the certain answers to CQs in the signature Σ_2 under $\mathcal{M}_1$ and $\mathcal{M}_2$ coincide. The computational complexity of deciding CQ-equivalence of schema mappings has been investigated in detail [30, 83]. Regarding the former, translated into the language of DL the program expressive power of a TBox $\mathcal{T}$ is the set of all triples $(\mathcal{A}, \boldsymbol{q}, \boldsymbol{a})$ such that $\mathcal{A}$ is an ABox, $\boldsymbol{q}$ is a CQ, and $\boldsymbol{a}$ is a tuple in $\mathsf{ind}(\mathcal{A})$ such that $\mathcal{T}, \mathcal{A} \models q(\boldsymbol{a})$. It follows that two TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ are Θ-CQ inseparable for a pair $\Theta = (\Sigma_1, \Sigma_2)$ iff $\mathcal{T}_1$ and $\mathcal{T}_2$ have the same program expressive power.

8 Discussion

We have discussed a few inseparability relations between description logic TBoxes and KBs, focussing on model-theoretic characterizations and deciding inseparability. In this section, we briefly survey three other important topics that were not covered in the main text. (1) We observe that many inseparability relations considered above (in particular, concept inseparability) fail to satisfy natural robustness conditions such as robustness under replacement, and discuss how to overcome this. (2) Since inseparability tends to be of high computational complexity or even undecidable, it is interesting to develop approximation algorithms; we present a brief overview of the state of the art. (3) One is often not only interested in deciding inseparability, but also in computing useful members of an equivalence class of inseparable ontologies such as uniform interpolants and the result of forgetting irrelevant symbols from an ontology. We briefly survey results in this area as well.

Inseparability and robustness. We have seen that robustness under replacement is a central property in applications of model inseparability to ontology reuse and module extraction. In principle, one can of course also use other inseparability relations for these tasks. The corresponding notion of robustness under replacement can be defined in a straightforward way [55, 61].

Definition 14. Let $\mathcal{L}$ be a DL and $\equiv_\Sigma$ an inseparability relation. Then $\mathcal{L}$ is *robust under replacement* for $\equiv_\Sigma$ if $\mathcal{T}_1 \equiv_\Sigma \mathcal{T}_2$ implies that $\mathcal{T}_1 \cup \mathcal{T} \equiv_\Sigma \mathcal{T}_2 \cup \mathcal{T}$ for all $\mathcal{L}$ TBoxes $\mathcal{T}_1, \mathcal{T}_2$ and $\mathcal{T}$ such that $\mathsf{sig}(\mathcal{T}) \cap \mathsf{sig}(\mathcal{T}_1 \cup \mathcal{T}_2) \subseteq \Sigma$.[12]

Thus, robustness under replacement ensures that Σ-inseparable TBoxes can be equivalently replaced by each other even if a new TBox that shares with $\mathcal{T}_1$ and $\mathcal{T}_2$ only Σ-symbols is added to both. This seems a useful requirement not only for TBox re-use and module extraction, but also for versioning and forgetting. Unfortunately, with the exception of model inseparability, none of the inseparability relations considered in the main part of this survey is robust under replacement for the DLs in question. The following counterexample is a variant of examples given in [61, 52].

Example 33. Suppose $\mathcal{T}_1 = \emptyset$, $\mathcal{T}_2 = \{A \sqsubseteq \exists r.B, E \sqcap B \sqsubseteq \bot\}$, and $\Sigma = \{A, E\}$. Then $\mathcal{T}_1$ and $\mathcal{T}_2$ are Σ-concept inseparable in expressive DLs such as $\mathcal{ALC}$ and they are Θ-CQ inseparable for $\Theta = (\Sigma, \Sigma)$. However, for $\mathcal{T} = \{\top \sqsubseteq E\}$ the TBoxes $\mathcal{T}_1 \cup \mathcal{T}$ and $\mathcal{T}_2 \cup \mathcal{T}$ are neither Σ-concept inseparable nor Θ-CQ inseparable.

The only DLs for which concept inseparability is robust under replacement are certain extensions of $\mathcal{ALC}$ with the universal role. Indeed, recall that by $\mathcal{L}^u$ we denote the extension of a DL $\mathcal{L}$ with the universal role u. Assume that $\mathcal{T}_1$ and $\mathcal{T}_2$ are Σ-concept inseparable in $\mathcal{L}^u$ and let $\mathcal{T}$ be an $\mathcal{L}^u$ TBox with $\mathsf{sig}(\mathcal{T}) \cap \mathsf{sig}(\mathcal{T}_1 \cup \mathcal{T}_2) \subseteq \Sigma$. As $\mathcal{L}$ extends $\mathcal{ALC}$, it is known that $\mathcal{T}$ is logically equivalent to a TBox of the form $\{C \equiv \top\}$, where C is an $\mathcal{L}^u$ concept. Let $D_0 \sqsubseteq D_1$ be a Σ-CI in $\mathcal{L}^u$. Then

$$\begin{aligned} \mathcal{T}_1 \cup \mathcal{T} \models D_0 \sqsubseteq D_1 \quad & \text{iff} \quad \mathcal{T}_1 \models D_0 \sqcap \forall u.C \sqsubseteq D_1 \\ & \text{iff} \quad \mathcal{T}_2 \models D_0 \sqcap \forall u.C \sqsubseteq D_1 \\ & \text{iff} \quad \mathcal{T}_2 \cup \mathcal{T} \models D_0 \sqsubseteq D_1, \end{aligned}$$

where the second equivalence holds by Σ-concept inseparability of $\mathcal{T}_1$ and $\mathcal{T}_2$ *if we assume that* $\mathsf{sig}(\mathcal{T}) \subseteq \Sigma$ (and so $\mathsf{sig}(C) \subseteq \Sigma$). Recall that, in the definition of robustness under replacement, we only require $\mathsf{sig}(\mathcal{T}) \cap \mathsf{sig}(\mathcal{T}_1 \cup \mathcal{T}_2) \subseteq \Sigma$, and so an additional step is needed for the argument to go through. This step is captured by the following definition.

[12] Robustness under replacement can be defined for KBs as well and is equally important in that case. In this short discussion, however, we only consider TBox inseparability.

Definition 15. Let $\mathcal{L}$ be a DL and $\equiv_\Sigma$ an inseparability relation. Then $\mathcal{L}$ is *robust under vocabulary extensions* for $\equiv_\Sigma$ if $\mathcal{T}_1 \equiv_\Sigma \mathcal{T}_2$ implies that $\mathcal{T}_1 \equiv_{\Sigma'} \mathcal{T}_2$ for all $\Sigma' \supseteq \Sigma$ with $\mathsf{sig}(\mathcal{T}_1 \cup \mathcal{T}_2) \cap \Sigma' \subseteq \Sigma$.

Let us return to the argument above. Clearly, if $\mathcal{L}^u$ is robust under vocabulary extensions for concept inseparability, then the second equivalence is justified and we can conclude that $\mathcal{L}^u$ is robust under replacement for concept inseparability. In [55], robustness under vocabulary extensions is investigated for many standard DLs and inseparability relations. In particular, the following is shown:

Theorem 40. *The DLs $\mathcal{ALC}^u$, $\mathcal{ALCI}^u$, $\mathcal{ALCQ}^u$, and $\mathcal{ALCQI}^u$ are robust under vocabulary extensions for concept inseparability, and thus also robust under replacement.*

Because of Theorem 40, it would be interesting to investigate concept inseparability for DLs with the universal role and establish, for example, the computational complexity of concept inseparability. We conjecture that the techniques used to prove the 2ExpTime upper bounds without the universal role can be used to obtain 2ExpTime upper bounds here as well.

We now consider robustness under replacement for DLs without the universal role. To simplify the discussion, we consider *weak robustness under replacement*, which preserves inseparability only if TBoxes $\mathcal{T}$ with $\mathsf{sig}(\mathcal{T}) \subseteq \Sigma$ are added to $\mathcal{T}_1$ and $\mathcal{T}_2$, respectively. It is then a separate task to lift weak robustness under replacement to full robustness under replacement using, for example, robustness under vocabulary extensions. It is, of course, straightforward to extend the inseparability relations studied in this survey in a minimal way so that weak robustness under replacement is achieved. For example, say that two $\mathcal{L}$ TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ are *strongly Σ-concept inseparable in $\mathcal{L}$* if, for all $\mathcal{L}$ TBoxes $\mathcal{T}$ with $\mathsf{sig}(\mathcal{T}) \subseteq \Sigma$, we have that $\mathcal{T}_1 \cup \mathcal{T}$ and $\mathcal{T}_2 \cup \mathcal{T}$ are Σ-concept inseparable. Similarly, say that two $\mathcal{L}$ TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ are *strongly Θ-$\mathcal{Q}$-inseparable* if, for all $\mathcal{L}$ TBoxes $\mathcal{T}$ with $\mathsf{sig}(\mathcal{T}) \subseteq \Sigma_1 \cap \Sigma_2$, we have that $\mathcal{T}_1 \cup \mathcal{T}$ and $\mathcal{T}_2 \cup \mathcal{T}$ are Θ-$\mathcal{Q}$ inseparable (we assume $\Theta = (\Sigma_1, \Sigma_2)$). Unfortunately, with the exception of results for the *DL-Lite* family, nothing is known about the properties of the resulting inseparability relations. It is proved in [52] that strong Θ-CQ inseparability is still in ExpTime for $\textit{DL-Lite}^{\mathcal{H}}_{core}$ if $\Sigma_1 = \Sigma_2$. We conjecture that this result still holds for arbitrary Θ. A variant of strong Θ-CQ inseparability is also discussed in [61] and analyzed for $\textit{DL-Lite}_{core}$ extended with (some) Boolean operators and unqualified number restrictions. However, the authors of [61] do not consider CQ-inseparability as defined in this survey but inseparability with respect to generalized CQs that use atoms $C(x)$ with C a Σ-concept in $\mathcal{L}$ instead of a concept name in Σ. This results in a stronger notion of inseparability that is preserved under definitorial extensions and has, for the *DL-Lite* dialects considered, many of the robustness properties introduced above. It would be of interest to extend this notion of query inseparability to DLs such as $\mathcal{ALC}$. Regarding strong concept and query inseparability, it would be interesting to investigate its algorithmic properties for $\mathcal{EL}$ and Horn-$\mathcal{ALC}$.

Approximation. We have argued throughout this survey that inseparability relations and conservative extensions can play an important role in a variety of applications including ontology versioning, ontology refinement, ontology reuse, ontology modularization, ontology mapping, knowledge base exchange and forgetting. One cannot help noticing, though, another common theme: the high computational complexity of the corresponding reasoning tasks, which can hinder the *practical use* of these notions or even make it infeasible. We now give a brief overview of methods that approximate the notions introduced in the previous sections while incurring lower computational costs. We will focus on modularization and logical difference.

Locality-based approximations have already been discussed in Section 5.3, where we showed how the extraction of depleting modules can be reduced to standard ontology reasoning. Notice that $\emptyset$-locality, in turn, can be approximated with a simple syntactic check. Following [44], let Σ be a signature. Define two sets of $\mathcal{ALCQI}$ concepts $\mathcal{C}^\bot_\Sigma$ and $\mathcal{C}^\top_\Sigma$ as follows:

$$\begin{aligned}\mathcal{C}^\bot_\Sigma &::= A^\bot \mid \neg C^\top \mid C \sqcap C^\bot \mid \exists r^\bot.C \mid \exists r.C^\bot \mid \geq n\, r^\bot.C \mid \geq n\, r.C^\bot,\\ \mathcal{C}^\top_\Sigma &::= \neg C^\bot \mid C^\top_1 \sqcap C^\top_2,\end{aligned}$$

where $A^\bot \notin \Sigma$ is an atomic concept, r is a role (a role name or an inverse role) and C is a concept, $C^\bot \in \mathcal{C}^\bot_\Sigma$, $C^\top_i \in \mathcal{C}^\top_\Sigma$, $i = 1, 2$, and $r^\bot$ is r or r^-, for $r \in \mathsf{N_R} \setminus \Sigma$. A CI α is *syntactically* $\bot$*-local* w.r.t. Σ if it is of the form $C^\bot \sqsubseteq C$ or $C \sqsubseteq C^\top$. A TBox $\mathcal{T}$ is $\bot$*-local* if all CIs in $\mathcal{T}$ are $\bot$-local. Then every TBox $\mathcal{T}$ that is syntactically $\bot$-local w.r.t. a signature Σ is $\emptyset$-local w.r.t. Σ, as shown in [44]. Notice that checking whether a CI is syntactically $\bot$-local can be done in linear time. A dual notion of syntactic $\top$-locality has been introduced in [25]. Both notions can be used to define $\bot$-local and $\top$-local modules; $\top$- and $\bot$-locality module extraction can be iterated leading to smaller modules [93].

A comprehensive study of different locality flavours [101] identified that there is no statistically significant difference in the sizes of semantic and syntactic locality modules. In contrast, [56] found that the difference in size between minimal modules (only available for acyclic $\mathcal{EL}$ TBoxes) and locality-based approximations can be large. In a separate line of research, [35] showed that intractable depleting module approximations for unrestricted OWL ontologies based on reductions to QBF can also be significantly smaller, indicating a possibility for better tractable approximations. Reachability-based approximations [80, 81] refine syntactic locality modules. While they are typically smaller, self-contained and justification preserving, reachability modules are only Σ-concept inseparable from the original TBox but not Σ-model inseparable. A variety of tractable approximations based on notions of inseparability ranging form classification inseparability to model inseparability can be computed by reduction to Datalog reasoning [92].

Syntactic restrictions on elements of $\mathsf{cDiff}_\Sigma(\mathcal{T}_1, \mathcal{T}_2)$ lead to approximations of concept inseparability. In [47], the authors consider counterexamples of the form $A \sqsubseteq B$, $A \sqsubseteq \neg B$, $A \sqsubseteq \exists r.B$, $A \sqsubseteq \forall r.B$ and $r \sqsubseteq s$ only, where A, B are Σ-concept names and r, s are Σ-roles, and use standard reasoners to check for entailment. This approach has been extended in [40] to allow inclusions between Σ-concepts to be constructed in accordance with some grammar rules. In [52], CQ-inseparability for $\textit{DL-Lite}^{\mathcal{H}}_{core}$ is approximated by reduction to a tractable simulation check between the canonical models. An experimental evaluation showed that this approach is incomplete in a very small number of cases on real-world ontologies.

Computing representatives. Inseparability relations are equivalence relations on classes of TBoxes. One is often interested not only in deciding inseparability, but also in computing useful members of an equivalence class of inseparable ontologies such as uniform interpolants (or, equivalently, the result of forgetting irrelevant symbols from an ontology). Recall from Section 3 that an ontology $\mathcal{O}_{\mathsf{forget}}$ is the result of forgetting a signature Γ in $\mathcal{O}$ for an inseparability relation $\equiv$ if $\mathcal{O}$ uses only symbols in $\Sigma = \mathsf{sig}(\mathcal{O}) \setminus \Gamma$ and $\mathcal{O}$ and $\mathcal{O}_{\mathsf{forget}}$ are Σ-inseparable for $\equiv$. Clearly, $\mathcal{O}_{\mathsf{forget}}$ can be regarded as a representation of its equivalence class under $\equiv_{\Sigma}$. For model-inseparability, this representation is unique up to logical equivalence while this need not be the case for other inseparability relations.

Forgetting has been studied extensively for various inseparability relations. A main problems addressed in the literature is that, for most inseparability relations and ontology languages, the result of forgetting is not guaranteed to be expressible in the language of the original ontology. For example, for the TBox $\mathcal{T} = \{A \sqsubseteq \exists r.B, B \sqsubseteq \exists r.B\}$, there is no $\mathcal{ALC}$ TBox using only symbols from $\Sigma = \{A, r\}$ that is Σ-concept inseparable from $\mathcal{T}$. This problem gives rise to three interesting research problems: given an ontology $\mathcal{O}$ and signature Σ, can we decide whether the result of forgetting exists in the language of $\mathcal{O}$ and, if so, compute it? If not, can we approximate it in a principled way? Or can we express it in a more powerful ontology language? The existence and computation of uniform interpolants for TBoxes under concept inseparability has been studied in [59] for acyclic $\mathcal{EL}$ TBoxes, in [79, 69] for arbitrary $\mathcal{EL}$ TBoxes, and for $\mathcal{ALC}$ and more expressive DLs in [72, 62]. The generalization to KBs has been studied in [105, 104, 63]. Approximations of uniform interpolants obtained by putting a bound on the role depth of relevant concept inclusions are studied in [103, 108, 104, 66]. In [63, 62], (a weak form of) uniform interpolants that do not exist in the original DL are captured using fix-point operators. The relationship between deciding concept inseparability and deciding the existence of uniform interpolants is investigated in [72]. Forgetting under model inseparability has been studied extensively in logic [34] and more recently for DLs [107]. Note that the computation of universal CQ solutions in knowledge exchange [2] is identical to forgetting the signature of the original KB under Σ-CQ-inseparability.

Uniform interpolants are not the only useful representatives of equivalence classes of inseparable ontologies. In the KB case, for example, it is natural to ask whether for a given KB $\mathcal{K}$ and signature Σ there exists a KB $\mathcal{K}'$ with empty TBox that is Σ-query inseparable from $\mathcal{K}$. In this case, answering a Σ-query in $\mathcal{K}$ could be reduced to evaluating the query in an ABox. Another example is TBox rewriting, which asks whether for a given TBox $\mathcal{T}$ in an expressive DL there exists a TBox $\mathcal{T}'$ that is $\mathsf{sig}(\mathcal{T})$-inseparable from $\mathcal{T}$ in a less expressive DL.

In this case tools that are only available for the less expressive DL but not for the expressive DL would become applicable to the rewritten TBox. First results regarding this question have been obtained in [57].

Acknowledgements. Elena Botoeva was supported by EU IP project Optique, grant n. FP7-318338. Carsten Lutz was supported by ERC grant 647289. Boris Konev, Frank Wolter and Michael Zakharyaschev were supported by the UK EPSRC grants EP/M012646, EP/M012670, EP/H043594, and EP/H05099X.

References

1. S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases.* Addison-Wesley, 1995.
2. M. Arenas, E. Botoeva, D. Calvanese, and V. Ryzhikov. Knowledge base exchange: The case of OWL 2 QL. *Artificial Intelligence*, 238:11–62, 2016.
3. M. Arenas, G. Gottlob, and A. Pieris. Expressive languages for querying the semantic web. In *Proceedings of the 33rd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS'14, Snowbird, UT, USA, June 22-27, 2014*, pages 14–26, 2014.
4. A. Artale, D. Calvanese, R. Kontchakov, and M. Zakharyaschev. The DL-Lite family and relations. *Journal of Artificial Intelligence Research (JAIR)*, 36:1–69, 2009.
5. F. Baader, M. Bienvenu, C. Lutz, and F. Wolter. Query and predicate emptiness in ontology-based data access. *Journal of Artificial Intelligence Research (JAIR)*, 56:1–59, 2016.
6. F. Baader, S. Brandt, and C. Lutz. Pushing the EL envelope. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005)*, pages 364–369, 2005.
7. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications.* Cambridge University Press, 2003. (2nd edition, 2007).
8. R. Berger. The undecidability of the domino problem. *Memoirs of the AMS*, 66, 1966.
9. M. Bienvenu, P. Hansen, C. Lutz, and F. Wolter. First order-rewritability and containment of conjunctive queries in Horn description logics. In *Proc. of the 25th Int. Joint Conf. on Artificial Intelligence (IJCAI 2016)*, pages 965 – 971, 2016.
10. M. Bienvenu and R. Rosati. Query-based comparison of mappings in ontology-based data access. In *Proc. of the 15th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2016)*, pages 197–206, 2016.
11. M. Bienvenu, B. ten Cate, C. Lutz, and F. Wolter. Ontology-based data access: A study through Disjunctive Datalog, CSP, and MMSNP. *ACM Trans. on Database Systems*, 39(4):33:1–33:44, 2014.
12. P. A. Bonatti, M. Faella, C. Lutz, L. Sauro, and F. Wolter. Decidability of circumscribed description logics revisited. In *Advances in Knowledge Representation, Logic Programming, and Abstract Argumentation - Essays Dedicated to Gerhard Brewka on the Occasion of His 60th Birthday*, pages 112–124, 2015.
13. E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem.* Perspectives in Mathematical Logic. Springer, 1997.
14. E. Botoeva, R. Kontchakov, V. Ryzhikov, F. Wolter, and M. Zakharyaschev. Games for query inseparability of description logic knowledge bases. *Artificial Intelligence*, 234:78 – 119, 2016.

15. E. Botoeva, C. Lutz, V. Ryzhikov, F. Wolter, and M. Zakharyaschev. Query-based entailment and inseparability for ALC ontologies. In *Proc. of the 25th Int. Joint Conf. on Artificial Intelligence (IJCAI 2016)*, pages 1001–1007, 2016.
16. E. Botoeva, C. Lutz, V. Ryzhikov, F. Wolter, and M. Zakharyaschev. Query-based entailment and inseparability for ALC ontologies (Full Version). CoRR Technical Report abs/1604.04164, arXiv.org e-Print archive, 2016. Available at http://arxiv.org/abs/1604.04164.
17. P. Byers and D. Pitt. Conservative extensions: a cautionary note. *EATCS-Bulletin*, pages 196–201, 1997.
18. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Data complexity of query answering in description logics. In *Proc. of the 10th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2006)*, pages 260–270, 2006.
19. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *Journal of Automated Reasoning*, 39(3):385–429, 2007.
20. D. Calvanese, T. Eiter, and M. Ortiz. Answering regular path queries in expressive description logics: An automata-theoretic approach. In *Proc. of the 22nd Nat. Conf. on Artificial Intelligence (AAAI 2007)*, pages 391–396, 2007.
21. C. C. Chang and H. J. Keisler. *Model Theory*, volume 73 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 1990.
22. K. Chatterjee and M. Henzinger. An O(n^2) time algorithm for alternating Büchi games. In *Proc. of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1386–1399. SIAM, 2012.
23. E. Clarke and H. Schlingloff. Model checking. In *Handbook of Automated Reasoning*, volume II, chapter 24, pages 1635–1790. Elsevier, Amsterdam, The Netherlands, 2001.
24. R. Conradi and B. Westfechtel. Version models for software configuration management. *ACM Computing Surveys (CSUR)*, 30(2):232–282, 1998.
25. B. Cuenca Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Modular reuse of ontologies: Theory and practice. *Journal of Artificial Intelligence Research (JAIR)*, 31:273–318, 2008.
26. B. Cuenca Grau and B. Motik. Reasoning over ontologies with hidden content: The import-by-query approach. *Journal of Artificial Intelligence Research (JAIR)*, 45:197–255, 2012.
27. G. D'Agostino and M. Hollenberg. Uniform interpolation, automata, and the modal μ-calculus. In *Advances in Modal Logic*, volume 1, 1998.
28. T. Eiter and M. Fink. Uniform equivalence of logic programs under the stable model semantics. In *Proc. of the 19th Int. Conf. on Logic Programming (ICLP 2003)*, pages 224–238, 2003.
29. R. Fagin, P. G. Kolaitis, R. J. Miller, and L. Popa. Data exchange: semantics and query answering. *Theoretical Computer Science*, 336(1):89–124, 2005.
30. R. Fagin, P. G. Kolaitis, A. Nash, and L. Popa. Towards a theory of schema-mapping optimization. In *Proc. of the 27th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS 2008)*, pages 33–42, 2008.
31. S. Feferman and R. L. Vaught. The first-order properties of algebraic systems. *Fundamenta Mathematicae*, 47:57–103, 1959.
32. S. Feng, M. Ludwig, and D. Walther. Foundations for the logical difference of EL-TBoxes. In *Global Conference on Artificial Intelligence (GCAI 2015)*, volume 36 of *EPiC Series in Computing*, pages 93–112. EasyChair, 2015.

33. T. French. *Bisimulation quantifiers for modal logics.* PhD thesis, University of Western Australia, 2006.
34. D. M. Gabbay, R. A. Schmidt, and A. Szałas. *Second-Order Quantifier Elimination: Foundations, Computational Aspects and Applications*, volume 12 of *Studies in Logic: Mathematical Logic and Foundations.* College Publications, 2008.
35. W. Gatens, B. Konev, and F. Wolter. Lower and upper approximations for depleting modules of description logic ontologies. In *Proc. of the 21st Eur. Conf. on Artificial Intelligence (ECAI 2014)*, volume 263, pages 345–350. IOS Press, 2014.
36. S. Ghilardi, C. Lutz, and F. Wolter. Did I damage my ontology? A case for conservative extensions in description logic. In *Proc. of the 10th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2006)*, pages 187–197. AAAI Press, 2006.
37. S. Ghilardi, C. Lutz, F. Wolter, and M. Zakharyaschev. Conservative extensions in modal logics. In *Proc. of AiML 6*, pages 187–207, 2006.
38. S. Ghilardi and M. Zawadowski. Undefinability of propositional quantifiers in the modal system S4. *Studia Logica*, 55, 1995.
39. B. Glimm, C. Lutz, I. Horrocks, and U. Sattler. Answering conjunctive queries in the $\mathcal{SHIQ}$ description logic. *Journal of Artificial Intelligence Research (JAIR)*, 31:150–197, 2008.
40. R. S. Gonçalves, B. Parsia, and U. Sattler. Concept-based semantic difference in expressive description logics. In *Proc. of the 11th Int. Semantic Web Conf. (ISWC 2012)*, volume 7649 of *Lecture Notes in Computer Science*, pages 99–115. Springer, 2012.
41. V. Goranko and M. Otto. Model theory of modal logic. In Blackburn, van Benthem, and Wolter, editors, *Handbook of Modal Logic*, pages 249–330. Elsevier, 2006.
42. B. C. Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Just the right amount: extracting modules from ontologies. In *Proc. of the 16th Int. World Wide Web Conf. (WWW 2007)*, pages 717–726. ACM, 2007.
43. B. C. Grau, I. Horrocks, Y. Kazakov, and U. Sattler. A logical framework for modularity of ontologies. In *Proc. of the 20th Int. Joint Conf. on Artificial Intelligence (IJCAI 2007)*, pages 298–303, 2007.
44. B. C. Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Extracting modules from ontologies: A logic-based approach. In *Modular Ontologies*, volume 5445 of *Lecture Notes in Computer Science*, pages 159–186. Springer, 2009.
45. I. Horrocks and U. Sattler. A description logic with transitive and inverse roles and role hierarchies. *J. Log. Comput.*, 9(3):385–410, 1999.
46. U. Hustadt, B. Motik, and U. Sattler. Reasoning in description logics by a reduction to disjunctive Datalog. *Journal of Automated Reasoning*, 39(3):351–384, 2007.
47. E. Jimenez-Ruiz, B. Cuenca Grau, I. Horrocks, and R. B. Llavori. Supporting concurrent ontology development: Framework, algorithms and tool. *Data and Knowledge Engineering*, 70(1):146–164, 2011.
48. E. Jiménez-Ruiz, T. R. Payne, A. Solimando, and V. A. M. Tamma. Limiting logical violations in ontology alignnment through negotiation. In *Proc. of the 15th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2016)*, pages 217–226, 2016.
49. Y. Kazakov. Consequence-driven reasoning for Horn-SHIQ ontologies. In *Proc. of the 21st Int. Joint Conf. on Artificial Intelligence (IJCAI 2009)*, pages 2040–2045, 2009.

50. E. Kharlamov, D. Hovland, E. Jiménez-Ruiz, D. Lanti, H. Lie, C. Pinkel, M. Rezk, M. G. Skjæveland, E. Thorstensen, G. Xiao, D. Zheleznyakov, and I. Horrocks. Ontology based access to exploration data at Statoil. In *Proc. of the 14th Int. Semantic Web Conf. (ISWC 2015)*, pages 93–112, 2015.
51. M. C. A. Klein, D. Fensel, A. Kiryakov, and D. Ognyanov. Ontology versioning and change detection on the web. In *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*, volume 2473 of *Lecture Notes in Computer Science*, pages 247–259. Springer Verlag, Berlin/Heidelberg, Germany, 2002.
52. B. Konev, R. Kontchakov, M. Ludwig, T. Schneider, F. Wolter, and M. Zakharyaschev. Conjunctive query inseparability of OWL 2 QL TBoxes. In *Proc. of the 25th Nat. Conf. on Artificial Intelligence (AAAI 2011)*, pages 221–226. AAAI Press, 2011.
53. B. Konev, M. Ludwig, D. Walther, and F. Wolter. The logical difference for the lightweight description logic EL. *Journal of Artificial Intelligence Research (JAIR)*, 44:633–708, 2012.
54. B. Konev, M. Ludwig, and F. Wolter. Logical difference computation with CEX2.5. In *Proc. of the 6th Int. Joint Conf. on Automated Reasoning (IJCAR 2012)*, Lecture Notes in Computer Science, Berlin/Heidelberg, Germany, 2012. Springer.
55. B. Konev, C. Lutz, D. Walther, and F. Wolter. Formal properties of modularisation. In Stuckenschmidt et al. [97], pages 25–66.
56. B. Konev, C. Lutz, D. Walther, and F. Wolter. Model-theoretic inseparability and modularity of description logic ontologies. *Artificial Intelligence*, 203:66–103, 2013.
57. B. Konev, C. Lutz, F. Wolter, and M. Zakharyaschev. Conservative rewritability of description logic TBoxes. In *Proc. of the 25th Int. Joint Conf. on Artificial Intelligence (IJCAI 2016)*, 2016.
58. B. Konev, D. Walther, and F. Wolter. The logical difference problem for description logic terminologies. In *Proc. of the 4th Int. Joint Conf. on Automated Reasoning (IJCAR 2008)*, volume 5195 of *Lecture Notes in Computer Science*, pages 259–274, Berlin/Heidelberg, Germany, 2008. Springer Verlag.
59. B. Konev, D. Walther, and F. Wolter. Forgetting and uniform interpolation in large-scale description logic terminologies. In *Proc. of the 21st Int. Joint Conf. on Artificial Intelligence (IJCAI 2009)*, pages 830–835, 2009.
60. R. Kontchakov, F. Wolter, and M. Zakharyaschev. Can you tell the difference between DL-Lite ontologies. In *Proc. of the 11th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2008)*, pages 285–295, 2008.
61. R. Kontchakov, F. Wolter, and M. Zakharyaschev. Logic-based ontology comparison and module extraction, with an application to DL-Lite. *Artificial Intelligence*, 174:1093–1141, 2010.
62. P. Koopmann and R. A. Schmidt. Count and forget: Uniform interpolation of $\mathcal{SHQ}$-ontologies. In *Proc. of the 7th Int. Joint Conf. on Automated Reasoning (IJCAR 2014)*, pages 434–448, 2014.
63. P. Koopmann and R. A. Schmidt. Uniform interpolation and forgetting for $\mathcal{ALC}$ ontologies with ABoxes. In *Proc. of the 29th Nat. Conf. on Artificial Intelligence (AAAI 2015)*, pages 175–181, 2015.
64. O. Kutz, T. Mossakowski, and D. Lücke. Carnap, Goguen, and the hyperontologies: Logical pluralism and heterogeneous structuring in ontology design. *Logica Universalis*, 4(2):255–333, 2010.

65. V. Lifschitz, D. Pearce, and A. Valverde. Strongly equivalent logic programs. *ACM Trans. on Computational Logic*, 2(4):526–541, 2001.
66. M. Ludwig and B. Konev. Practical uniform interpolation and forgetting for $\mathcal{ALC}$ tboxes with applications to logical difference. In *Proc. of the 14th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2014)*. AAAI Press, 2014.
67. M. Ludwig and D. Walther. The logical difference for $\mathcal{ELH}^r$-terminologies using hypergraphs. In *Proc. of the 21st Eur. Conf. on Artificial Intelligence (ECAI 2014)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 555–560. IOS Press, 2014.
68. C. Lutz, R. Piro, and F. Wolter. Description logic TBoxes: Model-theoretic characterizations and rewritability. In *Proc. of the 22nd Int. Joint Conf. on Artificial Intelligence (IJCAI 2011)*, pages 983–988, Menlo Park, CA, USA, 2011. AAAI Press.
69. C. Lutz, I. Seylan, and F. Wolter. An automata-theoretic approach to uniform interpolation and approximation in the description logic EL. In *Proc. of the 13th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2012)*, pages 286–296. AAAI Press, 2012.
70. C. Lutz, D. Walther, and F. Wolter. Conservative extensions in expressive description logics. In *Proc. of the 20th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 453–458, Menlo Park, CA, USA, 2007. AAAI Press.
71. C. Lutz and F. Wolter. Deciding inseparability and conservative extensions in the description logic EL. *J. of Symbolic Computation*, 45(2):194–228, 2010.
72. C. Lutz and F. Wolter. Foundations for uniform interpolation and forgetting in expressive description logics. In *Proc. of the 22nd Int. Joint Conf. on Artificial Intelligence (IJCAI 2011)*, pages 989–995. IJCAI/AAAI, 2011.
73. C. Lutz and F. Wolter. Non-uniform data complexity of query answering in description logics. In *Proc. of the 13th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2012)*, 2012.
74. C. Lutz and F. Wolter. Non-uniform data complexity of query answering in description logics. In *Proc. of KR*. AAAI Press, 2012.
75. M. J. Maher. Equivalences of logic programs. In *Foundations of Deductive Databases and Logic Programming*, pages 627–658. Morgan Kaufmann, 1988.
76. T. Maibaum. Conservative extensions, interpretations between theories and all that! In *Proc. of CAAP/FASE*, LNCS. Springer Verlag, 1997.
77. R. Mazala. Infinite games. In *Automata, Logics, and Infinite Games: A Guide to Current Research [outcome of a Dagstuhl seminar, February 2001]*, volume 2500 of *Lecture Notes in Computer Science*, pages 23–42. Springer, 2002.
78. N. Nikitina and S. Rudolph. ExpExpExplosion: Uniform interpolation in general EL terminologies. In *Proc. of the 20th Eur. Conf. on Artificial Intelligence (ECAI 2012)*, pages 618–623, 2012.
79. N. Nikitina and S. Rudolph. (Non-)succinctness of uniform interpolants of general terminologies in the description logic EL. *Artif. Intell.*, 215:120–140, 2014.
80. R. Nortje, A. Britz, and T. Meyer. Module-theoretic properties of reachability modules for SRIQ. In *Proc. of the 26th Int. Workshop on Description Logics (DL 2013)*, volume 1014 of *CEUR Workshop Proceedings*, pages 868–884. CEUR-WS.org, 2013.
81. R. Nortje, K. Britz, and T. Meyer. Reachability modules for the description logic $\mathcal{SRIQ}$. In *Proc. of the 19th Int. Conf. on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2013)*, volume 8312 of *Lecture Notes in Computer Science*, pages 636–652. Springer, 2013.

82. N. F. Noy and M. A. Musen. PromptDiff: A fixed-point algorithm for comparing ontology versions. In *Proc. of the 18th Nat. Conf. on Artificial Intelligence (AAAI 2002)*, pages 744–750, Menlo Park, CA, USA, 2002. AAAI Press.
83. R. Pichler, E. Sallinger, and V. Savenkov. Relaxed notions of schema mapping equivalence revisited. *Theory Comput. Syst.*, 52(3):483–541, 2013.
84. A. Pitts. On an interpretation of second-order quantification in first-order intuitionistic propositional logic. *J. of Symbolic Logic*, 57, 1992.
85. A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Linking data to ontologies. *J. on Data Semantics*, 10:133–173, 2008.
86. J. G. R. Diaconescu and P. Stefaneas. Logical support for modularisation. In G. Huet and G. Plotkin, editors, *Logical Environments*, 1993.
87. T. Redmond, M. Smith, N. Drummond, and T. Tudorache. Managing change: An ontology version control system. In *Proc. of the 5th Int. Workshop on OWL: Experiences and Directions (OWLED 2008)*, volume 432 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.
88. R. Reiter and F. Lin. Forget it! In *Proceedings of AAAI Fall Symposium on Relevance*, pages 154–159, 1994.
89. R. Robinson. Undecidability and nonperiodicity for tilings of the plane. *Inventiones Mathematicae*, 12:177–209, 1971.
90. A. A. Romero, B. C. Grau, and I. Horrocks. Modular combination of reasoners for ontology classification. In *Proc. of the 25th Int. Workshop on Description Logics (DL 2012)*, volume 846 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2012.
91. A. A. Romero, B. C. Grau, I. Horrocks, and E. Jiménez-Ruiz. MORe: a modular OWL reasoner for ontology classification. In *ORE*, volume 1015 of *CEUR Workshop Proceedings*, pages 61–67. CEUR-WS.org, 2013.
92. A. A. Romero, M. Kaminski, B. C. Grau, and I. Horrocks. Module extraction in expressive ontology languages via Datalog reasoning. *J. Artif. Intell. Res. (JAIR)*, 55:499–564, 2016.
93. U. Sattler, T. Schneider, and M. Zakharyaschev. Which kind of module should I extract? In *Proc. of the 22th Int. Workshop on Description Logics (DL 2009)*, volume 477 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2009.
94. A. Schaerf. *Query Answering in Concept-Based Knowledge Representation Systems: Algorithms, Complexity, and Semantic Issues.* PhD thesis, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", 1994.
95. P. Shvaiko and J. Euzenat. Ontology matching: State of the art and future challenges. *IEEE Trans. on Knowledge and Data Engineering*, 25(1):158–176, 2013.
96. A. Solimando, E. Jiménez-Ruiz, and G. Guerrini. Detecting and correcting conservativity principle violations in ontology-to-ontology mappings. In *Proc. of the 13th Int. Semantic Web Conf. (ISWC 2014)*, pages 1–16, 2014.
97. H. Stuckenschmidt, C. Parent, and S. Spaccapietra, editors. *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, volume 5445 of *Lecture Notes in Computer Science*. Springer, 2009.
98. K. Su, A. Sattar, G. Lv, and Y. Zhang. Variable forgetting in reasoning about knowledge. *Journal of Artificial Intelligence Research (JAIR)*, 35:677–716, 2009.
99. P. Veloso. Yet another cautionary note on conservative extensions: a simple case with a computing flavour. *EATCS-Bulletin*, pages 188–193, 1992.
100. P. Veloso and S. Veloso. Some remarks on conservative extensions. A socratic dialog. *EATCS-Bulletin*, pages 189–198, 1991.

101. C. D. Vescovo, P. Klinov, B. Parsia, U. Sattler, T. Schneider, and D. Tsarkov. Empirical study of logic-based modules: Cheap is cheerful. In *Proc. of the 12th Int. Semantic Web Conf. (ISWC 2013)*, volume 8218 of *Lecture Notes in Computer Science*, pages 84–100. Springer, 2013.
102. A. Visser. Uniform interpolation and layered bisimulation. In *Gödel '96 (Brno, 1996)*, volume 6 of *Lecture Notes Logic*. Springer Verlag, 1996.
103. K. Wang, Z. Wang, R. W. Topor, J. Z. Pan, and G. Antoniou. Concept and role forgetting in $\mathcal{ALC}$ ontologies. In *Proc. of the 8th Int. Semantic Web Conf. (ISWC 2009)*, volume 5823 of *Lecture Notes in Computer Science*, pages 666–681. Springer, 2009.
104. K. Wang, Z. Wang, R. W. Topor, J. Z. Pan, and G. Antoniou. Eliminating concepts and roles from ontologies in expressive descriptive logics. *Computational Intelligence*, 30(2):205–232, 2014.
105. Z. Wang, K. Wang, R. W. Topor, and J. Z. Pan. Forgetting for knowledge bases in DL-Lite. *Ann. of Mathematics and Artificial Intelligence*, 58(1-2):117–151, 2010.
106. T. Wilke. Alternating tree automata, parity games, and modal μ-calculus. *Bulletin of the Belgian Mathematical Society*, 8(2), 2001.
107. Y. Zhao and R. A. Schmidt. Concept forgetting in $\mathcal{ALCOI}$-ontologies using an ackermann approach. In *Proc. of the 14th Int. Semantic Web Conf. (ISWC 2015)*, pages 587–602, 2015.
108. Y. Zhou and Y. Zhang. Bounded forgetting. In W. Burgard and D. Roth, editors, *Proc. of the 25th Nat. Conf. on Artificial Intelligence (AAAI 2011)*. AAAI Press, 2011.