
Formal Properties of Modularisation

Boris Konev¹, Carsten Lutz², Dirk Walther¹, and Frank Wolter¹

¹ University of Liverpool, UK
`{konev, dwalther, wolter}@liverpool.ac.uk`

² University of Bremen, Germany
`clu@informatik.uni-bremen.de`

Summary. Modularity of ontologies is currently an active research field, and many different notions of a module have been proposed. In this paper, we review the fundamental principles of modularity and identify formal properties that a robust notion of modularity should satisfy. We explore these properties in detail in the contexts of description logic and classical predicate logic and put them into the perspective of well-known concepts from logic and modular software specification such as interpolation, forgetting and uniform interpolation. We also discuss reasoning problems related to modularity.

1 Introduction

The benefits of modular ontologies are manifold. In ontology design, modularity supports the structured and controlled development of large ontologies, enables ontology design by multiple, possibly distributed designers and allows the re-use of (parts of) already existing ontologies. In ontology deployment and usage, modularity can be exploited for right-sizing large ontologies (by selecting and using only the relevant part) and to speed up reasoning. Alas, making full use of modularity is hampered by the fact that there are many different definitions of what a module in an ontology actually is. In fact, it seems unlikely that there can be a unique and generally accepted such definition because the desirable properties of a module strongly depends on the intended application.

In this paper, our aim is to provide guidance for choosing the right notion of modularity. In particular, we give a survey of possible options and identify three formal properties that a notion of modularity may or may not enjoy and that can be used to evaluate the robustness of this notion in the context of a given application. We analyze whether the surveyed notions of modularity satisfy these robustness properties and provide further guidance by discussing the computational complexity of central decision problems associated with modularity.

To make different notions of modularity comparable to each other, we have to agree on some general framework for studying ontologies and modules. Generally speaking, a module is a part of a complex system that functions independently from this system. To define what a *module in an ontology* is, we thus have to specify what it means for such a module to function independently from the containing ontology. It is not a priori obvious how this can be done. We start with adopting the following, abstract view of an ontology: an ontology \mathcal{O} can be regarded as a black box that provides answers to queries about some vocabulary S of interest. The form that such queries take is one of the main distinguishing factors between different applications. Important cases include the following.

Classification. Classifying an ontology means to compute the sub-/superclass relationships between all atomic classes in the ontology. For example, if S is a vocabulary for buildings and architecture, then the query $\text{Church} \sqsubseteq \text{Building}$ asks whether every church is a building.

Subsumption queries. In other applications, one is interested in computing subsumption between complex class expressions. For example, if S is a biological vocabulary, then the query $\text{Father} \sqsubseteq \text{Living_being} \sqcap \exists \text{has_child}. asks whether every father is a living being having a child.$

Instance data. A popular application of ontologies is their use for providing a background theory when querying instance data. In this case, one is interested in instance queries that are posed to a knowledge base, which consists of an ontology and an ABox that stores instances of classes and relations. Note that we do not consider the ABox to be part of the ontology. To represent this setup in terms of queries posed to the ontology, we consider queries that consist of an instance query *together* with an ABox. For example, if S is a geographical vocabulary, then a query might consist of the instance data

$$\mathcal{A} = \{\text{Country}(\text{France}), \text{Country}(\text{Columbia}), \dots, \text{LocatedinEurope}(\text{France}), \dots\}$$

together with the conjunctive query $\text{EuropeanCountry}(\text{France})$. This query asks whether it follows from the instance data \mathcal{A} and the ontology that France is a European country. The answer is yes if, for example, the ontology states that every country LocatedinEurope is a EuropeanCountry .

These examples show that, to define what it means for a part of an ontology to “function independently”, we first have to fix a query language and a vocabulary of interest. Once this is done, two ontologies can be regarded equivalent if they give the same answers to all queries that can be built in the fixed query language with the fixed vocabulary. Similarly, given an ontology and its module, we can say that the module functions independently from the ontology if any query built in the fixed query language with the vocabulary associated with the module has the same answer when querying the module and the whole ontology.

Formally, these ideas can be captured by the concept of *inseparability*: given a query language \mathcal{QL} and a vocabulary S , two ontologies \mathcal{O}_1 and \mathcal{O}_2 are

S-inseparable w.r.t. \mathcal{QL} if they give the same answers to queries in \mathcal{QL} over S . There are various ways in which this notion can be used to define modularity of ontologies. For example, one can use the notion of inseparability to define a module as a subset of an ontology that is S -inseparable (and thus functions independently) from the whole ontology w.r.t. a query language \mathcal{QL} and for a signature S associated with the module. In this case, investigations into modularity boil down to investigating conservative extensions as the special case of S -inseparability in which one ontology is a subset of the other ontology and the vocabulary S coincides with the vocabulary of the smaller ontology [14]. Another, differently flavoured approach is to use triples $(\mathcal{O}, \mathcal{QL}, S)$ as a module, where \mathcal{O} is an ontology, \mathcal{QL} a query language and S a vocabulary. The pair (\mathcal{QL}, S) serves as an interface so that groups of independent modules can interact by querying each other via this interface. This setup is similar in spirit to formalisms such as DDLs and E-Connections [32, 18, 45, 4, 6]. In this case, S -inseparability is fundamental because it allows to define what it means that one module is equivalent to another one.

Since inseparability is at the core of most notions of modularity, our framework for studying and comparing such notions puts inseparability (and, as a variant, conservative extensions) into its focus. The robustness properties mentioned above are formulated directly in terms of inseparability. Here, the term “robustness” refers to the ramifications of changing the signature and manipulating in certain natural ways the involved ontologies. In particular, the properties ensure that modules and ontologies can be composed and decomposed in a transparent way. Our robustness properties are closely related to well-known notions from logic, in particular to interpolation and the Robinson joint consistency property. We explore this connection and also investigate the relation between inseparability on the one hand, and forgetting and uniform interpolation on the other. In principle, the general ideas presented in this paper are independent of the ontology language and query language that are used. To analyze robustness properties in concrete cases, though, we obviously have to fix both languages. As an ontology language, we mainly consider description logics (DLs), which are a highly relevant in this context due to the standardisation of the DL-based ontology language OWL by the W3C [5]. To round off our presentation, we will sometimes also give examples in terms of first- and second-order logic. We consider a wide range of query languages including classification queries, subsumption queries, instance queries, conjunctive queries and first- and second-order queries.

The rest of this paper is organised as follows. In Section 2, we introduce ontology languages and description logics. In Section 3, we define our framework for inseparability and introduce relevant robustness properties that notions of inseparability should enjoy. A variety of query languages together with the resulting robustness properties and decision problems are discussed in Section 5. A detailed investigation of robustness properties and their relation to interpolation is provided in Section 6. In Section 7 we explore the connection

Name	Syntax	Semantics
top concept	\top	$\Delta^{\mathcal{I}}$
bottom concept	\perp	\emptyset
negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
disjunction	$C \sqcup D$	$\neg(\neg C \sqcap \neg D)^{\mathcal{I}}$
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	$\forall r.C$	$\neg(\exists r.\neg C)^{\mathcal{I}}$

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

between inseparability and forgetting/uniform interpolation and establish a number of results regarding the existence of uniform interpolants and ontologies that “forget” certain symbols from a given ontology. Finally, we devote Section 8 to surveying results on lightweight DLs and acyclic ontologies. We finish with a brief discussion in Section 9.

2 Ontology Languages and Description Logics

We introduce a number of description logics, fix conventions for handling first- and second-order logic and give a number of basic definitions concerning signatures. The DL-literate reader may choose to skip this section.

2.1 The Description Logic \mathcal{ALC}

In DLs, ontologies are sets of implications between concepts, sometimes enriched with additional types of constraints. Thus, we start this section with introducing concepts, which are inductively defined from a set \mathbf{N}_C of *concept names* and a set \mathbf{N}_R of *role names*, using a set of *concept constructors*. From the perspective of first-order logic, concept names are unary predicates and role names are binary relations. Different sets of concept constructors give rise to different DLs.

In this paper, we will mainly be concerned with the description logic \mathcal{ALC} and its extensions. The concept constructors available in \mathcal{ALC} are shown in Table 1, where r denotes a role name and C and D denote concepts. A concept built from these constructors is called an \mathcal{ALC} -concept. A *concept implication* is an expression of the form $C \sqsubseteq D$, with C and D concepts. We write $C \equiv D$ instead of the two concept implications $C \sqsubseteq D$ and $D \sqsubseteq C$.

Definition 1 (\mathcal{ALC} -Ontology). An \mathcal{ALC} -ontology is a finite set of concept implications. \mathcal{ALC} -ontologies will also be called \mathcal{ALC} -TBoxes.

Some of the applications discussed in this paper are concerned not only with ontologies, but also with instance data. In DLs, such instance data is described using ABoxes. To introduce ABoxes, we fix a set \mathbf{N}_I of *individual names*, which correspond to constants in first-order logic. Then, an *assertion* is an expression of the form $C(a)$ or $r(a, b)$, where C is a concept, r a role name and a, b are individual names. An *ALC-ABox* is a finite set of assertions. We call the combination $\mathcal{K} = (\mathcal{O}, \mathcal{A})$ of an *ALC-ontology* and an *ALC-ABox* an *ALC-knowledge base*.

DL semantics is based on the notion of an *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$. The *domain* $\Delta^{\mathcal{I}}$ is a non-empty set and the *interpretation function* $\cdot^{\mathcal{I}}$ maps each concept name $A \in \mathbf{N}_C$ to a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, each role name $r \in \mathbf{N}_R$ to a binary relation $r^{\mathcal{I}}$ on $\Delta^{\mathcal{I}}$ and each individual name $a \in \mathbf{N}_I$ to an individual $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. The extension of $\cdot^{\mathcal{I}}$ to arbitrary concepts is defined inductively as shown in the third column of Table 1.

The semantics of *ALC-ontologies* is now defined as follows. An interpretation \mathcal{I} *satisfies* a concept implication $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, and \mathcal{I} is a *model* of an ontology \mathcal{O} if it satisfies all implications in \mathcal{O} . An ontology is *consistent* if it has a model. A concept C is *satisfiable w.r.t. an ontology* \mathcal{O} if there exists a model \mathcal{I} of \mathcal{O} such that $C^{\mathcal{I}} \neq \emptyset$. A concept C is *subsumed by a concept* D *w.r.t. an ontology* \mathcal{O} (written $\mathcal{O} \models C \sqsubseteq D$) if every model \mathcal{I} of \mathcal{O} satisfies the implication $C \sqsubseteq D$.

Now for the semantics of ABoxes and knowledge bases. An interpretation \mathcal{I} *satisfies* an assertion $C(a)$ 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 knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{A})$ if it is a model of both \mathcal{O} and \mathcal{A} . We say that \mathcal{A} is *consistent* if it has a model, and likewise for \mathcal{K} .

When working with ontologies and without ABoxes, the most relevant way of querying is *subsumption*: given an ontology \mathcal{O} and concepts C, D , check whether $\mathcal{O} \models C \sqsubseteq D$. In the presence of ABoxes, there are two prominent ways of querying: instance checking and conjunctive query answering; see, e.g., [24, 12, 11]. *Instance checking* means, given a knowledge base \mathcal{K} and an assertion $C(a)$, to check whether each model of \mathcal{K} satisfies $C(a)$. If this is the case, we write $\mathcal{K} \models C(a)$.

Example 1. Let \mathcal{O} be a geographical ontological defined as

$$\mathcal{O} = \{\text{European_Country} \equiv \text{Country} \sqcap \text{Located_in_Europe}\}$$

and \mathcal{A} an ABox defined as

$$\mathcal{A} = \{\text{Country}(\text{France}), \text{Country}(\text{Columbia}), \dots, \text{Located_in_Europe}(\text{France}), \dots\}.$$

Then $(\mathcal{O}, \mathcal{A}) \models \text{European_Country}(\text{France})$.

To discuss conjunctive query answering, we need a few preliminaries. An *atom* is of the form $C(v)$ or $r(v, v')$, where v, v' are from a *set of variables*

\mathbb{N}_V , C is a concept and r is a role name. An \mathcal{ALC} -conjunctive query is an expression of the form $\exists \mathbf{v}.\varphi(\mathbf{u}, \mathbf{v})$, where \mathbf{v} and \mathbf{u} are disjoint sequences of variables and φ is a conjunction of atoms using only variables from $\mathbf{v} \cup \mathbf{u}$ (we confuse vectors and sets when convenient). The *arity* of such a query is the length of \mathbf{u} . The variables in \mathbf{u} are the *answer variables* of q , and the ones in \mathbf{v} are the (existentially) *quantified variables*. Let \mathcal{K} be a knowledge base and $q = \exists \mathbf{v}.\varphi(\mathbf{u}, \mathbf{v})$ an n -ary conjunctive query. Then a sequence \mathbf{a} of individual names of length n is a *certain answer* to \mathcal{K} and q if for every model \mathcal{I} of \mathcal{K} , there is a mapping $\pi : \mathbf{v} \cup \mathbf{u} \rightarrow \Delta^{\mathcal{I}}$ such that

- if v is the i -th element of \mathbf{u} and a the i -th element of \mathbf{a} , then $\pi(v) = a^{\mathcal{I}}$;
- $C(v) \in \varphi$ implies $\pi(v) \in C^{\mathcal{I}}$, and $r(v, v') \in \varphi$ implies $(\pi(v), \pi(v')) \in r^{\mathcal{I}}$.

Then, *conjunctive query answering* means to compute, given a knowledge base \mathcal{K} and conjunctive query q , all certain answers to \mathcal{K} and q .

For our purposes, it usually suffices to work with an *instantiated conjunctive query* $\exists \mathbf{v}.\varphi(\mathbf{a}, \mathbf{v})$, in which the answer variables have been replaced with individual names. We write $\mathcal{K} \models \exists \mathbf{v}.\varphi(\mathbf{a}, \mathbf{v})$ if \mathbf{a} is a certain answer to the query $\exists \mathbf{v}.\varphi(\mathbf{u}, \mathbf{v})$.

2.2 First- and second-order Logic

We use standard notation for first- and second-order logic. Throughout the paper, we admit individual constants, truth constants \top and \perp , a binary equality symbol ‘=’ and an arbitrary number of predicates of any arity. Function symbols are not admitted. A *first-order ontology*, or \mathcal{FO} -ontology for short, is simply a finite set of first-order sentences. As usual, we write $\mathcal{O} \models \varphi$ if a first-order sentence φ is a consequence of an \mathcal{FO} -ontology \mathcal{O} .

We will often view an \mathcal{ALC} -ontology as an \mathcal{FO} -ontology. Indeed, it is well-known that most DLs such as \mathcal{ALC} can be conceived as (decidable) fragments of first-order logic [3]. Note that a DL interpretation is just a first-order interpretation restricted to only unary and binary predicates and constants. Then, (i) concepts correspond to formulas in one free variable, (ii) concept implications and ABox assertions correspond to sentences and (iii) ontologies, ABoxes and knowledge bases correspond to first-order theories.

In what follows, we use C^\sharp to denote the standard translation of an \mathcal{ALC} -concept C into an \mathcal{FO} -formula with one free variable x ; see [3]. Thus, we have $A^\sharp = A(x)$ for every concept name A and, inductively,

$$\begin{aligned} \top^\sharp &= x = x \\ \perp^\sharp &= \neg(x = x) \\ (C_1 \sqcap C_2)^\sharp &= C_1^\sharp \wedge C_2^\sharp \\ (\neg C)^\sharp &= \neg C^\sharp \\ (\exists r.C)^\sharp &= \exists y (r(x, y) \wedge C^\sharp[x/y]) \end{aligned}$$

where, in the last clause, y is a fresh variable. Then we can translate an ontology \mathcal{O} into a corresponding \mathcal{FO} -ontology

$$\mathcal{O}^\# := \{\forall x (C^\#(x) \rightarrow D^\#(x)) \mid C \sqsubseteq D \in \mathcal{O}\}.$$

Thus, subsumption in \mathcal{ALC} can be understood in terms of \mathcal{FO} -consequence: for all \mathcal{ALC} -concepts C, D , we have $\mathcal{O} \models C \sqsubseteq D$ iff $\mathcal{O}^\# \models \forall x.(C^\# \rightarrow D^\#)$. We can translate a knowledge base \mathcal{K} into an \mathcal{FO} -ontology $\mathcal{K}^\#$ in a similar way, using individual constants. Then, instance checking and checking whether a tuple \mathbf{a} is a certain answer to a conjunctive query can also be understood as first order consequence.

Example 2. Let

$$\mathcal{O} = \{\text{Father} \equiv \text{Male} \sqcap \exists \text{has_child}.\top\}$$

define a father as a male who has a child. Then

$$\mathcal{O}^\# = \{\forall x (\text{Father}(x) \leftrightarrow (\text{Male}(x) \wedge \exists y \text{ has_child}(x, y)))\}.$$

Second-order logic extends first-order logic by means of quantification over variables P for sets and relations. An \mathcal{SO} -ontology \mathcal{O} is a finite set of \mathcal{SO} -sentences. We write $\mathcal{O} \models \varphi$ if a second-order sentence φ follows from an \mathcal{SO} -ontology \mathcal{O} . Clearly, every \mathcal{FO} -ontology is an \mathcal{SO} -ontology as well.

2.3 Signatures

The notion of a signature plays an important role in this paper. In a nutshell, a *signature* is a finite set of extra-logical symbols, i.e., symbols whose interpretation is not fixed a priori by the semantics. In the context of DLs, a signature may contain concept names, role names and individual names. Logical symbols such as the truth constants \perp, \top and the Boolean operators \sqcap and \neg are not part of a signature. In the context of first- and second-order logic, a signature consists of predicate symbols (except equality) and individual constants. The equality symbol is not included because its interpretation is fixed a priori.

The *signature* $\text{sig}(\mathcal{O})$ of an \mathcal{ALC} -ontology \mathcal{O} is the set of concept and role names that occur in \mathcal{O} , and likewise for the signature $\text{sig}(C)$ of a concept C and $\text{sig}(C \sqsubseteq D)$ of a concept inclusion $C \sqsubseteq D$. For example,

$$\text{sig}(\{\top \sqsubseteq \exists r.B \sqcap \forall r.\perp\}) = \{B, r\}.$$

The signature $\text{sig}(\alpha)$ of an ABox assertion α , $\text{sig}(\mathcal{A})$ of an ABox \mathcal{A} and $\text{sig}(\mathcal{K})$ of a knowledge base \mathcal{K} is defined similarly, but additionally includes all occurring individual names.

The *signature* $\text{sig}(\varphi)$ of a first- or second-order formula φ is the set of all predicate and constant symbols (except equality) used in φ . Note that $\text{sig}(\forall P.\varphi) = \text{sig}(\exists P.\varphi) = \text{sig}(\varphi)$ for every \mathcal{SO} -formula φ . This notion is lifted

Name	Syntax	Semantics	Identifier
number restrictions	$(\leq n r C)$	$\{d \mid \#\{e \mid (d, e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\} \leq n\}$	\mathcal{Q}
	$(\geq n r C)$	$\{d \mid \#\{e \mid (d, e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\} \geq n\}$	
nominals	$\{a\}$	$\{a^{\mathcal{I}}\}$	\mathcal{O}
inverse role	r^{-}	$(r^{\mathcal{I}})^{-1}$	\mathcal{I}
universal role	u	$\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$	\mathcal{U}
role inclusions	$r \sqsubseteq s$	$r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$	\mathcal{H}

Table 2. Additional constructors: syntax and semantics.

to ontologies in the obvious way. For an \mathcal{SO} -sentence φ and a relation symbol S we sometimes write $\exists S.\varphi$ instead of $\exists P.\varphi[P/S]$, where P is a variable for relations of the same arity as S and $\varphi[P/S]$ results from φ by replacing S by P . Clearly, $\text{sig}(\exists S.\varphi) = \text{sig}(\varphi) \setminus \{S\}$.

In this paper, we are often interested in concepts and ontologies that are formulated using a specific signature. Therefore, we talk of an S -ontology \mathcal{O} if $\text{sig}(\mathcal{O}) \subseteq S$, and likewise for S -concepts, etc. When we want to emphasise both the DL \mathcal{L} in which an ontology is formulated and the signature S , we talk about \mathcal{L}_S -ontologies.

2.4 Some Extensions of \mathcal{ALC}

We introduce here the most important extensions of \mathcal{ALC} used in this paper. Some additional extensions (and fragments) are introduced as needed. The extensions considered here fall into three categories: (i) additional concept constructors, (ii) additional roles and (iii) additional statements in TBoxes. The extensions are listed in Table 2, where $\#X$ denotes the size of a set X and double horizontal lines mark the border between extensions of type (i), (ii) and (iii), from top to bottom. The last column lists the identifier for each extension, which is simply appended to the name \mathcal{ALC} . For example, \mathcal{ALC} extended with number restrictions and inverse roles is denoted by \mathcal{ALCQI} and the extension of a DL \mathcal{L} with the universal role is denoted by \mathcal{LU} .

In the following, we only give some remarks regarding the listed extensions and refer the reader to the DL literature for more details [3]. When inverse roles are present, they can be used inside existential and universal restrictions, and also inside number restrictions (if present) and role inclusions (if present). In contrast, the universal role u is only allowed inside existential and universal restrictions. We remark that the universal role is less common than the other extensions of \mathcal{ALC} , but it will play a prominent role in this paper.

Some care is required for defining the signature in extensions of \mathcal{ALC} . If we work with a description logic \mathcal{L} that includes nominals, then signatures (of concepts, ontologies, ABoxes, etc.) include also the individual names a occurring as a nominal $\{a\}$. For example, the signature of the \mathcal{ALCO} -ABox

$\{(A \sqcap \exists r. \{a\})(b)\}$ is $\{A, r, a, b\}$. In contrast to nominals, the universal role is regarded as a logical symbol and is not part of the signature. This is justified by the fact that the interpretation of the universal role u is fixed a priori. Note also that for its translation $(\exists u.D)^\# = \exists x.D^\#$ to first-order logic, no relation symbol is required.

3 Inseparability and Conservative Extensions

We lay the foundation of the general framework that we use to study modularity. This framework consists of conventions that fix (in a rather liberal way) what an ontology, an ontology language, a query, and a query language is, and of the fundamental notions that we use to define modularity: S -inseparability and S -conservative extensions. We also introduce the relevant decision problems associated with S -inseparability and S -conservative extensions.

3.1 Basic Notions and Conventions

For us, an *ontology* is a finite set of second-order sentences, an *ontology language* is a (commonly infinite) set of second-order sentences, a *query* is a second-order sentence, and a *query language* is a (finite or infinite) set of queries. By the first-order translations given in Section 2, this view captures ontologies and query languages based on DLs. Observe that ontology languages and query languages are defined in the same way, and we will often make use of that. We now define the core notions for defining modularity.

Definition 2 (Inseparability and Conservative Extension). Let \mathcal{QL} be a query language, $\mathcal{O}_1, \mathcal{O}_2$ ontologies and S a signature. We say that

- \mathcal{O}_1 and \mathcal{O}_2 are S -inseparable w.r.t. \mathcal{QL} and write $\mathcal{O}_1 \approx_S^{\mathcal{QL}} \mathcal{O}_2$ iff for all $\varphi \in \mathcal{QL}$ with $\text{sig}(\varphi) \subseteq S$, we have $\mathcal{O}_1 \models \varphi$ iff $\mathcal{O}_2 \models \varphi$.
- \mathcal{O}_2 is an S -conservative extension of \mathcal{O}_1 w.r.t. \mathcal{QL} iff $\mathcal{O}_2 \supseteq \mathcal{O}_1$ and \mathcal{O}_1 and \mathcal{O}_2 are S -inseparable w.r.t. \mathcal{QL} . If, in addition, $S = \text{sig}(\mathcal{O}_1)$, then we say that \mathcal{O}_2 is a *conservative extension* of \mathcal{O}_1 w.r.t. \mathcal{QL} .

We say that $\varphi \in \mathcal{QL}$ *separates* \mathcal{O}_1 and \mathcal{O}_2 iff $\mathcal{O}_1 \models \varphi$ and $\mathcal{O}_2 \not\models \varphi$ or vice versa.

Observe that if $\mathcal{O}_1 \subseteq \mathcal{O}_2$ are formulated in first-order logic and \mathcal{QL} consists of all first-order sentences, then our definition of a conservative extension w.r.t. \mathcal{QL} coincides with the standard definition of a conservative extension used in mathematical logic [14].

For any query language \mathcal{QL} and any signature S , the relation $\approx_S^{\mathcal{QL}}$ of S -inseparability w.r.t. \mathcal{QL} is clearly an equivalence relation. Moreover, the following two implications are easily seen to hold:

1. if $\mathcal{O}_1 \approx_S^{\mathcal{QL}} \mathcal{O}_2$ and $S' \subseteq S$, then $\mathcal{O}_1 \approx_{S'}^{\mathcal{QL}} \mathcal{O}_2$;

2. if $\mathcal{O}_1 \approx_S^{\mathcal{QL}} \mathcal{O}_2$ and $\mathcal{QL}' \subseteq \mathcal{QL}$, then $\mathcal{O}_1 \approx_S^{\mathcal{QL}'} \mathcal{O}_2$.

The largest query language \mathcal{QL} considered in this paper is \mathcal{QL}_{SO} , the set of sentences of second-order logic. It follows from Point 2 above that inseparability w.r.t. \mathcal{QL}_{SO} implies inseparability w.r.t. any other query language that fits into our framework.

3.2 Examples

We give several examples that illustrate the importance of S -inseparability and conservative extensions for handling ontologies in general and for defining notions of modularity in particular. For simplicity, we concentrate on the rather expressive query language \mathcal{QL}_{SO} for now and defer the introduction of DL-based query languages to the subsequent section.

Example 3. The following ontology defines the binary relation P (‘part of’) as a transitive and reflexive relation:

$$\mathcal{O}_1 = \{ \forall x P(x, x), \\ \forall x \forall y \forall z (P(x, y) \wedge P(y, z) \rightarrow P(x, z)) \}$$

We now define another ontology that also defines P, but which does this in terms of another binary relation PP (‘proper part of’) that is transitive and irreflexive:

$$\mathcal{O}_2 = \{ \forall x \neg PP(x, x), \\ \forall x \forall y \forall z (PP(x, y) \wedge PP(y, z) \rightarrow PP(x, z)), \\ \forall x \forall y (P(x, y) \leftrightarrow (x = y \vee PP(x, y))) \}.$$

Suppose that the ontology \mathcal{O}_2 is used in an application that refers only the predicate P (but not PP) and is based on the query language \mathcal{QL} . Can we replace \mathcal{O}_2 with the somewhat simpler ontology \mathcal{O}_1 without causing any changes or even corruption? In our framework, answering this question means checking whether or not \mathcal{O}_1 and \mathcal{O}_2 are $\{P\}$ -inseparable w.r.t. \mathcal{QL}_{SO} . The answer is ‘no’: the \mathcal{QL}_{SO} -sentence

$$\varphi = \forall x \forall y ((P(x, y) \wedge P(y, x)) \rightarrow x = y)$$

separates \mathcal{O}_1 and \mathcal{O}_2 since $\mathcal{O}_1 \not\models \varphi$ and $\mathcal{O}_2 \models \varphi$. While \mathcal{O}_1 and \mathcal{O}_2 are toy ontologies, questions of this kind are of obvious importance for real applications. To conclude this example, we remark that \mathcal{O}_1 and \mathcal{O}_2 become $\{P\}$ -inseparable w.r.t. \mathcal{QL}_{SO} by adding φ to \mathcal{O}_1 .

Example 4. According to the introduction, a module should function independently from the containing ontology and conservative extensions can capture this. Thus, let us use conservative extensions to give an example definition of a module.

Definition 3. Let $\mathcal{O}_1 \subseteq \mathcal{O}_2$ be ontologies, S a signature, and \mathcal{QL} a query language. Then \mathcal{O}_1 is a weak S -module of \mathcal{O}_2 w.r.t. \mathcal{QL} if \mathcal{O}_2 is an S -conservative extension of \mathcal{O}_1 w.r.t. \mathcal{QL} .

As a concrete example, consider the \mathcal{ALC} -ontology

$$\mathcal{O}_1 = \{ \text{Male} \equiv \text{Human} \sqcap \neg \text{Female}, \\ \text{Human} \sqsubseteq \forall \text{has_child}.\text{Human} \}$$

and its extension

$$\mathcal{O}_2 = \mathcal{O}_1 \cup \{ \text{Father} \equiv \text{Male} \sqcap \exists \text{has_child}.\top \}.$$

If $S = \text{sig}(\mathcal{O}_1)$, then \mathcal{O}_1 is a weak S -module of \mathcal{O}_2 .

Example 5. Consider again the ontologies \mathcal{O}_1 and \mathcal{O}_2 from Example 4. The extension of \mathcal{O}_1 to \mathcal{O}_2 is simple, but a rather typical case in practice: one or more newly introduced concept names are defined in terms of concept and role names that are already defined in the existing ontology. The existing concept and role names are only *used*, but not affected in any way by this extension. In particular, no new consequences referring only to previously existing concept and role names is derivable, and thus the extension is conservative. Extensions of this simple kind are called *definitorial*.

The notion of a module defined in Definition 3 is much more liberal than this: for example, it allows the extension to make statements about existing symbols as long as these statements are already entailed by the existing ontology. As another example for the definition of a module, we give a stronger one that is closer to the idea of definitorial extensions:

Definition 4. Let $\mathcal{O}_1 \subseteq \mathcal{O}_2$ be ontologies, S a signature and \mathcal{QL} a query language. Then \mathcal{O}_1 is a strong S -module of \mathcal{O}_2 w.r.t. \mathcal{QL} if it is a weak S -module of \mathcal{O}_2 w.r.t. \mathcal{QL} and, in addition, $\mathcal{O}_2 \setminus \mathcal{O}_1$ and the empty ontology are S -inseparable w.r.t. \mathcal{QL} .

To see an example that illustrates the difference between Definitions 3 and 4, let

$$\mathcal{O}'_2 = \mathcal{O}_1 \cup \{ \text{Father} \sqsubseteq \text{Human}, \text{Father} \equiv \text{Male} \sqcap \exists \text{has_child}.\top \}.$$

Let $S = \text{sig}(\mathcal{O}_1)$. Then \mathcal{O}_1 is not a strong S -module of \mathcal{O}'_2 because

$$\mathcal{O}'_2 \setminus \mathcal{O}_1 \models \text{Male} \sqcap \exists \text{has_child}.\top \sqsubseteq \text{Human}$$

and this inclusion does not follow from the empty ontology. However, we already have $\mathcal{O}_1 \models \text{Male} \sqcap \exists \text{has_child}.\top \sqsubseteq \text{Human}$ and, indeed, it is possible to prove that \mathcal{O}_1 is a weak S -module of \mathcal{O}'_2 .

We do not claim that Definitions 3 and 4 are the only reasonable definitions of a module. As indicated already in the introduction, many subtle variations and even completely different approaches are possible. What is common to

virtually all definitions of a module is that inseparability and conservative extensions play a central role. In this paper, we will not favour a particular definition of a module, but rather study modularity directly in terms of inseparability and conservative extensions. We also remark that there are other interesting applications of inseparability such as ontology versioning and module extraction [31, 29, 28, 15].

3.3 Decision Problems

The notions of inseparability and conservative extension give rise to decision problems in a natural way. Let \mathcal{L} be an ontology language and \mathcal{QL} a query language. Then

- the *S-inseparability problem for $(\mathcal{L}, \mathcal{QL})$* is to decide, given \mathcal{L} -ontologies \mathcal{O}_1 and \mathcal{O}_2 and a signature S , whether $\mathcal{O}_1 \approx_S^{\mathcal{QL}} \mathcal{O}_2$;
- the *S-conservativity problem for $(\mathcal{L}, \mathcal{QL})$* is to decide, given \mathcal{L} -ontologies \mathcal{O}_1 and \mathcal{O}_2 with $\mathcal{O}_1 \subseteq \mathcal{O}_2$ and a signature S , whether $\mathcal{O}_1 \approx_S^{\mathcal{QL}} \mathcal{O}_2$;
- the *conservativity problem for $(\mathcal{L}, \mathcal{QL})$* is to decide, given \mathcal{L} -ontologies \mathcal{O}_1 and \mathcal{O}_2 with $\mathcal{O}_1 \subseteq \mathcal{O}_2$ whether $\mathcal{O}_1 \approx_S^{\mathcal{QL}} \mathcal{O}_2$ with $S = \text{sig}(\mathcal{O}_1)$;

In view of the examples given in this section, the utility of these problems should be clear: they can be used to decide whether a given part of an ontology is a module, whether two ontologies can be exchanged in some application, whether an extension to an ontology that is intended to be definitorial has damaged the terms that were already defined, etc.

We will discuss these problems in more detail when surveying the options for query languages in Section 5. For now, we only remark that, in the special case where $\mathcal{L} \subseteq \mathcal{QL}$ and $S \supseteq \text{sig}(\mathcal{O}_1 \cup \mathcal{O}_2)$, it is not hard to see that

$$\mathcal{O}_1 \approx_S^{\mathcal{QL}} \mathcal{O}_2 \text{ iff } \mathcal{O}_1 \models \varphi \text{ for all } \varphi \in \mathcal{O}_2 \text{ and } \mathcal{O}_2 \models \varphi \text{ for all } \varphi \in \mathcal{O}_1.$$

It follows that in this particular case checking *S*-inseparability reduces to logical entailment and thus to standard reasoning in \mathcal{QL} .

4 Robustness Properties

We introduce the three robustness properties for modularity. Since we study modularity in terms of inseparability, we also formulate these properties in terms of inseparability.

Definition 5 (Robustness Properties). Let \mathcal{L} be an ontology language and \mathcal{QL} a query language. Then $(\mathcal{L}, \mathcal{QL})$ is *robust*

- *under vocabulary extensions* if, for all \mathcal{L} -ontologies \mathcal{O}_1 and \mathcal{O}_2 and signatures S, S' with $S' \cap \text{sig}(\mathcal{O}_1 \cup \mathcal{O}_2) \subseteq S$, the following holds:

$$\mathcal{O}_1 \approx_S^{\mathcal{QL}} \mathcal{O}_2 \quad \Rightarrow \quad \mathcal{O}_1 \approx_{S'}^{\mathcal{QL}} \mathcal{O}_2;$$

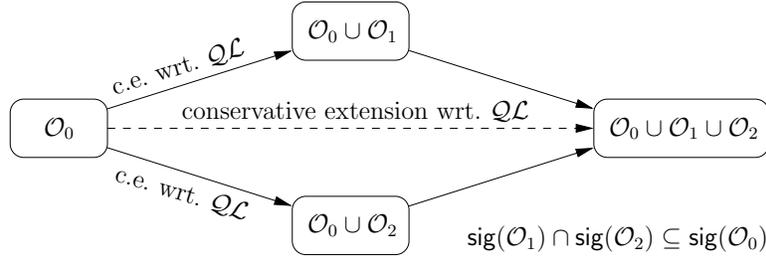


Fig. 1. Robustness under joins.

- *under joins* if, for all \mathcal{L} -ontologies \mathcal{O}_1 and \mathcal{O}_2 and signatures S with $\text{sig}(\mathcal{O}_1) \cap \text{sig}(\mathcal{O}_2) \subseteq S$, the following holds for $i = 1, 2$:

$$\mathcal{O}_1 \approx_S^{\mathcal{QL}} \mathcal{O}_2 \Rightarrow \mathcal{O}_i \approx_S^{\mathcal{QL}} \mathcal{O}_1 \cup \mathcal{O}_2;$$

- *under replacement* if, for all \mathcal{L} -ontologies \mathcal{O}_1 , \mathcal{O}_2 and \mathcal{O} and signatures S with $\text{sig}(\mathcal{O}) \cap \text{sig}(\mathcal{O}_1 \cup \mathcal{O}_2) \subseteq S$, the following holds:

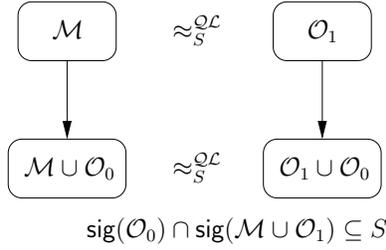
$$\mathcal{O}_1 \approx_S^{\mathcal{QL}} \mathcal{O}_2 \Rightarrow \mathcal{O}_1 \cup \mathcal{O} \approx_S^{\mathcal{QL}} \mathcal{O}_2 \cup \mathcal{O}.$$

In the remainder of this section, we give examples that motivate the usefulness of the properties given in Definition 5.

Robustness under vocabulary extensions. In practice, most ontologies constantly evolve: they are regularly being extended, updated, corrected, etc. This evolution usually results in frequent modifications of the ontology's signature. For this reason, inseparability should be robust under changes of the signature, and this is what robustness under vocabulary extensions is about. In many applications, being inseparable or a conservative extension is of doubtful relevance when a simple addition of fresh symbols to the signature (i.e., symbols that are not mentioned in any of the involved ontologies at all) can change this situation.

Robustness under joins. Taking the union of two or more ontologies is an essential and frequently used operation. For example, the extension of an existing ontology with new concept inclusions can be conceived as a union, and also the import of an existing ontology into a newly constructed one. Robustness under joins is concerned with the behaviour of inseparability regarding this operation. Its virtue is probably best appreciated by considering the following consequence of robustness under joins, formulated in terms of conservative extensions:

- (*) for all \mathcal{L} -ontologies \mathcal{O}_0 , \mathcal{O}_1 , \mathcal{O}_2 , if $\mathcal{O}_0 \cup \mathcal{O}_1$ and $\mathcal{O}_0 \cup \mathcal{O}_2$ are conservative extensions of \mathcal{O}_0 w.r.t. \mathcal{QL} and $\text{sig}(\mathcal{O}_1) \cap \text{sig}(\mathcal{O}_2) \subseteq \text{sig}(\mathcal{O}_0)$, then $\mathcal{O}_0 \cup \mathcal{O}_1 \cup \mathcal{O}_2$ is a conservative extension of \mathcal{O}_0 w.r.t. \mathcal{QL} .

**Fig. 2.** Robustness under replacement.

To see why (*) is useful, suppose that two (or more) ontology designers simultaneously extend an ontology \mathcal{O}_0 , but work on different parts. They produce new sets of concept inclusions \mathcal{O}_1 and \mathcal{O}_2 to be added to \mathcal{O}_0 and ensure that $\mathcal{O}_0 \cup \mathcal{O}_i$ is a conservative extension of \mathcal{O}_0 w.r.t. the query language \mathcal{QL} that is used in the intended application, for $i \in \{1, 2\}$. Now, robustness under joins guarantees (via (*)) that the joint extension $\mathcal{O}_0 \cup \mathcal{O}_1 \cup \mathcal{O}_2$ is also a conservative extension of \mathcal{O}_0 w.r.t. \mathcal{QL} , as illustrated in Figure 1. In the sketched situation, robustness under joins thus ensures that the two designers can work truly independently, i.e., they will not be forced to redesign \mathcal{O}_1 and \mathcal{O}_2 because of unexpected interactions.

Robustness under replacement. This property is critical when a module in an ontology is supposed to be *re-used* in another ontology. We consider two example scenarios.

Assume that an ontology designer develops an ontology \mathcal{O}_0 for hospital administration and wants to reuse a set S of medical terms from a medical ontology \mathcal{O}_1 . A typical example for \mathcal{O}_1 would be SNOMED CT, the Systematized Nomenclature of Medicine, Clinical Terms [46]. The designer knows that queries to his ontology will be formulated in a query language \mathcal{QL} . Instead of importing the whole ontology \mathcal{O}_1 , which potentially is very large (SNOMED CT comprises ~ 0.4 million medical terms), he chooses to import a weak S -module \mathcal{M} of \mathcal{O}_1 w.r.t. \mathcal{QL} , i.e., $\mathcal{M} \subseteq \mathcal{O}_1$ and $\mathcal{M} \approx_S^{\mathcal{QL}} \mathcal{O}_1$. If $(\mathcal{L}, \mathcal{QL})$ is robust under replacement, where \mathcal{L} is the language in which \mathcal{O}_0 and \mathcal{O}_1 are formulated, then it follows that

$$\mathcal{M} \cup \mathcal{O}_0 \approx_S^{\mathcal{QL}} \mathcal{O}_1 \cup \mathcal{O}_0.$$

It is thus indeed possible to use \mathcal{M} instead of \mathcal{O}_1 in \mathcal{O}_0 without losing consequences for the signature S in \mathcal{QL} ; observe that this does not follow from the definition of a weak module alone. This situation is illustrated in Figure 2. Observe that the argument does not depend on the details of \mathcal{O}_0 and does not break when \mathcal{O}_0 evolves.

For a second example, we consider the same scenario. Assume now that the designer of the ontology \mathcal{O}_0 for hospital administration imports the whole ontology \mathcal{O}_1 to reuse the set S of medical terms. He designs \mathcal{O}_0 such that the

terms from S as defined in \mathcal{O}_1 are not corrupted, formalised by $\mathcal{O}_0 \cup \mathcal{O}_1 \approx_S^{\mathcal{QL}} \mathcal{O}_1$. Also assume that the ontology \mathcal{O}_1 is updated frequently. The designer wants to ensure that, when an old version of \mathcal{O}_1 is replaced with a new one, the terms from S as given in the new version of \mathcal{O}_1 are still not corrupted. Since he cannot foresee the changes to \mathcal{O}_1 that will occur in the future, he needs to design \mathcal{O}_0 such that $\mathcal{O}_0 \cup \mathcal{O}_1 \approx_S^{\mathcal{QL}} \mathcal{O}_1$ for all ontologies \mathcal{O}_1 with $\text{sig}(\mathcal{O}_0) \cap \text{sig}(\mathcal{O}_1) \subseteq S$. If $(\mathcal{L}, \mathcal{QL})$ is robust under replacement, this is easy: it simply suffices to ensure that $\mathcal{O}_0 \approx_{\mathcal{L}}^{\mathcal{QL}} \emptyset$. This use case has been discussed in more detail in [17].

The importance of robustness under replacement has first been observed in [16, 15], see also [17]. It also plays an important role in the context of the inseparability of programs in logic programming and answer set programming [35, 20]. Finally, we make a note on the interplay between robustness under replacements and strong modules as introduced in Definition 4: if $(\mathcal{L}, \mathcal{QL})$ enjoys robustness under replacements and in the realistic case that $\text{sig}(\mathcal{O}_2 \setminus \mathcal{O}_1) \cap \text{sig}(\mathcal{O}_1) \subseteq S$, we have that \mathcal{O}_1 is a strong S -module of \mathcal{O}_2 w.r.t. \mathcal{QL} iff $\mathcal{O}_2 \setminus \mathcal{O}_1$ and the empty ontology are S -inseparable w.r.t. \mathcal{QL} , i.e., the condition that \mathcal{O}_1 is a weak module of \mathcal{O}_2 is redundant.

5 Query languages, Robustness and Complexity

So far, we have used the powerful query language \mathcal{QL}_{SO} . For many applications, this is inappropriate because \mathcal{QL}_{SO} can separate ontologies for which no differences are observable in the application. In this section, we introduce a number of DL-based query languages used in applications and discuss the corresponding notions of S -inseparability and S -conservative extensions, as well as their robustness properties and the complexity of the decision problems given in Section 3.3. We start with rather weak query languages and gradually move towards more expressive ones including \mathcal{QL}_{FO} and \mathcal{QL}_{SO} .

5.1 Inconsistency

A very weak query language \mathcal{QL}_{\perp} can be defined as $\{\top \sqsubseteq \perp\}$, i.e., \mathcal{QL}_{\perp} consists of an unsatisfiable first-order sentence. Clearly, $\mathcal{O} \models \top \sqsubseteq \perp$ iff \mathcal{O} is inconsistent. Thus, two ontologies \mathcal{O}_1 and \mathcal{O}_2 are S -inseparable w.r.t. \mathcal{QL}_{\perp} if either both \mathcal{O}_1 and \mathcal{O}_2 are inconsistent, or both are consistent. Observe that S -inseparability w.r.t. \mathcal{QL}_{\perp} does not depend on the actual signature S .

The query language \mathcal{QL}_{\perp} is too weak for defining a reasonable notion of modularity. However, \mathcal{QL}_{\perp} can be used to ensure that the modification of an ontology has not caused inconsistency: the extension \mathcal{O}_2 of an ontology \mathcal{O}_1 by a set of sentences \mathcal{O} does not cause inconsistency iff \mathcal{O}_2 is a conservative extension of \mathcal{O}_1 w.r.t. \mathcal{QL}_{\perp} .

For every ontology language \mathcal{L} , $(\mathcal{L}, \mathcal{QL}_{\perp})$ is robust under vocabulary extensions because S -inseparability does not depend on S . It is robust neither

under joins nor replacement, for any of the ontology languages introduced in Section 2. Let, for example, $\mathcal{O}_1 = \{A \equiv \top\}$ and $\mathcal{O}_2 = \{A \equiv \perp\}$. Then $\mathcal{O}_1 \approx^{\mathcal{QL}_\perp} \mathcal{O}_2$, but $\mathcal{O}_1 \not\approx^{\mathcal{QL}_\perp} \mathcal{O}_1 \cup \mathcal{O}_2$ because \mathcal{O}_1 and \mathcal{O}_2 are consistent, but $\mathcal{O}_1 \cup \mathcal{O}_2$ is inconsistent. Thus, $(\mathcal{ALC}, \mathcal{QL}_\perp)$ is not robust under joins. The decision problems associated with $(\mathcal{L}, \mathcal{QL}_\perp)$ obviously have the same complexity as deciding ontology consistency in \mathcal{L} .

5.2 Subsumption between concept names

The query language \mathcal{QL}_{CN} is defined as the set of all queries $A \sqsubseteq B$, where A and B are concept names or truth constants \perp and \top . Two ontologies \mathcal{O}_1 and \mathcal{O}_2 are S -inseparable w.r.t. \mathcal{QL}_{CN} iff they give raise to the same classification regarding the concept names in S , i.e., the sub-/superclass relation between any two concept names in S is identical.

Using \mathcal{QL}_{CN} as a query language is useful if the application relies only on the classification, i.e., changes in the meaning of a symbol are considered acceptable as long as the classification does not change. Observe that S -inseparability w.r.t. \mathcal{QL}_{CN} implies S -inseparability w.r.t. \mathcal{QL}_\perp , but the converse does not hold whenever $S \neq \emptyset$. Also observe that S -inseparability w.r.t. \mathcal{QL}_{CN} is oblivious to the addition and deletion of role and individual names to and from S .

As illustrated by the following example, the query language \mathcal{QL}_{CN} is still very weak when used to define modularity.

Example 6. Reconsider the ontologies from Example 5:

$$\begin{aligned} \mathcal{O}_1 &= \{\text{Male} \equiv \text{Human} \sqcap \neg \text{Female}, \text{Human} \sqsubseteq \forall \text{has_child}.\text{Human}\} \\ \mathcal{O}'_2 &= \mathcal{O}_1 \cup \{\text{Father} \sqsubseteq \text{Human}, \text{Father} \equiv \text{Male} \sqcap \exists \text{has_child}.\top\}. \end{aligned}$$

Let $S = \text{sig}(\mathcal{O}_1)$. Then \mathcal{O}_1 is a weak S -module because the same implications between concept names in S are derivable from \mathcal{O}_1 and \mathcal{O}'_2 . However, this is true only because we restrict ourselves to subsumption between concept names, as we had said already that $\text{Male} \sqcap \exists \text{has_child}.\top \sqsubseteq \text{Human}$ separates the two ontologies.

It is easy to see that $(\mathcal{L}, \mathcal{QL}_{\text{CN}})$ is robust under vocabulary extensions for any ontology language \mathcal{L} introduced in Section 2. As shown by the next example, robustness under joins and replacement fails.

Example 7. Let $\mathcal{O}_1 = \{A \sqsubseteq \exists r.B\}$, $\mathcal{O}_2 = \{\exists r.B \sqsubseteq E\}$, and $S = \{A, B, r, E\}$. Then $\mathcal{O}_1 \approx_S^{\mathcal{QL}_{\text{CN}}} \mathcal{O}_2$. Failure of robustness under joins follows from $\mathcal{O}_1 \not\approx_S^{\mathcal{QL}_{\text{CN}}} \mathcal{O}_1 \cup \mathcal{O}_2$ since $\mathcal{O}_1 \cup \mathcal{O}_2 \models A \sqsubseteq E$ and $\mathcal{O}_1 \not\models A \sqsubseteq E$.

For failure of robustness under replacement consider $\mathcal{O} = \{A \sqsubseteq \neg \exists r.B\}$. Then $\mathcal{O}_1 \cup \mathcal{O} \models A \sqsubseteq \perp$ but $\mathcal{O}_2 \cup \mathcal{O} \not\models A \sqsubseteq \perp$.

Still, \mathcal{QL}_{CN} has useful applications, e.g., for efficient implementations of classification in a DL reasoner and in distributed description logics, where subsumptions between concept names are usually all that matter. The decision

problems associated with $(\mathcal{L}, \mathcal{QL}_{\text{CN}})$ are not harder than deciding subsumption between concept names in \mathcal{L} .

5.3 Subsumption between complex concepts

For any description logic \mathcal{L} , the query language $\mathcal{QL}_{\mathcal{L}}$ is defined as the set of all queries $C \sqsubseteq D$, where C and D are \mathcal{L} -concepts. If \mathcal{L} admits role inclusions, we also include queries $r \sqsubseteq s$, with r and s roles. Thus, two ontologies \mathcal{O}_1 and \mathcal{O}_2 are S -inseparable w.r.t. $\mathcal{QL}_{\mathcal{L}}$ iff they entail the same subsumptions between \mathcal{L}_S -concepts and the same role inclusions $r \sqsubseteq s$ with $r, s \in S$, if role inclusions are present in \mathcal{L} . The following example shows that even for propositional DLs \mathcal{L} without role names, $\mathcal{QL}_{\mathcal{L}}$ is strictly stronger than \mathcal{QL}_{CN} .

Example 8. Let

$$\mathcal{O}_1 = \emptyset, \quad \mathcal{O}_2 = \{\text{Parent} \sqcap \text{Male} \sqsubseteq \text{Father}\}, \quad S = \{\text{Parent}, \text{Male}, \text{Father}\}.$$

Then \mathcal{O}_2 is an S -conservative extension of \mathcal{O}_1 w.r.t. \mathcal{QL}_{CN} , but $\text{Parent} \sqcap \text{Male} \sqsubseteq \text{Father}$ separates \mathcal{O}_1 and \mathcal{O}_2 w.r.t. $\mathcal{QL}_{\text{ACC}}$.

When choosing different DLs \mathcal{L} , the resulting notions of inseparability usually differ.

Example 9. Let

$$\mathcal{O}_1 = \{\text{Human} \sqsubseteq \exists \text{parent} . \top\}, \quad \mathcal{O}_2 = \{\text{Human} \sqsubseteq \exists \text{parent} . \text{Male} \sqcap \exists \text{parent} . \neg \text{Male}\}$$

and $S = \{\text{Human}, \text{parent}\}$. Then \mathcal{O}_1 and \mathcal{O}_2 are S -inseparable w.r.t. $\mathcal{QL}_{\text{ACC}}$, but $\text{Human} \sqsubseteq \neg(\leq 1 \text{parent} \top)$ S -separates \mathcal{O}_1 and \mathcal{O}_2 w.r.t. $\mathcal{QL}_{\text{ACCQ}}$.

If \mathcal{L} provides all Boolean operators, a useful *alternative definition* of $\mathcal{QL}_{\mathcal{L}}$ is the set of queries $C \sqsubseteq \perp$ with C an \mathcal{L} -concept, i.e., two ontologies \mathcal{O}_1 and \mathcal{O}_2 are S -inseparable w.r.t. $\mathcal{QL}_{\mathcal{L}}$ iff there is no \mathcal{L}_S -concept C that is satisfiable w.r.t. \mathcal{O}_1 but not w.r.t. \mathcal{O}_2 , or vice versa. To see that the definitions are equivalent, observe that $C \sqsubseteq D$ separates \mathcal{O}_1 and \mathcal{O}_2 iff $C \sqcap \neg D \sqsubseteq \perp$ separates \mathcal{O}_1 and \mathcal{O}_2 .

The query languages $\mathcal{QL}_{\mathcal{L}}$ are often appropriate during the design process of an ontology.

Example 10. Assume an ontology designer extends a medical ontology \mathcal{O}_0 written in a DL \mathcal{L} with a set \mathcal{O}_1 of \mathcal{L} -sentences in order to cover some part of medicine, say anatomy, in more detail. He wants to ensure that this operation does not impact the subsumptions between complex concepts built over symbols that are already defined in \mathcal{O}_0 and not related to anatomy. Then, he should ensure that $\mathcal{O}_0 \cup \mathcal{O}_1$ is an S -conservative extension of \mathcal{O}_0 w.r.t. $\mathcal{QL}_{\mathcal{L}}$, where S consists of all symbols from \mathcal{O}_0 that are unrelated to anatomy. Similarly, conservative extensions w.r.t. $\mathcal{QL}_{\mathcal{L}}$ can be used to ensure that *deletions* of sentences from an ontology do not change the subsumptions between concepts in an unexpected way. In the case that existing sentences are modified, the designer should use inseparability instead of conservative extensions.

The query languages $\mathcal{QL}_{\mathcal{L}}$ are powerful enough to define useful notions of modularity, for example via Definitions 3 and 4. In particular, inseparability w.r.t. $\mathcal{QL}_{\mathcal{L}}$ is suited to ensure a relatively strong form of module independence, in contrast to inseparability w.r.t. \mathcal{QL}_{\perp} and \mathcal{QL}_{CN} . We now discuss the formal properties of $\mathcal{QL}_{\mathcal{L}}$ in more detail. We concentrate on giving an overview of available results and thus defer longer proofs to later sections or the appendix.

Robustness Properties

In many cases, $(\mathcal{L}, \mathcal{QL}_{\mathcal{L}})$ is robust under vocabulary extensions and joins. The following result is a consequence of Theorems 15 and 16 in Section 6.2.

Theorem 1. *Let \mathcal{L} be any of the DLs \mathcal{ALC} , \mathcal{ALCQ} , \mathcal{ALCI} , \mathcal{ALCQI} , \mathcal{ALCU} , \mathcal{ALCQU} , \mathcal{ALCTU} , \mathcal{ALCQIU} . Then $(\mathcal{L}, \mathcal{QL}_{\mathcal{L}})$ is robust under vocabulary extensions and joins.*

Thus, standard constructors such as inverse roles and number restrictions do not cause any difficulties as far as robustness under vocabulary extensions and joins is concerned. The situation is different for nominals and role hierarchies.

Proposition 1. *$(\mathcal{ALCO}, \mathcal{QL}_{\mathcal{ALCO}})$ and $(\mathcal{ALCH}, \mathcal{QL}_{\mathcal{ALCH}})$ are not robust under vocabulary extensions and joins.*

Proof. We start with non-robustness under vocabulary extensions in \mathcal{ALCO} . Let

$$\begin{aligned}\mathcal{O}_1 &= \{\top \sqsubseteq \exists r.\top\}, \\ \mathcal{O}_2 &= \mathcal{O}_1 \cup \{A \sqsubseteq \forall r.\neg A, \neg A \sqsubseteq \forall r.A\}.\end{aligned}$$

Then \mathcal{O}_2 is a conservative extension of \mathcal{O}_1 w.r.t. $\mathcal{QL}_{\mathcal{ALCO}}$, thus $\mathcal{O}_1 \approx_S^{\mathcal{QL}_{\mathcal{ALCO}}} \mathcal{O}_2$ for $S = \{r\}$. Now observe that $\{a\} \sqsubseteq \forall r.\neg\{a\}$ separates the two ontologies w.r.t. $\mathcal{QL}_{\mathcal{ALCO}}$, for any nominal $\{a\}$. Thus, $\mathcal{O}_1 \not\approx_{S'}^{\mathcal{QL}_{\mathcal{ALCO}}} \mathcal{O}_2$ for $S' = S \cup \{a\}$. Observe that the nominal $\{a\}$ has no connection whatsoever with the two ontologies \mathcal{O}_1 and \mathcal{O}_2 .

Now consider \mathcal{ALCH} . Let

$$\begin{aligned}\mathcal{O}_1 &= \{\top \sqsubseteq \forall r_i \forall r_j . \perp \mid i, j = 1, 2\} \cup \{\exists r_1.\top \equiv \exists r_2.\top\}, \\ \mathcal{O}_2 &= \mathcal{O}_1 \cup \{s \sqsubseteq r_1, s \sqsubseteq r_2, \exists r_1.\top \sqsubseteq \exists s.\top\}.\end{aligned}$$

Then \mathcal{O}_2 is a conservative extension of \mathcal{O}_1 w.r.t. $\mathcal{QL}_{\mathcal{ALCH}}$, thus $\mathcal{O}_1 \approx_S^{\mathcal{QL}_{\mathcal{ALCH}}} \mathcal{O}_2$ for $S = \{r_1, r_2\}$. However, $\exists r_1.\top \sqcap \forall r_1.A \sqsubseteq \exists r_2.A$ separates the two ontologies w.r.t. $\mathcal{QL}_{\mathcal{ALCH}}$, where A is a fresh concept name. The proof of non-robustness under joins is similar and deferred to Proposition 7 in Appendix A. \square

In Section 4, we have argued that robustness under replacement is an essential property for most applications of inseparability, such as modularity. Theorem 1 thus indicates that one has to be rather careful when using \mathcal{ALCO} and \mathcal{ALCH} as a query language. In particular, weak and strong modules as in Definition 3 and 4 are inappropriate. However, it is possible to “build in” robustness under vocabulary extensions when defining a module:

Definition 6. Let $\mathcal{O}_1 \subseteq \mathcal{O}_2$ be ontologies, S a signature, and \mathcal{QL} a query language. Then \mathcal{O}_1 is a weak S -module in \mathcal{O}_2 w.r.t. \mathcal{QL} robust under vocabulary extensions if \mathcal{O}_2 is an S' -conservative extension of \mathcal{O}_1 w.r.t. \mathcal{QL} for all S' such that $\text{sig}(\mathcal{O}_1 \cup \mathcal{O}_2) \cap S' \subseteq S$.

In a similar way, it is possible to define modules with robustness under joins built in.

We now turn to robustness under replacement. This property fails for many standard description logics such as the ones in the following result.

Theorem 2. Let \mathcal{L} be any of the DLs \mathcal{ALC} , \mathcal{ALCQ} , \mathcal{ALCI} , \mathcal{ALCQI} , \mathcal{ALCO} , \mathcal{ALCHO} . Then $(\mathcal{L}, \mathcal{QL}_{\mathcal{L}})$ is not robust under replacement.

Proof. Let

$$\mathcal{O}_1 = \emptyset, \quad \mathcal{O}_2 = \{A \sqsubseteq \exists r.B\}, \quad S = \{A, B\}, \quad \mathcal{O} = \{A \equiv \top, B \equiv \perp\}.$$

It is not hard to see that for every first-order sentence φ with $\text{sig}(\varphi) \subseteq S$,

$$\mathcal{O}_2 \models \varphi \text{ iff } \{\exists x A(x) \rightarrow \exists y B(y)\} \models \varphi.$$

However, the \mathcal{FO} -ontology $\{\exists x A(x) \rightarrow \exists y B(y)\}$ has no non-tautological consequences that can be formulated in $\mathcal{QL}_{\mathcal{L}}$ using only symbols from S . Hence, \mathcal{O}_2 is an S -conservative extension of \mathcal{O}_1 w.r.t. $\mathcal{QL}_{\mathcal{L}}$. But then, $\mathcal{O}_1 \cup \mathcal{O} \not\approx_S^{\mathcal{QL}_{\mathcal{L}}} \mathcal{O}_2 \cup \mathcal{O}$ since $\top \sqsubseteq \perp$ separates the two ontologies. \square

In Section 4, we have argued that robustness under vocabulary extensions is needed for the reuse modules. Thus, Theorem 2 indicates that one has to be careful when using the query languages listed in the theorem for this application. For example, modules such as in Definitions 3, 4 and 6 are not appropriate. One approach to fix this is to define a notion of module that has robustness under replacements built in:

Definition 7. Let $\mathcal{O}_1 \subseteq \mathcal{O}_2$ be ontologies, S a signature, and \mathcal{QL} a query language. Then \mathcal{O}_2 is a weak S -module of \mathcal{O}_2 w.r.t. \mathcal{QL} robust under replacement if for all ontologies \mathcal{O} with $\text{sig}(\mathcal{O}) \cap \text{sig}(\mathcal{O}_2) \subseteq S$, $\mathcal{O}_2 \cup \mathcal{O}$ is an S -conservative extension of $\mathcal{O}_1 \cup \mathcal{O}$ w.r.t. \mathcal{QL} .

The next result identifies a second approach to deal with Theorem 2: if robustness under replacement is desired, switch from $(\mathcal{L}, \mathcal{QL}_{\mathcal{L}})$ to $(\mathcal{L}, \mathcal{QL}_{\mathcal{LU}})$. If this is done, robustness under replacements is recovered and it suffices to work with weak modules as in Definition 3. We shall see in Section 6 that the two approaches to deal with Theorem 2 are identical in some rather strong sense. For now, we only show that the addition of a universal role to the query language usually recovers robustness under replacement.

Theorem 3. Let \mathcal{L} be any of the DLs \mathcal{ALC} , \mathcal{ALCI} , \mathcal{ALCQ} , \mathcal{ALCQI} . Then $(\mathcal{LU}, \mathcal{QL}_{\mathcal{LU}})$, and thus also $(\mathcal{L}, \mathcal{QL}_{\mathcal{LU}})$, is robust under replacement.

Proof. Let $\mathcal{O}_1 \approx_S^{\mathcal{QL}_L} \mathcal{O}_2$ and assume that $\mathcal{O}_1 \cup \mathcal{O} \models C_0 \sqsubseteq D_0$, where $\text{sig}(\mathcal{O}) \cap \text{sig}(\mathcal{O}_1 \cup \mathcal{O}_2) \subseteq S$ and $\text{sig}(C_0 \sqsubseteq D_0) \subseteq S$. Let $S' = S \cup \text{sig}(\mathcal{O})$. By robustness under signature extensions, $\mathcal{O}_1 \approx_{S'}^{\mathcal{QL}_L} \mathcal{O}_2$. Clearly

$$\mathcal{O}_1 \models \forall u. \prod_{C \sqsubseteq D \in \mathcal{O}} \neg C \sqcup D \sqsubseteq \forall u. (\neg C_0 \sqcup D_0).$$

Since $\mathcal{O}_1 \approx_{S'}^{\mathcal{QL}_L} \mathcal{O}_2$, \mathcal{O}_2 entails the same subsumption. Thus $\mathcal{O}_2 \cup \mathcal{O} \models C_0 \sqsubseteq D_0$, as required. \square

We present one additional observation regarding robustness under replacement. In the proof of Theorem 2, r is a role that does not occur in S . It turns out that such roles are required to obtain counterexamples to robustness under replacement. The following result is proved in Appendix A.

Theorem 4. *Let \mathcal{L} be any of the DLs \mathcal{ALC} , \mathcal{ALCI} , \mathcal{ALCQ} , \mathcal{ALCQI} , S a signature, and assume that \mathcal{O}_1 and \mathcal{O}_2 contain only roles from S . Then*

$$\mathcal{O}_1 \approx_S^{\mathcal{QL}_L} \mathcal{O}_2 \quad \text{implies} \quad \mathcal{O}_1 \cup \mathcal{O} \approx_S^{\mathcal{QL}_L} \mathcal{O}_2 \cup \mathcal{O},$$

for all \mathcal{L} -ontologies \mathcal{O} with $\text{sig}(\mathcal{O}) \cap \text{sig}(\mathcal{O}_1 \cup \mathcal{O}_2) \subseteq S$.

Decision Problems

We now discuss the computational complexity of deciding inseparability and conservative extensions w.r.t. \mathcal{QL}_L . The following result is due to [22, 37].

Theorem 5. *Let \mathcal{L} be any of the DLs \mathcal{ALC} , \mathcal{ALCI} , \mathcal{ALCQ} and \mathcal{ALCQI} . Then the S -inseparability problem for $(\mathcal{L}, \mathcal{QL}_L)$ is 2-EXPTIME-complete. Moreover, the S -conservativity and conservativity problem are 2-EXPTIME-complete as well.*

We remark that, in [22, 37], the results stated in Theorem 5 are proved for S -conservativity w.r.t. \mathcal{QL}_L for \mathcal{ALC} and \mathcal{ALCQI} only. However, it is not too difficult to extend the proofs to \mathcal{ALCI} and \mathcal{ALCQ} and S -inseparability instead of S -conservativity. For the extensions of the logics \mathcal{L} with the universal role we note the following conjecture.

Conjecture 1. Let \mathcal{L} be any of the DLs \mathcal{ALCU} , \mathcal{ALCIU} , \mathcal{ALCQU} , \mathcal{ALCQIU} . Then the S -inseparability problem for $(\mathcal{L}, \mathcal{QL}_L)$ is 2-EXPTIME-complete.

Interestingly, the addition of nominals to \mathcal{ALCQI} leads to undecidability [37].

Theorem 6. *For $(\mathcal{ALCQIO}, \mathcal{QL}_{\mathcal{ALCQIO}})$, the S -inseparability problem and conservativity problem are undecidable.*

Recall that \mathcal{LO} denotes the extension of a DL \mathcal{L} by nominals. We conjecture that even with nominals, S -inseparability is still 2-EXPTIME-complete for the DLs from above which include \mathcal{ALC} but are strictly below \mathcal{ALCQI} :

Conjecture 2. Let \mathcal{L} be any of the DLs \mathcal{ALC} , \mathcal{ALCI} , \mathcal{ALCQ} or their extension by the universal role. Then the S -inseparability problem for $(\mathcal{LO}, \mathcal{QL}_{\mathcal{LO}})$ is 2-EXPTIME-complete.

Notions of a module such as the ones given in Definitions 6 and 7 give rise to other decision problems than the S -inseparability problem and the conservativity problem. The following result addresses the case of Definition 6. It is proved in [17] for $(\mathcal{ALCO}, \mathcal{QL}_{\mathcal{ALCO}})$. The proof is easily extended to the other listed cases.

Theorem 7. *Let \mathcal{L} be any of the DLs \mathcal{ALC} , \mathcal{ALCI} , \mathcal{ALCQ} or their extension by the universal role. Given two \mathcal{LO} -ontologies \mathcal{O}_1 and \mathcal{O}_2 and a signature S , it is undecidable whether \mathcal{O}_1 is a weak module of \mathcal{O}_2 w.r.t. $\mathcal{QL}_{\mathcal{LO}}$ robust under replacement.*

5.4 ABoxes and Conjunctive Queries

If ontologies are used together with an ABox, inseparability based on subsumption is usually too weak. In this section, we define notions of inseparability based on instance checking and conjunctive query answering.

In principle, there are two ways to include the ABox into our framework: as part of the ontology and as part of the query language. The first option is appropriate if the ontology and ABox are closely coupled, e.g. they are designed and maintained together, and the ABox does not change significantly more often than the ontology. In this case, we should define S -inseparability between *knowledge bases*.

Definition 8. Let \mathcal{L} be a description logic. We call knowledge bases \mathcal{K}_1 and \mathcal{K}_2

- a) S -inseparable w.r.t. \mathcal{L} -instance checking iff, for any \mathcal{L} -assertion $C(a)$ with $\text{sig}(C) \subseteq S$, we have $\mathcal{K}_1 \models C(a)$ iff $\mathcal{K}_2 \models C(a)$.
- b) S -inseparable w.r.t. \mathcal{L} -conjunctive queries iff, for all instantiated \mathcal{L} -conjunctive queries $q = \exists \mathbf{v}.\varphi(\mathbf{v}, \mathbf{a})$ with $\text{sig}(q) \subseteq S$, we have $\mathcal{K}_1 \models q$ iff $\mathcal{K}_2 \models q$.

It is easy to see that the latter implies the former. However, the converse is false. For a simple example, let $\mathcal{L} = \mathcal{ALC}$,

$$\mathcal{O}_1 = \emptyset, \quad \mathcal{O}_2 = \{A \sqsubseteq \exists r.B\}, \quad \mathcal{A}_1 = \mathcal{A}_2 = \{A(a)\}, \quad \text{and } S = \{A, B\}.$$

Then \mathcal{K}_1 and \mathcal{K}_2 are S -inseparable w.r.t. instance checking, but the conjunctive query $\exists x.B(x)$ separates the two knowledge bases. When applied to knowledge bases with an empty ABox, S -inseparability w.r.t. \mathcal{L} -instance checking and (subsumption-based) S -inseparability w.r.t. $\mathcal{QL}_{\mathcal{L}}$ coincide. To see this, note that (i) if $C \sqsubseteq D$ separates \mathcal{K}_1 and \mathcal{K}_2 , then so does $D \sqcup \neg C(a)$, for any $a \in \mathbf{N}_{\mathcal{C}}$; conversely (ii) if $C(a)$ separates \mathcal{K}_1 and \mathcal{K}_2 , then so does $\top \sqsubseteq C$.

It should be clear that the above two notions of inseparability fit into our framework: knowledge bases can be translated into sets of \mathcal{FO} -sentences, and thus can be viewed as ontologies. However, in most applications that use ABoxes, the above approach does not seem appropriate because the ontology (conceptual modelling) and the ABox (actual data) have a different status. In particular, the ABox is usually unknown when the ontology is developed and changes much more frequently than the ontology. This observation suggests that it is useful to consider a notion of inseparability that captures ABoxes as unknown “black boxes” and thus quantifies over *all possible ABoxes*. This corresponds to making the ABox part of the query, instead of the ontology.

Definition 9. Let \mathcal{L} be a description logic. We say that ontologies \mathcal{O}_1 and \mathcal{O}_2 are

- c) *S-inseparable w.r.t. \mathcal{L} -instance checking* iff, for all \mathcal{L} -ABoxes \mathcal{A} with $\text{sig}(\mathcal{A}) \subseteq S$ and \mathcal{L} -assertions $C(a)$ with $\text{sig}(C) \subseteq S$, we have $(\mathcal{O}_1, \mathcal{A}) \models C(a)$ iff $(\mathcal{O}_2, \mathcal{A}) \models C(a)$.
- d) *S-inseparable w.r.t. \mathcal{L} -conjunctive queries* iff, for all \mathcal{L} -ABoxes \mathcal{A} with $\text{sig}(\mathcal{A}) \subseteq S$ and all instantiated \mathcal{L} -conjunctive queries $q = \exists \mathbf{v}.\varphi(\mathbf{v}, \mathbf{a})$, we have $(\mathcal{O}_1, \mathcal{A}) \models q$ iff $(\mathcal{O}_2, \mathcal{A}) \models q$.

Observe that c)-d) apply to ontologies, in contrast to a)-b), which apply to knowledge bases. Again, it is easy to see that if two ontologies are *S-inseparable w.r.t. \mathcal{L} -conjunctive queries*, then they are *S-inseparable w.r.t. \mathcal{L} -instance checking*. The converse fails, with essentially the same argument as for a) and b) above.

Note that *S-inseparability* of two ontologies \mathcal{O}_1 and \mathcal{O}_2 w.r.t. \mathcal{L} -instance checking implies *S-inseparability w.r.t. $\mathcal{QL}_{\mathcal{L}}$* because, as above, if $C \sqsubseteq D$ separates \mathcal{O}_1 and \mathcal{O}_2 , then so does the empty ABox together with the instance query $D \sqcup \neg C(a)$. However, unlike for a), the converse does not hold.

Example 11. Let

$$\mathcal{O}_1 = \emptyset, \quad \mathcal{O}_2 = \mathcal{O}_1 \cup \{A \sqsubseteq \forall r.\neg A, \neg A \sqsubseteq \forall r.A\}, \quad S = \{r\}.$$

Then \mathcal{O}_1 and \mathcal{O}_2 are *S-inseparable w.r.t. $\mathcal{QL}_{\mathcal{ALC}}$* , but $(\mathcal{O}_1, \{r(a, a)\})$ is consistent and $(\mathcal{O}_2, \{r(a, a)\})$ is inconsistent. Thus, the ABox $\mathcal{A} = \{r(a, a)\}$ and assertion $\perp(a)$ separate \mathcal{O}_1 and \mathcal{O}_2 . Note that this example is similar to the counterexample given in the proof of Proposition 1.

S-inseparability w.r.t. \mathcal{L} -instance checking is stronger than *S-inseparability w.r.t. $\mathcal{QL}_{\mathcal{L}}$* because of the availability of the ABox, which allows us to fix a part of the model up to isomorphism.

We show that the notions of inseparability given under c) and d) live inside our framework. In what follows, we use individual names as first-order constants. For an \mathcal{L} -ABox \mathcal{A} , define a corresponding first-order sentence

$$\mathcal{A}^\# := \bigwedge_{C(a) \in \mathcal{A}} C^\#[x/a] \wedge \bigwedge_{r(a,b) \in \mathcal{A}} r(a,b).$$

For an instantiated conjunctive query $\exists \mathbf{v}.\varphi(\mathbf{v}, \mathbf{a})$, define a corresponding first-order sentence q^\sharp by replacing each concept atom $C(a)$ in φ with $C^\sharp[x/a]$.

Definition 10. Let \mathcal{L} be a description logic. Then

- $\mathcal{QL}_{\mathcal{L}}^I$ is the set of first-order sentences $\mathcal{A}^\sharp \rightarrow C^\sharp[x/a]$, where \mathcal{A} is an \mathcal{L} -ABox, C an \mathcal{L} -concept and q an individual name;
- $\mathcal{QL}_{\mathcal{L}}^q$ is the set of first-order sentences $\mathcal{A}^\sharp \rightarrow q^\sharp$, where \mathcal{A} is an \mathcal{L} -ABox and q is an instantiated \mathcal{L} -conjunctive query.

It is easy to see that two ontologies are S -inseparable w.r.t. \mathcal{L} -instance checking in the sense of Definition 9 c) iff they are S -inseparable w.r.t. $\mathcal{QL}_{\mathcal{L}}^I$ and that they are S -inseparable w.r.t. \mathcal{L} -conjunctive queries in the sense of Definition 9 d) iff they are S inseparable w.r.t. $\mathcal{QL}_{\mathcal{L}}^q$.

As indicated by the discussion above, query languages such as $\mathcal{QL}_{\mathcal{L}}^I$ and $\mathcal{QL}_{\mathcal{L}}^q$ induce an even stronger notion of S -inseparability than $\mathcal{QL}_{\mathcal{L}}$. Intuitively, their power is between that of $\mathcal{QL}_{\mathcal{L}}$ and $\mathcal{QL}_{\mathcal{FO}}$, as discussed in Section 5.5 below. For $\mathcal{QL}_{\mathcal{L}}^I$ and $\mathcal{QL}_{\mathcal{L}}^q$, neither the computational complexity of deciding inseparability nor robustness properties have been investigated for description logics extending \mathcal{ALC} . In Section 8, we will discuss a number of results for these query languages for ontologies based on weak DLs such as \mathcal{EL} and DL-Lite.

5.5 Semantic Inseparability, First- and Second-order Queries

The query languages \mathcal{QL} considered so far are all proper fragments of first-order logic. In fact, it is not difficult to show that inseparability w.r.t. $\mathcal{QL}_{\mathcal{FO}}$, the set of all first-order sentences, is stronger than any of them. To see this, consider

$$\mathcal{O}_1 = \emptyset, \quad \mathcal{O}_2 = \{A \sqsubseteq \exists r.(A \sqcap B) \sqcap \exists r.(A \sqcap \neg B)\}, \quad S = \{A\}$$

Then, \mathcal{O}_2 is an S -conservative extension of \mathcal{O}_1 w.r.t. all query languages considered so far, but the first-order sentence $\exists x.A(x) \Rightarrow \exists y.(x \neq y \wedge A(y))$ separates \mathcal{O}_1 and \mathcal{O}_2 .

Theorem 8. *Let \mathcal{L} be a fragment of \mathcal{FO} . Then $(\mathcal{L}, \mathcal{QL}_{\mathcal{FO}})$ is robust under vocabulary extensions, joins and under replacement.*

Proof. Robustness under vocabulary extensions and joins follow from Theorems 15 and 16 in Section 6.2. Robustness under replacement follows from robustness under vocabulary extensions and the fact that first-order logic is closed under Boolean operators: from $\mathcal{O}_1 \approx_S^{\mathcal{FO}} \mathcal{O}_2$, it follows that for all \mathcal{L} -ontologies \mathcal{O} with $\text{sig}(\mathcal{O}) \cap \text{sig}(\mathcal{O}_1 \cup \mathcal{O}_2) \subseteq S$ and \mathcal{FO} -sentences φ with $\text{sig}(\varphi) \subseteq S$, we have

$$\mathcal{O}_1 \models \bigwedge \mathcal{O} \rightarrow \varphi \quad \Leftrightarrow \quad \mathcal{O}_2 \models \bigwedge \mathcal{O} \rightarrow \varphi,$$

which implies

$$\mathcal{O}_1 \cup \mathcal{O} \models \varphi \quad \Leftrightarrow \quad \mathcal{O}_2 \cup \mathcal{O} \models \varphi.$$

□

An even stronger query language than $\mathcal{QL}_{\mathcal{FO}}$ is $\mathcal{QL}_{\mathcal{SO}}$, the set of all second-order sentences. As has already been mentioned, S -inseparability w.r.t. $\mathcal{QL}_{\mathcal{SO}}$ implies inseparability w.r.t. *any* query language allowed in our framework. Interestingly, inseparability w.r.t. $\mathcal{QL}_{\mathcal{SO}}$ is equivalent to a semantic notion of inseparability, which we introduce next. Given a model \mathcal{I} and a signature S , we denote by $\mathcal{I}_{|S}$ the S -reduct of \mathcal{I} ; i.e., $\Delta^{\mathcal{I}_{|S}} = \Delta^{\mathcal{I}}$, $X^{\mathcal{I}_{|S}} = X^{\mathcal{I}}$ for all $X \in S$, and predicates not in S are not interpreted by $\mathcal{I}_{|S}$.

Definition 11. Let $\mathcal{O}_1, \mathcal{O}_2$ be ontologies and S a signature. We say that

- \mathcal{O}_1 and \mathcal{O}_2 are *semantically S -inseparable* and write $\mathcal{O}_1 \approx_S^{\text{sem}} \mathcal{O}_2$ if

$$\{\mathcal{I}_{|S} \mid \mathcal{I} \models \mathcal{O}_1\} = \{\mathcal{I}_{|S} \mid \mathcal{I} \models \mathcal{O}_2\}.$$

- \mathcal{O}_2 is a *semantic S -conservative extension* of \mathcal{O}_1 if $\mathcal{O}_1 \subseteq \mathcal{O}_2$ and $\mathcal{O}_1 \approx_S^{\text{sem}} \mathcal{O}_2$.

We simply speak of a *semantic conservative extension* if $S = \text{sig}(\mathcal{O}_1)$. In the literature, the term ‘model conservative extension’ is used synonymously. We now show equivalence of semantic inseparability and inseparability w.r.t. $\mathcal{QL}_{\mathcal{SO}}$.

Theorem 9. Let \mathcal{O}_1 and \mathcal{O}_2 be ontologies and S a signature. Then the following conditions are equivalent:

- $\mathcal{O}_1 \approx_S^{\mathcal{QL}_{\mathcal{SO}}} \mathcal{O}_2$;
- $\mathcal{O}_1 \approx_S^{\text{sem}} \mathcal{O}_2$.

Proof. The implication from Point 1 to Point 2 follows from the fact that no second-order sentence using only predicates from S can distinguish two models whose reducts to S are isomorphic. The other direction holds since we have $\mathcal{O}_1 \approx_S^{\text{sem}} \mathcal{O}_2$ if \mathcal{O}_1 implies the second-order sentence $\exists S_1 \cdots \exists S_n. \bigwedge \mathcal{O}_2$ with $\{S_1, \dots, S_n\} = \text{sig}(\mathcal{O}_2) \setminus S$ and \mathcal{O}_2 implies the second-order sentence $\exists S'_1 \cdots \exists S'_m. \bigwedge \mathcal{O}_1$ with $\{S'_1, \dots, S'_m\} = \text{sig}(\mathcal{O}_1) \setminus S$. □

The following robustness properties are easily proved by exploiting the high expressive power of \mathcal{SO} .

Theorem 10. Let \mathcal{L} be a fragment of \mathcal{SO} . Then $(\mathcal{L}, \mathcal{QL}_{\mathcal{SO}})$ is robust under vocabulary extensions, joins and under replacement.

The relation between inseparability w.r.t. $\mathcal{QL}_{\mathcal{FO}}$ and semantic inseparability has extensively been discussed in the literature on software specification [26, 40, 47]. In [8, 48], the reader can find a proof that semantic inseparability is strictly stronger than inseparability w.r.t. $\mathcal{QL}_{\mathcal{FO}}$. However, when

restricted to *finite models*, the two notions are equivalent because every finite model can be described up to isomorphism in first-order logic with equality. The computational complexity of deciding semantic S -inseparability has been studied in [37, 28].

Theorem 11. *For \mathcal{ALC} -ontologies, semantic conservativity is Π_1^1 -hard, thus neither decidable nor recursively enumerable.*

Conditions under which semantic inseparability becomes decidable are investigated in [28]. For example, if S contains no roles, then semantic S -inseparability becomes decidable:

Theorem 12. *For \mathcal{ALC} -ontologies \mathcal{O}_1 and \mathcal{O}_2 and signatures S containing no roles, it is decidable (and $\text{NEXPTIME}^{\text{NP}}$ -complete) whether $\mathcal{O}_1 \approx_S^{\text{sem}} \mathcal{O}_2$.*

6 More on Robustness

We take a closer look at the robustness properties and relate them to standard notions from logic such as interpolation and Robinson consistency.

6.1 Robustness under Replacement

We start with addressing the question why $(\mathcal{FO}, \mathcal{QL}_{\mathcal{FO}})$ and $(\mathcal{LU}, \mathcal{QL}_{\mathcal{LU}})$, with \mathcal{L} as in Theorem 3, are robust under replacement while $(\mathcal{L}, \mathcal{QL}_{\mathcal{L}})$ are not. A first hint is given by the simple proof of Theorem 8, which crucially exploits that the query language $\mathcal{QL}_{\mathcal{FO}}$ is closed under the Boolean operators. This is clearly not the case for $\mathcal{QL}_{\mathcal{L}}$, but also not for $\mathcal{QL}_{\mathcal{LU}}$. However, it is well-known that there is a close connection between the universal role and the Boolean operators, see e.g. [21]. To make this connection formal, we extend $\mathcal{QL}_{\mathcal{L}}$ to the query language $\mathcal{QL}_{\mathcal{L}}^B$ that consists of all *Boolean combinations* of $\mathcal{QL}_{\mathcal{L}}$ -sentences: $\mathcal{QL}_{\mathcal{L}}^B$ is the set of sentences ψ defined by

$$\psi ::= \varphi \mid \psi_1 \wedge \psi_2 \mid \neg\psi,$$

where $\varphi \in \mathcal{QL}_{\mathcal{L}}$ and ψ, ψ_1, ψ_2 range over $\mathcal{QL}_{\mathcal{L}}^B$ -sentences. The next result explains why $(\mathcal{LU}, \mathcal{QL}_{\mathcal{LU}})$ is robust under replacement. It is an easy consequence of the well-known fact that $\mathcal{QL}_{\mathcal{LU}}$ and $\mathcal{QL}_{\mathcal{L}}^B$ have the same expressive power in the following sense [21]: for every $\varphi \in \mathcal{QL}_{\mathcal{LU}}$, there exists a $\varphi^* \in \mathcal{QL}_{\mathcal{L}}^B$ with $\text{sig}(\varphi) = \text{sig}(\varphi^*)$ such that $\mathcal{I} \models \varphi$ iff $\mathcal{I} \models \varphi^*$ holds for all interpretations \mathcal{I} , and vice versa.

Theorem 13. *Let \mathcal{L} be any of the DLs \mathcal{ALC} , \mathcal{ALCI} , \mathcal{ALCQ} , \mathcal{ALCQI} . Then $\approx_S^{\mathcal{QL}_{\mathcal{L}}^B} = \approx_S^{\mathcal{QL}_{\mathcal{LU}}}$ for any signature S .*

In Section 5.3, two approaches have been identified to deal with non-robustness under replacements of the query language $\mathcal{QL}_{\mathcal{L}}$ when \mathcal{L} does not contain the universal role: either build robustness under replacements into the definition of a module as in Definition 7 or switch from $(\mathcal{L}, \mathcal{QL}_{\mathcal{L}})$ to $(\mathcal{L}, \mathcal{QL}_{\mathcal{LU}})$. We now show that these two approaches are actually identical. To do this, it is convenient to define a notion of inseparability that has robustness under replacements build in, analogous to Definition 7: $\mathcal{QL}_{\mathcal{L}}^{\mathcal{O}}$ consists of all sentences

$$\left(\bigwedge \mathcal{O}^{\sharp}\right) \rightarrow \varphi,$$

where \mathcal{O} is an \mathcal{L} -ontology and φ a query from $\mathcal{QL}_{\mathcal{L}}$.

Theorem 14. *Let \mathcal{L} be any of the DLs \mathcal{ALC} , \mathcal{ALCT} , \mathcal{ALCQ} , \mathcal{ALCQI} . Then $\approx_S^{\mathcal{QL}_{\mathcal{L}}^{\mathcal{O}}} = \approx_S^{\mathcal{QL}_{\mathcal{LU}}}$ for any signature S .*

Proof. By Theorem 13, it suffices to show that $\approx_S^{\mathcal{QL}_{\mathcal{L}}^{\mathcal{O}}} = \approx_S^{\mathcal{QL}_{\mathcal{L}}^{\mathcal{B}}}$. The ‘if’ direction is easy since it is not hard to see that $\mathcal{QL}_{\mathcal{L}}^{\mathcal{O}} \subseteq \mathcal{QL}_{\mathcal{L}}^{\mathcal{B}}$. For the ‘only if’ direction, assume $\mathcal{O}_1 \approx_S^{\mathcal{QL}_{\mathcal{L}}^{\mathcal{O}}} \mathcal{O}_2$. Assume that $\mathcal{O}_2 \models \varphi$, where φ is a Boolean combination of \mathcal{L}_S -implications. Clearly, φ is equivalent to a conjunction of formulas of the form

$$(\top \sqsubseteq C_0) \vee \dots \vee (\top \sqsubseteq C_m) \vee \neg(\top \sqsubseteq C_{m+1}) \vee \dots \vee \neg(\top \sqsubseteq C_n)$$

and each such conjunct is equivalent to

$$(\top \sqsubseteq C_0) \vee \dots \vee (\top \sqsubseteq C_m) \vee \neg(\top \sqsubseteq D),$$

where $D = C_{m+1} \sqcap \dots \sqcap C_n$. Moreover, using the fact that models of \mathcal{L} -ontologies are closed under disjoint unions, one can show that there exists $i \leq m$ such that $\mathcal{O}_2 \models (\top \sqsubseteq C_i) \vee \neg(\top \sqsubseteq D)$, which implies $\mathcal{O}_2 \cup \{\top \sqsubseteq D\} \models \top \sqsubseteq C_i$. From $\mathcal{O}_1 \approx_S^{\mathcal{QL}_{\mathcal{L}}^{\mathcal{O}}} \mathcal{O}_2$, we obtain $\mathcal{O}_1 \cup \{\top \sqsubseteq D\} \models \top \sqsubseteq C_i$. Since this holds for all conjuncts of φ , we get $\mathcal{O}_1 \models \varphi$. \square

Now, the two mentioned approaches are identical since it clearly follows from Theorem 14 that \mathcal{O}_1 is a weak S -module w.r.t. $\mathcal{QL}_{\mathcal{L}}$ robust under replacement iff \mathcal{O}_1 is a weak S -module w.r.t. $\mathcal{QL}_{\mathcal{LU}}$ according to Definition 3.

Finally, Theorem 14 can provide us with another interesting perspective on the relationship between $(\mathcal{L}, \mathcal{QL}_{\mathcal{L}})$ and $(\mathcal{L}, \mathcal{QL}_{\mathcal{LU}})$. Obviously, $\approx_S^{\mathcal{QL}_{\mathcal{LU}}} \subseteq \approx_S^{\mathcal{QL}_{\mathcal{L}}}$. However, $\approx_S^{\mathcal{QL}_{\mathcal{LU}}}$ is much more than just *some* coarsening of $\approx_S^{\mathcal{QL}_{\mathcal{L}}}$ that yields robustness under replacement: it is the maximal one. The following is a direct consequence of Theorem 14 and the definition of $\mathcal{QL}_{\mathcal{L}}^{\mathcal{O}}$.

Corollary 1. *Let \mathcal{L} be any of the DLs \mathcal{ALC} , \mathcal{ALCT} , \mathcal{ALCQ} , \mathcal{ALCQI} and S a signature. Then $\approx_S^{\mathcal{QL}_{\mathcal{LU}}}$ is the maximal subset of $\approx_S^{\mathcal{QL}_{\mathcal{L}}}$ such that for all \mathcal{L} -ontologies \mathcal{O}_1 , \mathcal{O}_2 and \mathcal{O} with $\text{sig}(\mathcal{O}) \cap \text{sig}(\mathcal{O}_1 \cup \mathcal{O}_2) \subseteq S$, $\mathcal{O}_1 \approx_S^{\mathcal{QL}_{\mathcal{L}}} \mathcal{O}_2$ implies $\mathcal{O}_1 \cup \mathcal{O} \approx_S^{\mathcal{QL}_{\mathcal{L}}} \mathcal{O}_2 \cup \mathcal{O}$.*

6.2 Robustness under Vocabulary Extensions and Interpolation

In this section, we discuss robustness under vocabulary extensions and its relationship to interpolation. We consider the following standard notion of interpolation.

Definition 12 (Interpolation). A query language \mathcal{QL} has *weak interpolation* iff for every set Ψ of \mathcal{QL} -sentences and every \mathcal{QL} -sentence φ such that $\Psi \models \varphi$, there exists a set $I(\Psi, \varphi)$ of \mathcal{QL} -sentence such that

- $\text{sig}(I(\Psi, \varphi)) \subseteq \text{sig}(\Psi) \cap \text{sig}(\varphi)$;
- $\Psi \models I(\Psi, \varphi)$;
- $I(\Psi, \varphi) \models \varphi$.

\mathcal{QL} has *interpolation* if there always exists a *finite* set $I(\Psi, \varphi)$ with these properties.

For any *compact*³ query language \mathcal{QL} , weak interpolation implies interpolation. Interpolation has been investigated extensively in mathematical logic and for modal logics closely related to DLs. For example, propositional logic, first- and second-order logic, and basic modal logic have interpolation [14, 27]. Proofs of interpolation for a variety of DLs are given in Appendix B.

The following proposition shows that weak interpolation implies robustness under vocabulary extensions:

Proposition 2. *Suppose \mathcal{L} and \mathcal{QL} are given and $\mathcal{L} \subseteq \mathcal{QL}$. If \mathcal{QL} has weak interpolation, then $(\mathcal{L}, \mathcal{QL})$ is robust under vocabulary extensions.*

Proof. Suppose $\mathcal{O}_1 \approx_S^{\mathcal{QL}} \mathcal{O}_2$ and let φ be a \mathcal{QL} -sentence with $\text{sig}(\varphi) \cap \text{sig}(\mathcal{O}_1 \cup \mathcal{O}_2) \subseteq S$ such that $\mathcal{O}_1 \models \varphi$. By weak interpolation, there exists an interpolant $I(\mathcal{O}_1, \varphi)$. From $\text{sig}(I(\mathcal{O}_1, \varphi)) \subseteq S$ we obtain $\mathcal{O}_2 \models I(\mathcal{O}_1, \varphi)$. Hence $\mathcal{O}_2 \models \varphi$. \square

Theorem 15. *Let \mathcal{L} be any of the DLs \mathcal{ALC} , \mathcal{ALCQ} , \mathcal{ALCI} , \mathcal{ALCQI} , \mathcal{ALCU} , \mathcal{ALCQU} , \mathcal{ALCIU} , \mathcal{ALCQIU} . The following are robust under vocabulary extensions:*

1. $(\mathcal{L}, \mathcal{QL}_{\mathcal{L}})$;
2. $(\mathcal{L}', \mathcal{QL}_{\mathcal{FO}})$, for any fragment \mathcal{L}' of first-order logic;
3. $(\mathcal{L}', \mathcal{QL}_{\mathcal{SO}})$, for any fragment \mathcal{L}' of second-order logic.

Proof. By Proposition 9 of Appendix B, the mentioned languages $\mathcal{QL}_{\mathcal{L}}$ have interpolation and it thus remains to apply Proposition 2 to establish Point 1. Points 2 and 3 follow from Proposition 2 and the fact that first- and second order logic have interpolation.

³ \mathcal{QL} is compact if $\Psi \models \varphi$ implies that there exists a finite subset Ψ' of Ψ such that $\Psi' \models \varphi$. First-order logic and its fragments are compact. Second-order logic and first-order logic over finite models are not compact.

We state a partial converse of Proposition 2. An *infinitary ontology* is a finite or infinite set of second-order sentences.

Proposition 3. *Suppose $(\mathcal{QL}, \mathcal{QL})$ is robust under vocabulary extensions for infinitary ontologies. Then \mathcal{QL} has weak interpolation.*

Proof. Assume $\mathcal{O} \models \varphi$, where \mathcal{O} is a set of \mathcal{QL} -sentences and φ a \mathcal{QL} -sentence. Set $S = \text{sig}(\mathcal{O}) \cap \text{sig}(\varphi)$ and

$$\mathcal{O}' = \{\psi \in \mathcal{QL} \mid \mathcal{O} \models \psi, \text{sig}(\psi) \subseteq S\}.$$

Then $\mathcal{O} \models \mathcal{O}'$ and \mathcal{O} and \mathcal{O}' are S -inseparable w.r.t. \mathcal{QL} . By robustness under vocabulary extensions for infinitary ontologies, \mathcal{O} and \mathcal{O}' are S' -inseparable w.r.t. \mathcal{QL} , where $S' = \text{sig}(\varphi)$. From $\mathcal{O} \models \varphi$ we obtain $\mathcal{O}' \models \varphi$. Hence $I(\mathcal{O}, \varphi) = \mathcal{O}'$ is as required. \square

An interesting logic which is robust under vocabulary extension but does not have interpolation is first-order logic *over finite models*. One can easily show weak interpolation for this logic, and, therefore, robustness under vocabulary extensions. The argument for failure of interpolation is as follows: using a binary predicate symbol $<$ and a unary predicate symbol red , one can write a finite set \mathcal{O}_1 of first-order sentences that is satisfied exactly in those finite models which have an even number of points (state that $<$ is a linear order, exactly every second point is red, and the first and last point have distinct colours). Use a different binary relation symbol $<'$ and unary predicate green for a finite set \mathcal{O}_2 of first-order axioms which is satisfied exactly in finite models with an odd number of points. Then $\mathcal{O}_1 \models_{\text{fin}} \neg \bigwedge \mathcal{O}_2$, but there does not exist a finite interpolant.

6.3 Robustness under Joins and Interpolation

We discuss the relation between robustness under joins and interpolation. For $(\mathcal{QL}_{\mathcal{FO}}, \mathcal{QL}_{\mathcal{FO}})$, robustness under joins is easily seen to be equivalent to the well-known Robinson joint consistency property [14]: if T_1 and T_2 are consistent first-order theories both extending an S -complete theory T_0 with $\text{sig}(T_0) \subseteq S$ (i.e. $T_0 \models \varphi$ or $T_0 \models \neg\varphi$ for all φ over S) and $\text{sig}(T_1) \cap \text{sig}(T_2) \subseteq S$, then $T_1 \cup T_2$ is consistent. As this property is known to be closely related to interpolation, it is no surprise that robustness under joins is closely related to interpolation as well.

Proposition 4. *Let $\mathcal{L} \subseteq \mathcal{QL}$ and assume that \mathcal{QL} is closed under Boolean operators. If \mathcal{QL} has weak interpolation, then $(\mathcal{L}, \mathcal{QL})$ is robust under joins.*

Proof. Suppose \mathcal{QL} has weak interpolation, \mathcal{O}_1 and \mathcal{O}_2 are S -inseparable w.r.t. \mathcal{QL} and $\text{sig}(\mathcal{O}_1) \cap \text{sig}(\mathcal{O}_2) \subseteq S$. Assume $\mathcal{O}_1 \cup \mathcal{O}_2 \models \varphi$ where $\text{sig}(\varphi) \subseteq S$. Then

$$\mathcal{O}_1 \models \bigwedge \mathcal{O}_2 \rightarrow \varphi.$$

Take an interpolant $I(\mathcal{O}_1, \bigwedge \mathcal{O}_2 \rightarrow \varphi)$ and observe that its signature is contained in S . Then $\mathcal{O}_2 \models I(\mathcal{O}_1, \bigwedge \mathcal{O}_2 \rightarrow \varphi)$, by S -inseparability of \mathcal{O}_1 and \mathcal{O}_2 w.r.t. \mathcal{QL} and the assumption that \mathcal{QL} is closed under Boolean operators. Hence $\mathcal{O}_2 \models \bigwedge \mathcal{O}_2 \rightarrow \varphi$. But then $\mathcal{O}_2 \models \varphi$. \square

Theorem 16. *Let \mathcal{L} be any of the DLs \mathcal{ALC} , \mathcal{ALCQ} , \mathcal{ALCI} , \mathcal{ALCQI} , \mathcal{ALCU} , \mathcal{ALCTU} , \mathcal{ALCQU} , \mathcal{ALCQIU} . The following are robust under joins.*

1. $(\mathcal{L}, \mathcal{QL}_{\mathcal{L}})$;
2. $(\mathcal{L}', \mathcal{QL}_{\mathcal{FO}})$, for any fragment \mathcal{L}' of first-order logic;
3. $(\mathcal{L}', \mathcal{QL}_{\mathcal{SO}})$, for any fragment \mathcal{L}' of second-order logic.

Proof. Using Proposition 4, Points 2 and 3 follow from the fact that first- and second-order logic have interpolation and are closed under Boolean operators. For \mathcal{ALC} , \mathcal{ALCQ} , \mathcal{ALCI} and \mathcal{ALCQI} , robustness under joins is proved in Proposition 10 of Appendix A.

It remains to show robustness under joins for \mathcal{LU} if \mathcal{L} is any of \mathcal{ALC} , \mathcal{ALCI} , \mathcal{ALCQ} , \mathcal{ALCQI} . By Proposition 9 of Appendix B, $\mathcal{QL}_{\mathcal{LU}}$ has interpolation. As $\mathcal{QL}_{\mathcal{LU}}$ and $\mathcal{QL}_{\mathcal{L}}^B$ have the same expressive power (see Section 6.1), $\mathcal{QL}_{\mathcal{L}}^B$ has interpolation. Hence, by Proposition 4, $(\mathcal{QL}_{\mathcal{L}}^B, \mathcal{QL}_{\mathcal{L}}^B)$ is robust under joins. Once more since $\mathcal{QL}_{\mathcal{LU}}$ and $\mathcal{QL}_{\mathcal{L}}^B$ have the same expressive power, $(\mathcal{LU}, \mathcal{QL}_{\mathcal{LU}})$ is robust under joins. \square

The following is a partial converse of Proposition 4.

Proposition 5. *Let \mathcal{QL} be a fragment of first-order logic closed under Boolean operators such that $(\mathcal{QL}, \mathcal{QL})$ is robust under joins for infinitary ontologies. Then \mathcal{QL} has interpolation.*

Proof. Suppose $\mathcal{O} \models \varphi$. Let $S = \text{sig}(\mathcal{O}) \cap \text{sig}(\varphi)$ and

$$\Psi_0 = \{\psi \in \mathcal{QL} \mid \text{sig}(\psi) \subseteq S, \mathcal{O} \models \psi\}.$$

We show that Ψ_0 is an interpolant for (\mathcal{O}, φ) . Assume not. Then $\Psi_0 \cup \{\neg\varphi\}$ is satisfiable. Take a model \mathcal{I} satisfying $\Psi_0 \cup \{\neg\varphi\}$ and denote by \mathcal{O}' the set of \mathcal{QL} -sentences ψ with $\text{sig}(\psi) \subseteq S$ which are true in \mathcal{I} . Then both $\mathcal{O}' \cup \mathcal{O}$ and $\mathcal{O}' \cup \{\neg\varphi\}$ are S -conservative extensions of \mathcal{O}' w.r.t. \mathcal{QL} . By robustness under joins, $\mathcal{O}' \cup \mathcal{O} \cup \{\neg\varphi\}$ is an S -conservative extension of \mathcal{O}' w.r.t. \mathcal{QL} ; in particular, it is consistent. Hence $\mathcal{O} \not\models \varphi$ and we have derived a contradiction. \square

7 Uniform Interpolation and Forgetting

Assume that we want to re-use the information that an ontology \mathcal{O} provides about a certain signature S in an application where only queries formulated in \mathcal{QL} are relevant. Then we have two options. The first is to extract an S -module, i.e., to identify a subset \mathcal{O}' of \mathcal{O} that is S -inseparable from \mathcal{O} w.r.t.

\mathcal{QL} (or satisfies even stronger conditions, cf. the other possible definitions of a module). In this approach, which is pursued in [28, 15], the extracted module may also contain symbols not in S . The second option is to construct a new ontology \mathcal{O}_S that *contains only symbols from S* and has the same consequences in \mathcal{QL} over S as \mathcal{O} in the sense that \mathcal{O} and \mathcal{O}_S are S -inseparable w.r.t. \mathcal{QL} . Whether such an ontology \mathcal{O}_S exists and can be effectively constructed depends on the ontology language used, the signature S and the query language \mathcal{QL} . This problem has been studied by different research communities under various names such as *forgetting* [34, 43, 50], *uniform interpolation* [42, 23, 49] and *variable elimination*.

Definition 13 (Uniform Interpolation (Forgetting)). *($\mathcal{L}, \mathcal{QL}$) has uniform interpolation if for every \mathcal{L} -ontology \mathcal{O} and signature S there exists an \mathcal{L} -ontology \mathcal{O}_S with $\text{sig}(\mathcal{O}_S) \subseteq S$ such that $\mathcal{O} \approx_{S'}^{\mathcal{QL}} \mathcal{O}_S$ for all S' with $\text{sig}(\mathcal{O}) \cap S' \subseteq S$. In this case, \mathcal{O}_S is called a S -uniform interpolant of \mathcal{O} w.r.t. $(\mathcal{L}, \mathcal{QL})$.*

Note that if $(\mathcal{QL}, \mathcal{QL})$ has uniform interpolation, then \mathcal{QL} has interpolation: assume that $\mathcal{O} \models \varphi$ and let $S = \text{sig}(\mathcal{O}) \cap \text{sig}(\varphi)$. Then $\mathcal{O} \models \mathcal{O}_S$ and $\mathcal{O}_S \models \varphi$. Thus, \mathcal{O}_S is an interpolant. This explains the name *uniform* interpolant as it is an interpolant that does not depend on the right hand side of the consequence.

In the context of forgetting, it can also be sensible to define uniform interpolation in a slightly less strict way, namely by demanding inseparability only w.r.t. S rather than the signatures S' used in Definition 13. By Proposition 2, this definition is equivalent to the stronger one if \mathcal{QL} has interpolation.

There is an intimate connection between the computation of S -uniform interpolants and deciding S -inseparability. In particular, if $(\mathcal{L}, \mathcal{QL})$ has uniform interpolation, then S -inseparability of \mathcal{L} -ontologies \mathcal{O} and \mathcal{O}' w.r.t. \mathcal{QL} can be decided by first computing S -uniform interpolants \mathcal{O}_S of \mathcal{O} w.r.t. $(\mathcal{L}, \mathcal{QL})$ and \mathcal{O}'_S of \mathcal{O}' w.r.t. $(\mathcal{L}, \mathcal{QL})$, and then checking whether $\mathcal{O}'_S \models \varphi$ for all $\varphi \in \mathcal{O}_S$ and $\mathcal{O}_S \models \varphi$ for all $\varphi \in \mathcal{O}'_S$.

It is easy to see that $(\mathcal{QL}_{S\mathcal{O}}, \mathcal{QL}_{S\mathcal{O}})$ has uniform interpolation: given an ontology \mathcal{O} in second-order logic and a signature S , let \mathcal{O}_S consist of the sentence

$$\exists S_1 \dots \exists S_n. \bigwedge \mathcal{O},$$

where $\{S_1, \dots, S_n\} = \text{sig}(\mathcal{O}) \setminus S$. Then \mathcal{O}_S is a uniform interpolant. We now investigate uniform interpolants for weaker languages.

Example 12. Let

$$\mathcal{O} = \{\text{Hand} \sqsubseteq \text{Body_part}, \text{Body_part} \sqsubseteq \text{Physical_object}\}$$

and $S = \{\text{Hand}, \text{Physical_object}\}$. Then $\mathcal{O}_S = \{\text{Hand} \sqsubseteq \text{Physical_object}\}$ is an S -uniform interpolant of \mathcal{O} w.r.t. any of the query languages \mathcal{QL} considered in this paper.

Example 13. Let

$$\mathcal{O} = \{\text{Human} \sqsubseteq \exists \text{child_of.Male}\}, \quad S = \{\text{Human, Male}\}$$

Then $\mathcal{O}_S = \{\text{Human} \sqsubseteq \exists u.\text{Male}\}$ is an S -uniform interpolant of \mathcal{O} w.r.t. any query language with expressivity between \mathcal{QL}_{ALCU} and \mathcal{QL}_{SO} .

Theorem 17. *Let \mathcal{L} be any of the DLs ALC , $ALCQ$, $ALCI$, $ALCQI$, $ALCU$, $ALCIU$, $ALCQU$, $ALCQIU$. Then for every \mathcal{L} -ontology \mathcal{O} and every signature S that consists of concept names, there exists an S -uniform interpolant of \mathcal{O} w.r.t. $(\mathcal{L}, \mathcal{QL}_{\mathcal{L}})$.*

Proof. By Theorem 15, all $(\mathcal{L}, \mathcal{QL}_{\mathcal{L}})$ are robust under vocabulary extensions. Thus, it is sufficient to show that there is an ontology \mathcal{O}_S with $\mathcal{O} \approx_S^{\mathcal{QL}_{\mathcal{L}}} \mathcal{O}_S$. The set

$$\mathcal{O}'_S = \{C \sqsubseteq D \mid \mathcal{O} \models C \sqsubseteq D, C, D \text{ } \mathcal{L}_S\text{-concepts}\}.$$

is our starting point. It has the required property, but may be infinite.

Assume first that \mathcal{L} does not contain the universal role. Then, since S does not contain any roles, every concept C from \mathcal{O}'_S is equivalent to some Boolean expression over S . As there are only finitely many non-equivalent such expressions, we obtain a finite set $\mathcal{O}_S \subseteq \mathcal{O}'_S$ which is S -inseparable from \mathcal{O}'_S w.r.t. $\mathcal{QL}_{\mathcal{L}}$. \mathcal{O}_S is as required.

Now assume that \mathcal{L} contains the universal role. Then every \mathcal{L}_S -concept S can be regarded as a formula of modal logic S5. It is known (and straightforward to prove) that there are only finite many non-equivalent S5-formulas in a given number of variables. Hence, again, we obtain a finite set $\mathcal{O}_S \subseteq \mathcal{O}'_S$ which is S -inseparable from \mathcal{O}'_S w.r.t. $\mathcal{QL}_{\mathcal{L}}$. \mathcal{O}_S is as required. \square

It seems worthwhile to point out that uniform interpolants may be large: the smallest such \mathcal{O}_S from Theorem 17 can be of size exponential in \mathcal{O} [29]. In [29], it is proved that none of the combinations $(\mathcal{L}, \mathcal{QL}_{\mathcal{L}})$ from Theorem 17 has uniform interpolation.

Theorem 18. *Let \mathcal{L} be any of the DLs ALC , $ALCQ$, $ALCI$, $ALCQI$, $ALCU$, $ALCIU$, $ALCQU$, $ALCQIU$. Then $(\mathcal{L}, \mathcal{QL}_{\mathcal{L}})$ does not have uniform interpolation.*

Proof. For the ALC -ontology \mathcal{O} and signature S given in the proof of Lemma 6 of [29] there does not exist a \mathcal{FO} -ontology \mathcal{O}_S with $\text{sig}(\mathcal{O}_S) \subseteq S$ such that \mathcal{O} and \mathcal{O}_S are S -inseparable w.r.t. \mathcal{QL} , for any of the query languages \mathcal{QL} listed above. \square

It follows from the proof of Theorem 18 that $(\mathcal{QL}_{\mathcal{FO}}, \mathcal{QL}_{\mathcal{FO}})$ does not have uniform interpolation. We still provide a direct proof.

Proposition 6. *$(\mathcal{QL}_{\mathcal{FO}}, \mathcal{QL}_{\mathcal{FO}})$ does not have uniform interpolation.*

Proof. Let \mathcal{O} be an axiomatisation of the theory of dense linear-orders using the binary relation symbol $<$. Each model of \mathcal{O} has an infinite domain, \mathcal{O} therefore implies $\varphi_n = \exists x_1 \cdots \exists x_n \bigwedge_{i \neq j} x_i \neq x_j$, for all $n \geq 1$. But there does not exist a finite and consistent set of first-order axioms over the empty signature which implies all φ_n , $n \geq 1$. \square

We have seen that standard DLs and first-order logic do not have uniform interpolation. This defect can be cured by adding second-order quantification. We briefly discuss the extension $\mathcal{ALC}\mu$ of \mathcal{ALC} that has uniform interpolation. $\mathcal{ALC}\mu$ -concepts C are defined as follows. Let V be an infinite set of concept variables. Then

- every \mathcal{ALC} -concept, possibly with some concept names replaced with concept variables, is an $\mathcal{ALC}\mu$ -concept;
- if C is an $\mathcal{ALC}\mu$ -concept in which $X \in V$ occurs positively (under an even number of negations), then $\mu X.C$ is an $\mathcal{ALC}\mu$ -concept.

To assign a semantics to $\mathcal{ALC}\mu$ -concepts, an interpretation is combined with an assignment $\tau : V \rightarrow 2^{\Delta^{\mathcal{I}}}$. Then the extension $(\mu X.C)^{\mathcal{I}, \tau}$ of $\mu X.C$ is defined as

$$(\mu X.C)^{\mathcal{I}, \tau} = \bigcap \{S \subseteq \Delta^{\mathcal{I}} \mid C^{\mathcal{I}, \tau'} \subseteq S, \tau'(X) = S, \text{ for all } Y \neq X: \tau'(Y) = \tau(Y)\}.$$

A *closed* $\mathcal{ALC}\mu$ -concept is a $\mathcal{ALC}\mu$ -concept without free concept variables. An $\mathcal{ALC}\mu$ -ontology is a finite set of implications $C \sqsubseteq D$, where C and D are closed $\mathcal{ALC}\mu$ -concepts. Other notions are now defined in the same way as for \mathcal{ALC} . $\mathcal{ALC}\mu$ is a very powerful description logic in which subsumption is still EXPTIME-complete. We refer the reader to [7, 3] for further information.

The following result can now be proved using uniform interpolation results for the modal μ -calculus from [19].

Theorem 19. $(\mathcal{ALC}\mu, \mathcal{QL}_{\mathcal{ALC}\mu})$ has uniform interpolation.

8 Weaker Description Logics and Acyclic TBoxes

So far, we have concentrated on extensions of \mathcal{ALC} and ontologies that are sets of implications between concepts or even first- and second-order sentences. In this section, we have a brief look at what happens if we consider weaker DLs and/or a weaker form of ontology called an acyclic ontology.

8.1 \mathcal{EL}

\mathcal{EL} and its extensions form a family of lightweight description logics that are popular for the formulation of large medical and biological ontologies such as SNOMED CT. Technically, \mathcal{EL} is the fragment of \mathcal{ALC} that admits only the constructors $C \sqcap D$ and $\exists r.C$ and \mathcal{EL} -ontologies are finite sets of implications

$C \sqsubseteq D$ between \mathcal{EL} -concepts C, D . In \mathcal{EL} , subsumption and a number of other relevant reasoning tasks can be solved in polynomial time [2]. Note that every \mathcal{EL} -ontology is satisfiable and thus subsumption is not reducible to satisfiability and the query language \mathcal{QL}_\perp does not separate any \mathcal{EL} -ontologies.

In the following, we briefly summarise what is known about modularity of \mathcal{EL} -ontologies. The query languages $\mathcal{QL}_{\mathcal{EL}}$, $\mathcal{QL}_{\mathcal{EL}}^I$, and $\mathcal{QL}_{\mathcal{EL}}^q$ are defined in the same way as the corresponding query languages for \mathcal{ALC} , except that all involved concepts have to be formulated in \mathcal{EL} . The following theorem summarises the results for inseparability in \mathcal{EL} obtained in [38, 29, 39]:

Theorem 20.

(i) *S-inseparability w.r.t. $\mathcal{QL}_{\mathcal{EL}}^I$ coincides with S-inseparability w.r.t. $\mathcal{QL}_{\mathcal{EL}}$ but does not coincide with S-inseparability w.r.t. $\mathcal{QL}_{\mathcal{EL}}^q$.*

(ii) *For \mathcal{QL} any of $\mathcal{QL}_{\mathcal{EL}}$, $\mathcal{QL}_{\mathcal{EL}}^I$, $\mathcal{QL}_{\mathcal{EL}}^q$:*

- *S-inseparability w.r.t. \mathcal{QL} is EXPTIME-complete.*
- *$(\mathcal{EL}, \mathcal{QL})$ is robust under signature extensions and joins, but not under replacement.*
- *$(\mathcal{EL}, \mathcal{QL})$ does not have uniform interpolation.*

(iii) *Semantic conservativity is undecidable for \mathcal{EL} -ontologies.*

Many interesting problems remain open for \mathcal{EL} . For example, nothing is known about the minimal query language extending $\mathcal{QL}_{\mathcal{EL}}$ and robust under replacement.

8.2 DL-Lite

The DL-Lite family of description logics consists of lightweight languages whose main application is to describe constraints over data repositories. In contrast to other DLs, data complexity of query answering is within LOGSPACE for most members of the family, and conjunctive queries over ontologies and ABoxes can be effectively rewritten as SQL queries so that standard database query engines can be used for query answering [9, 10, 1].

Modularity properties and the complexity of corresponding reasoning problems have been investigated in [30, 31] for the dialects $DL-Lite_{bool}$ and $DL-Lite_{horn}$. It turns out that those languages are rather well-behaved. We summarise here the behaviour of $DL-Lite_{bool}$ and refer to [31] for information about $DL-Lite_{horn}$. $DL-Lite_{bool}$ concepts are constructed from N_C and N_R using the Boolean operators, \sqcap and \neg , and *unqualified* number restrictions ($\geq n r$) and ($\leq n r$), where r is a role name or its inverse. A $DL-Lite_{bool}$ ontology is a finite set of implications between $DL-Lite_{bool}$ -concepts. The query languages $\mathcal{QL}_{DL-Lite_{bool}}$ and $\mathcal{QL}_{DL-Lite_{bool}}^q$ are defined in the same way as the corresponding query languages for \mathcal{ALC} , except that now all concepts involved range over $DL-Lite_{bool}$ concepts. The following theorem summarises what is known about $DL-Lite_{bool}$ [31]:

Theorem 21. Let $(\mathcal{L}, \mathcal{QL}) = (DL\text{-Lite}_{\text{bool}}, \mathcal{QL}_{DL\text{-Lite}_{\text{bool}}})$. Then the following holds:

- S -inseparability w.r.t. \mathcal{QL} is Π_2^P -complete;
- $(\mathcal{L}, \mathcal{QL})$ is robust under vocabulary extensions and joins, but not under replacement;
- $(\mathcal{L}, \mathcal{QL})$ has uniform interpolation.

Let $(\mathcal{L}, \mathcal{QL}) = (DL\text{-Lite}_{\text{bool}}, \mathcal{QL}_{DL\text{-Lite}_{\text{bool}}}^q)$. Then the following holds:

- S -inseparability w.r.t. \mathcal{QL} is Π_2^P -complete;
- $(\mathcal{L}, \mathcal{QL})$ is robust under vocabulary extensions, joins and replacement.

For results on uniform interpolation of $(DL\text{-Lite}, \mathcal{QL}_{DL\text{-Lite}_{\text{bool}}}^q)$, we once more refer to [31]. Experimental results on deciding S -inseparability using QBF-solvers are also reported in [31].

8.3 Acyclic Ontologies

For a description logic \mathcal{L} , an *acyclic \mathcal{L} -ontology* \mathcal{O} is a finite set of expressions $A \equiv C$ and $A \sqsubseteq C$, A a concept name, such that

- no concept name occurs twice on the left hand side and
- the relation $\prec_{\mathcal{O}} \subseteq \mathbf{N}_{\mathcal{C}} \times \mathbf{N}_{\mathcal{C}}$, defined by $(A, B) \in \prec_{\mathcal{O}}$ iff B occurs in C for some $A \equiv C \in \mathcal{O}$ or $A \sqsubseteq C \in \mathcal{O}$, is acyclic.

We refer the reader to [3] for more information. Many ontologies from practical applications are acyclic, including such prominent cases as SNOMED CT. In this section, we discuss the impact on S -inseparability and modularity of switching from general ontologies as used so far to acyclic ones.

For members of the \mathcal{EL} family of DLs, this switch can significantly reduce the complexity of reasoning about inseparability and may enable desirable features such as uniform interpolation. In particular, it has been shown that

- S -inseparability of acyclic \mathcal{EL} -ontologies w.r.t. $\mathcal{QL}_{\mathcal{EL}}$ is tractable, in contrast to EXPTIME-hardness for general \mathcal{EL} -ontologies [29]. Experiments show that S -inseparability of distinct versions of the huge SNOMED CT ontology can be swiftly decided in practice.
- deciding whether \mathcal{O}_2 is a semantic S -conservative extension of \mathcal{O}_1 is tractable for acyclic \mathcal{EL} -ontologies $\mathcal{O}_1, \mathcal{O}_2$ and signatures $S \supseteq \text{sig}(\mathcal{O}_1)$, in contrast to undecidability for general \mathcal{EL} -ontologies [28]. An efficient module extraction algorithm based on this result has been implemented and successfully used with SNOMED CT.

As already mentioned, another benefit of acyclic ontologies over general ones is uniform interpolation. While $(\mathcal{EL}, \mathcal{QL}_{\mathcal{EL}})$ does not admit uniform interpolation, an S -uniform interpolant w.r.t. $(\mathcal{EL}, \mathcal{QL}_{\mathcal{EL}})$ exists for every acyclic \mathcal{EL} -ontology \mathcal{O} and signature S [29].

For expressive DL such as \mathcal{ALC} , acyclicity of ontologies typically does not yield any benefits. For example, it is still Π_1^1 -hard to decide whether an acyclic \mathcal{ALC} -ontology is a semantic S -conservative extension of an empty ontology [28] and even for acyclic \mathcal{ALC} -ontologies and signatures S there does not always exist a uniform S -interpolant w.r.t. $(\mathcal{ALC}, \mathcal{QL}_{\mathcal{ALC}})$ [29].

9 Conclusion

We have investigated the notion of inseparability of ontologies w.r.t. a query language. As argued in the introduction and throughout the paper, this notion is central to logic-based approaches to modularity of ontologies. In particular, inseparability is commonly used to define independence of a module inside an ontology, and it can also be employed to understand and control the ramifications of re-using an ontology within another ontology. We have argued that the notion of inseparability has to be parameterised by a query language, and have identified three important meta-properties of inseparability: robustness under vocabulary extension, under joins, and under replacement. We have also investigated the relationship between these properties and interpolation, and discussed the computation complexity of deciding inseparability. Finally, we have briefly touched upon the relationship between inseparability and forgetting/uniform interpolation. Numerous technical problems are still open. To mention a few, the robustness properties and computational complexity of inseparability w.r.t. the query languages $\mathcal{QL}_{\mathcal{L}}^q$ defined in terms of conjunctive queries have not yet been investigated in any detail for DLs above \mathcal{ALC} . Also, ‘positive’ results for forgetting/uniform interpolation have been established only for very lightweight fragments of \mathcal{ALC} .

The theory developed in this paper has been evaluated in practice for ontologies formulated in the lightweight description logics \mathcal{EL} and DL-Lite, with rather promising results [29, 31, 28]. In contrast, no ‘practical’ algorithms or experimental results have yet been obtained for deciding inseparability between ontologies formulated in more expressive languages such as \mathcal{ALC} . Thus, it remains to be explored whether deciding inseparability in such languages is feasible in practice, or whether more pragmatic approaches such as the locality-based one of [17] are the only feasible logic-based way to approach inseparability of ontologies formulated in expressive DLs.

We have confined our investigation to specific ontology and query languages, all of them fragments of second-order logic. In general, it would be interesting to develop a more general framework that allows to integrate ontologies formulated in (almost) arbitrary languages, covering, for example, non-classical logics, algebraic formalisms and non-monotonic languages. In software specification, the notion of institutions provides such a framework [25] and, recently, institutions have been proposed as a tool to investigate the modularity of ontologies [33, 36, 44]. However, a lot of work remains to

be done. For example, the important distinction between query and ontology language has not yet been made explicit in the institutions approach.

References

1. A. Artale, D. Calvanese, R. Kontchakov and M. Zakharyashev. DL-Lite in the light of first-order logic. In *Proceedings of AAAI'07*, pages 361–366. AAAI Press, 2007.
2. F. Baader, S. Brandt and C. Lutz. Pushing the \mathcal{EL} envelope. In *Proceedings of IJCAI'05*, pages 364–369. Professional Book Center, 2005.
3. F. Baader, D. Calvanese, D. McGuinness, D. Nardi and P. Patel-Schneider. *The Description Logic Handbook: Theory, implementation and applications*. Cambridge University Press, 2003.
4. J. Bao, D. Caragea and V. Honavar. On the semantics of linking and importing in modular ontologies. In *Proceedings of ISWC'06*, pages 72–86. Springer, 2006.
5. S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. Patel-Schneider and L. Stein. OWL Web Ontology Language reference. W3C Recommendation, 10 February 2004.
6. P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini and H. Stuckenschmidt. Contextualizing ontologies. *Journal of Web Semantics*, 1(4):325-343, 2004.
7. J. Bradfield and C. Stirling. Modal μ -calculus. In P. Blackburn, J. van Benthem and F. Wolter, editors, *Handbook of Modal Logic*. Elsevier, 2006.
8. P. Byers and D. H. Pitt. Conservative extensions: a cautionary note. *Bulletin of the EATCS*, 41:196–201, 1990.
9. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini and R. Rosati. DL-Lite: Tractable description logics for ontologies. In *Proceedings of AAAI'05*, pages 602–607. AAAI Press / MIT Press, 2005.
10. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini and R. Rosati. Data complexity of query answering in description logics. In *Proceedings of KR'06*, pages 260–270. AAAI Press, 2006.
11. D. Calvanese, T. Eiter and M. Ortiz. Answering regular path queries in expressive description logics: An automata-theoretic approach. In *Proceedings of AAAI'07*, pages 391–396. AAAI Press, 2007.
12. D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini and R. Rosati. Data complexity of query answering in description logics. In *Proceedings of KR'06*, pages 260–270. AAAI Press, 2006.
13. B. ten Cate, W. Conradie, M. Marx and Y. Venema. Definitorially complete description logics. In *Proceedings of KR'06*, pages 79–89. AAAI Press, 2006.
14. C. C. Chang and H. J. Keisler. *Model Theory*, volume 73 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 1990.
15. B. Cuenca Grau, I. Horrocks, Y. Kazakov and U. Sattler. Just the right amount: Extracting modules from ontologies. In *Proceedings of WWW'07*, pages 717–727. ACM, 2007.
16. B. Cuenca Grau, I. Horrocks, Y. Kazakov and U. Sattler. A logical framework for modularity of ontologies. In *Proceedings of IJCAI'07*, pages 298–303. AAAI Press, 2007.
17. B. Cuenca Grau, I. Horrocks, Y. Kazakov and U. Sattler. Modular reuse of ontologies: Theory and practice. *Journal of Artificial Intelligence Research*, 31:273–318, 2008.

18. B. Cuenca Grau, B. Parsia, E. Sirin and A. Kalyanpur. Modularity and web ontologies. In *Proceedings of KR'06*, pages 198–209. AAAI Press, 2006.
19. G. D'Agostino and M. Hollenberg. Uniform interpolation, automata, and the modal μ -calculus. In *Proceedings of AiML'98*, volume 1. CSLI Publishers, 1998.
20. T. Eiter, M. Fink and S. Woltran. Semantical characterizations and complexity of equivalences in answer set programming. *ACM Transactions in Computational Logic*, 8(3), 2007.
21. D. M. Gabbay, A. Kurucz, F. Wolter and M. Zakharyashev. *Many-Dimensional Modal Logics: Theory and Applications*, volume 148 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 2003.
22. S. Ghilardi, C. Lutz and F. Wolter. Did I damage my ontology? A case for conservative extensions in description logic. In *Proceedings of KR'06*, pages 187–197. AAAI Press, 2006.
23. S. Ghilardi and M. Zawadowski. Undefinability of propositional quantifiers in the modal system S4. *Studia Logica*, 55(2): 259–271, 1995.
24. B. Glimm, C. Lutz, I. Horrocks and U. Sattler. Answering conjunctive queries in the *SHIQ* description logic. *Journal of Artificial Intelligence Research*, 31:150–197, 2008.
25. J. A. Goguen and R. M. Burstall. Institutions: Abstract model theory for specification and programming. *Journal of ACM*, 39(1):95–146, 1992.
26. J. A. Goguen, R. Diaconescu and P. Stefaneas. Logical support for modularisation. In *Proceedings of Logical Environments*, pages 83–130. Cambridge University Press, 1993.
27. V. Goranko and M. Otto. Model theory of modal logic. In P. Blackburn, J. van Benthem and F. Wolter, editors, *Handbook of Modal Logic*. Elsevier, 2007.
28. B. Konev, C. Lutz, D. Walther and F. Wolter. Semantic modularity and module extraction in description logics. In *Proceedings of ECAI'08*, pages 55–59. IOS Press, 2008.
29. B. Konev, D. Walther and F. Wolter. The logical difference problem for description logic terminologies. In *Proceedings of IJCAR'08*, pages 259–274. Springer, 2008.
30. R. Kontchakov, F. Wolter and M. Zakharyashev. Modularity in DL-Lite. In *Proceedings of DL'07*, pages 76–87. CEUR-WS, 2007.
31. R. Kontchakov, F. Wolter and M. Zakharyashev. Can you tell the difference between DL-Lite ontologies. In *Proceedings of KR'08*, pages 285–295. AAAI Press, 2008.
32. O. Kutz, C. Lutz, F. Wolter and M. Zakharyashev. E-connections of abstract description systems. *Artificial Intelligence*, 156(1):1–73, 2004.
33. O. Kutz and T. Mossakowski. Conservativity in structured ontologies. In *Proceedings of ECAI'08*. IOS Press, 2008.
34. J. Lang, P. Liberatore and P. Marquis. Propositional independence: formula-variable independence and forgetting. *Journal of Artificial Intelligence Research*, 18:391–443, 2003.
35. V. Lifschitz, D. Pearce and A. Valverde. Strongly equivalent logic programs. *ACM Transactions on Computational Logic*, 2(4):526–541, 2001.
36. K. Lüttich, C. Masolo and S. Borgo. Development of modular ontologies in CASL. In *Proceedings of WoMO'06*. CEUR-WS, 2006.
37. C. Lutz, D. Walther and F. Wolter. Conservative extensions in expressive description logics. In *Proceedings of IJCAI'07*, pages 453–458. AAAI Press, 2007.

38. C. Lutz and F. Wolter. Conservative extensions in the lightweight description logic \mathcal{EL} . In *Proceedings of CADE'07*, pages 84–99. Springer, 2007.
39. C. Lutz and F. Wolter. Deciding inseparability and conservative extensions in the description logic \mathcal{EL} . Technical report, University of Liverpool, 2008.
40. T. S. E. Maibaum. Conservative extensions, interpretations between theories and all that! In *Proceedings of TAPSOFT'97*, pages 40–66. Springer, 1997.
41. L. Maksimova. Complexity of interpolation and related problems in positive calculi. *Journal of Symbolic Logic*, 67(1):397–408, 2002.
42. A. Pitts. On an interpretation of second-order quantification in first-order intuitionistic propositional logic. *Journal of Symbolic Logic*, 57(1):33–52, 1992.
43. R. Reiter and F. Lin. Forget it! In *Proceedings of AAAI Fall Symposium on Relevance*, pages 154–159. AAAI Press, 1994.
44. M. Schorlemmer and Y. Kalfoglou. Institutionalising ontology-based semantic integration. *Applied Ontology*, 2008. To appear.
45. L. Serafini, A. Borgida and A. Taminin. Aspects of distributed and modular ontology reasoning. In *Proceedings of IJCAI'05*, pages 570–575. Professional Book Center, 2005.
46. K. Spackman. Managing clinical terminology hierarchies using algorithmic calculation of subsumption: Experience with SNOMED-RT. *JAMIA*, 2000. Fall Symposium Special Issue.
47. P. Veloso. Yet another cautionary note on conservative extensions: a simple case with a computing flavour. *EATCS-Bulletin*, 46:188–193, 1992.
48. P. Veloso and S. Veloso. Some remarks on conservative extensions. a socratic dialog. *EATCS-Bulletin*, 43:189–198, 1991.
49. A. Visser. Uniform interpolation and layered bisimulation. In *Gödel '96 (Brno, 1996)*, volume 6 of *Lecture Notes in Logic*. Springer Verlag, 1996.
50. Z. Wang, K. Wang, R. W. Topor and J. Z. Pan. Forgetting Concepts in DL-Lite. In *Proceedings of ESWC'08*, pages 245–257. IOS Press, 2008.

A Deferred Proofs for Section 5

Proposition 7. *Neither $(\mathcal{ALCO}, \mathcal{QL}_{\mathcal{ALCO}})$ nor $(\mathcal{ALCH}, \mathcal{QL}_{\mathcal{ALCH}})$ is robust under joins.*

Proof. We start with \mathcal{ALCO} . Let $\mathcal{O}_0 = \{\top \sqsubseteq \exists r.\top\}$ and

$$\mathcal{O}_1 = \mathcal{O}_0 \cup \{A \sqsubseteq \forall r.\neg A, \neg A \sqsubseteq \forall r.A\}, \quad \mathcal{O}_2 = \mathcal{O}_0 \cup \{\{a\} \sqsubseteq \exists r.\{a\}\}.$$

Then both, \mathcal{O}_1 and \mathcal{O}_2 , are conservative extensions of \mathcal{O}_0 w.r.t. $\mathcal{QL}_{\mathcal{ALCO}}$. On the other hand, $\mathcal{O}_1 \cup \mathcal{O}_2$ is inconsistent. It follows that $\mathcal{O}_1 \approx_S^{\mathcal{ALCO}} \mathcal{O}_2$ for $S = \{r\}$ but $\mathcal{O}_1 \cup \mathcal{O}_2 \not\approx_S^{\mathcal{ALCO}} \mathcal{O}_2$.

For \mathcal{ALCH} take

$$\begin{aligned} \mathcal{O}_0 &= \{\top \sqsubseteq \forall r_i \forall r_j. \perp \mid i, j = 1, 2\} \cup \{\exists r_1.\top \equiv \exists r_2.\top\}, \\ \mathcal{O}_1 &= \mathcal{O}_0 \cup \{s \sqsubseteq r_1, s \sqsubseteq r_2, \exists r_1.\top \sqsubseteq \exists s.\top\}, \end{aligned}$$

and

$$\mathcal{O}_2 = \mathcal{O}_0 \cup \{\exists r_1.\top \sqsubseteq \forall r_1.B \sqcap \forall r_2.\neg B\}$$

Then \mathcal{O}_1 and \mathcal{O}_2 are conservative extensions of \mathcal{O}_0 w.r.t. $\mathcal{QL}_{\mathcal{ALCH}}$. Hence \mathcal{O}_1 and \mathcal{O}_2 are S -inseparable w.r.t. $\mathcal{QL}_{\mathcal{ALCH}}$ for $S = \{r_1, r_2\}$.

On the other hand, $\mathcal{O}_1 \not\models \exists r_1.\perp \sqsubseteq \perp$ and $\mathcal{O}_1 \cup \mathcal{O}_2 \models \exists r_1.\top \sqsubseteq \perp$ and, therefore, \mathcal{O}_1 and $\mathcal{O}_1 \cup \mathcal{O}_2$ are not S -inseparable w.r.t. $\mathcal{QL}_{\mathcal{ALCH}}$. \square

Theorem 4. Let \mathcal{L} be any of the DLs \mathcal{ALC} , \mathcal{ALCI} , \mathcal{ALCO} , \mathcal{ALCQI} , S a signature, and assume that \mathcal{O}_1 and \mathcal{O}_2 contain only roles from S . Then

$$\mathcal{O}_1 \approx_S^{\mathcal{QL}_{\mathcal{L}}} \mathcal{O}_2 \quad \text{implies} \quad \mathcal{O}_1 \cup \mathcal{O} \approx_S^{\mathcal{QL}_{\mathcal{L}}} \mathcal{O}_2 \cup \mathcal{O},$$

for all \mathcal{L} -ontologies \mathcal{O} with $\text{sig}(\mathcal{O}) \cap \text{sig}(\mathcal{O}_1 \cup \mathcal{O}_2) \subseteq S$.

Proof. We give a sketch for the case $\mathcal{L} = \mathcal{ALC}$. Let $\mathcal{O}_1 \approx_S^{\mathcal{QL}_{\mathcal{L}}} \mathcal{O}_2$ and assume that $\mathcal{O}_1 \cup \mathcal{O} \models C_0 \sqsubseteq D_0$, where $\text{sig}(\mathcal{O}) \cap \text{sig}(\mathcal{O}_1 \cup \mathcal{O}_2) \subseteq S$ and $\text{sig}(C_0 \sqsubseteq D_0) \subseteq S$. Let $S' = S \cup \text{sig}(\mathcal{O})$. By robustness under vocabulary extensions, $\mathcal{O}_1 \approx_{S'}^{\mathcal{QL}_{\mathcal{L}}} \mathcal{O}_2$. Let

$$\Gamma = \{\forall r_1. \dots \forall r_n. (\neg C \sqcup D) \mid C \sqsubseteq D \in \mathcal{O}, r_i \in S', n \geq 0\}.$$

Using the condition that there are no additional roles in \mathcal{O}_1 , it is not difficult to show that $d \in (\neg C_0 \sqcup D_0)^{\mathcal{I}}$ for every model \mathcal{I} of \mathcal{O}_1 and $d \in \Delta^{\mathcal{I}}$ such that $d \in E^{\mathcal{I}}$ for all $E \in \Gamma$. By compactness, there exists a finite subset Γ' of Γ with the same property. It follows that

$$\mathcal{O}_1 \models \prod \Gamma' \sqcap C_0 \sqsubseteq D_0$$

from which we obtain $\mathcal{O}_2 \models \prod \Gamma' \sqcap C_0 \sqsubseteq D_0$. Since $\mathcal{O} \models \top \sqsubseteq \prod \Gamma'$, this implies $\mathcal{O}_2 \cup \mathcal{O} \models C_0 \sqsubseteq D_0$. \square

B Interpolation

We provide a proof of the interpolation property and robustness under joins for basic DLs. Interpolation has been extensively investigated for many modal logics closely related to DLs [41], and also for some description logics [13]. However, one has to be careful when transferring interpolation results from modal logic to DL: most interpolation results in modal logic regard modal operators as logical symbols and thus not as a part of the signature. This implies that even if the input formulas Ψ or φ do not contain a modal operator \Box , this operator is nevertheless permitted in the interpolant for Ψ and φ . In DLs, the corresponding constructor $\forall r$ is not permitted in the interpolant unless the role r occurs in Ψ or φ .

Let \mathcal{L} be a DL. For an interpretation \mathcal{I} , point $d \in \Delta^{\mathcal{I}}$ and signature S , we set

$$t_{\mathcal{I}}^{\mathcal{L},S}(d) = \{C \mid d \in C^{\mathcal{I}}, C \text{ an } \mathcal{L}_S\text{-concept}\}.$$

We say that two points d_1 and d_2 from possibly distinct interpretations \mathcal{I}_1 and \mathcal{I}_2 are \mathcal{L} , S -equivalent, written $(\mathcal{I}_1, d_1) \sim_S^{\mathcal{L}} (\mathcal{I}_2, d_2)$, if

$$t_{\mathcal{I}_1}^{\mathcal{L},S}(d_1) = t_{\mathcal{I}_2}^{\mathcal{L},S}(d_2).$$

We drop the \mathcal{L} and write $d_1 \sim_S d_2$ instead of $(\mathcal{I}_1, d_1) \sim_S^{\mathcal{L}} (\mathcal{I}_2, d_2)$ if \mathcal{I}_1 , \mathcal{I}_2 and \mathcal{L} are understood. A mapping f from $\Delta^{\mathcal{I}_1}$ to $\Delta^{\mathcal{I}_2}$ is called S -invariant iff $x \sim_S f(x)$ for all x in the domain of f .

Proposition 8. *Let \mathcal{L} be any of the DLs \mathcal{ALC} , \mathcal{ALCQ} , \mathcal{ALCI} , \mathcal{ALCQI} , \mathcal{ALCU} , \mathcal{ALCQU} , \mathcal{ALCTU} , or \mathcal{ALCQTU} . Let Ψ_1 and Ψ_2 be sets of \mathcal{QLC} -sentences and S a signature such that $\text{sig}(\Psi_1) \cap \text{sig}(\Psi_2) \subseteq S$. Assume there are models \mathcal{I}_1 and \mathcal{I}_2 of Ψ_1 and Ψ_2 , respectively, such that*

(a) *for all \mathcal{L}_S -concepts C : $C^{\mathcal{I}_1} \neq \emptyset$ iff $C^{\mathcal{I}_2} \neq \emptyset$.*

Then there exists a model \mathcal{I} of $\Psi_1 \cup \Psi_2$ such that

- *for all \mathcal{L}_S -concepts C : $C^{\mathcal{I}} \neq \emptyset$ iff $C^{\mathcal{I}_1} \neq \emptyset$ iff $C^{\mathcal{I}_2} \neq \emptyset$.*

If, in addition,

(b) *there are points $d_1 \in \Delta^{\mathcal{I}_1}$ and $e_1 \in \Delta^{\mathcal{I}_2}$ such that $d_1 \sim_S^{\mathcal{L}} e_1$,*

then \mathcal{I} contains a point d satisfying

- *$t_{\mathcal{I}}^{\mathcal{L},S_1}(d) = t_{\mathcal{I}_2}^{\mathcal{L},S_1}(e_1)$, where $S_1 = S \cup ((\mathbf{N}_C \cup \mathbf{N}_R) \setminus \text{sig}(\Psi_1))$ and*
- *$t_{\mathcal{I}}^{\mathcal{L},S_2}(d) = t_{\mathcal{I}_1}^{\mathcal{L},S_2}(d_1)$, where $S_2 = S \cup ((\mathbf{N}_C \cup \mathbf{N}_R) \setminus \text{sig}(\Psi_2))$.*

Proof. Let \mathcal{L} be any of the eight DLs listed in the proposition. We assume that \mathcal{L} contains inverse roles (the prove for DLs without inverse rules is simpler and is easily obtained from the proof below). Let \mathcal{I}_1 and \mathcal{I}_2 be interpretations satisfying condition (a). We use a standard construction from model theory. Consider the disjoint union \mathcal{I}_0 of \mathcal{I}_1 and \mathcal{I}_2 :

$$\mathcal{I}_0 = (\Delta^{\mathcal{I}_1} \cup \Delta^{\mathcal{I}_2}, A_1^{\mathcal{I}_0}, A_2^{\mathcal{I}_0}, \cdot^{\mathcal{I}_1}, \cdot^{\mathcal{I}_2}),$$

where we assume that $\Delta^{\mathcal{I}_1} \cap \Delta^{\mathcal{I}_2} = \emptyset$ and that $A_i^{\mathcal{I}_0} = \Delta^{\mathcal{I}_i}$ for fresh concept names A_i , $i = 1, 2$. Then \mathcal{I}_0 is elementarily equivalent to an interpretation \mathcal{I}'_0 , which is countably recursively saturated [14]. This means that, if every finite subset of a recursive set of first-order formulas is satisfied in \mathcal{I}'_0 , then the recursive set itself is satisfied in \mathcal{I}'_0 . We will not go into the details of this construction but will use the following consequences: by taking, instead of \mathcal{I}_1 and \mathcal{I}_2 , the corresponding substructures \mathcal{I}'_1 and \mathcal{I}'_2 (induced by $A_1^{\mathcal{I}'_0}$ and $A_2^{\mathcal{I}'_0}$) of \mathcal{I}'_0 , we obtain interpretations which still satisfy condition (a) of Proposition 8 and have, in addition, the following properties:

1. For each $d_1 \in \Delta^{\mathcal{I}'_1}$, there exists $d_2 \in \Delta^{\mathcal{I}'_2}$ such that $(\mathcal{I}'_1, d_1) \sim_{\mathcal{L}}^{\mathcal{I}'_0} (\mathcal{I}'_2, d_2)$ and vice versa;
2. if $(\mathcal{I}'_1, d_1) \sim_{\mathcal{L}}^{\mathcal{I}'_0} (\mathcal{I}'_2, d_2)$ and $(d_1, e_1) \in r^{\mathcal{I}'_1}$ with $r \in S$, then there exists $e_2 \in \Delta^{\mathcal{I}'_2}$ with $(d_2, e_2) \in r^{\mathcal{I}'_2}$ and $(\mathcal{I}'_1, e_1) \sim_{\mathcal{L}}^{\mathcal{I}'_0} (\mathcal{I}'_2, e_2)$;
3. if $(\mathcal{I}'_1, d_1) \sim_{\mathcal{L}}^{\mathcal{I}'_0} (\mathcal{I}'_2, d_2)$ and $(d_2, e_2) \in r^{\mathcal{I}'_2}$ with $r \in S$, then there exists $e_1 \in \Delta^{\mathcal{I}'_1}$ with $(d_1, e_1) \in r^{\mathcal{I}'_1}$ and $(\mathcal{I}'_1, e_1) \sim_{\mathcal{L}}^{\mathcal{I}'_0} (\mathcal{I}'_2, e_2)$;
4. if $(\mathcal{I}'_1, d_1) \sim_{\mathcal{L}}^{\mathcal{I}'_0} (\mathcal{I}'_2, d_2)$ and $r \in S \cup S^-$ and \mathcal{L} contains *number restrictions*, then there exists an S -invariant bijection between $\{d \mid (d_1, d) \in r^{\mathcal{I}'_1}\}$ and $\{e \mid (d_2, e) \in r^{\mathcal{I}'_2}\}$.

Intuitively, for Point 4 above, we need number restrictions in \mathcal{L} , because, without them, DLs are too weak to determine the number of r -successors of a node. In what follows, we use \mathcal{I}_1 and \mathcal{I}_2 to denote \mathcal{I}'_1 and \mathcal{I}'_2 , respectively. If condition (b) of Proposition 8 is satisfied, then take d_1 and e_1 satisfying (b). Otherwise, take arbitrary S -equivalent points $d_1 \in \Delta^{\mathcal{I}_1}$ and $e_1 \in \Delta^{\mathcal{I}_2}$, which exist by Point 1 above. We unravel the model \mathcal{I}_1 starting from d_1 as follows: take infinitely many copies d^i , $i \geq 0$, of each $d \in \Delta^{\mathcal{I}_1}$ and define \mathcal{J}_1 by taking as the domain $\Delta^{\mathcal{J}_1}$ the set of all finite sequences

$$(d_1, r_2, d_2^{i_2}, r_3, d_3^{i_3}, \dots, r_n, d_n^{i_n}),$$

where $d_i \in \Delta^{\mathcal{I}_1}$, $r_i \in \mathbf{N}_R \cup \mathbf{N}_R^- \cup \{\delta\}$ (with δ being some fresh “dummy” relation symbol) and $i_j \geq 0$ for $j \geq 2$, such that the following conditions hold:

- (i) $i_j = 0$ whenever $r_j \in \mathbf{N}_R \cup \mathbf{N}_R^-$ and \mathcal{L} contains qualified number restrictions;
- (ii) $(d_i, d_{i+1}) \in r_{i+1}^{\mathcal{I}_1}$ whenever $r_i \in \mathbf{N}_R \cup \mathbf{N}_R^-$;
- (iii) $d_i \neq d_{i+2}$ whenever $r_{i+1} = (r_{i+2})^-$ and \mathcal{L} contains qualified number restrictions.

The interpretation function $\cdot^{\mathcal{J}_1}$ of \mathcal{J}_1 is defined as follows:

- $(d_1, r_2, d_2^{i_2}, \dots, r_n, d_n^{i_n}) \in A^{\mathcal{J}_1}$ iff $d_n \in A^{\mathcal{I}_1}$, for all $A \in \mathbf{N}_C$;
- for all $r \in \mathbf{N}_R$, $r^{\mathcal{J}_1}$ consists of all pairs

$$((d_1, r_2, d_2^{i_2}, \dots, r_n, d_n^{i_n}), (d_1, r_2, d_2^{i_2}, \dots, r_n, d_n^{i_n}, r_{n+1}, d_n^{i_{n+1}})) \in \Delta^{\mathcal{J}_1} \times \Delta^{\mathcal{J}_1},$$

where $r_{n+1} = r$, and

$$((d_1, r_2, d_2^{i_2}, \dots, r_n, d_n^{i_n}, r_{n+1}, d_n^{i_{n+1}}), (d_1, r_2, d_2^{i_2}, \dots, r_n, d_n^{i_n})) \in \Delta^{\mathcal{J}_1} \times \Delta^{\mathcal{J}_1}$$

with $r_{n+1} = r^-$.

It is not difficult to show, that $(d_1, r_2, d_2^{i_2}, \dots, r_n, d_n^{i_n}) \sim_{\mathbf{N}_C \cup \mathbf{N}_R} d_n$, for every $d_n \in \Delta^{\mathcal{I}_1}$. Observe that conditions (i) and (iii) ensure that, if \mathcal{L} contains qualified number restrictions, then the number of r -successors of any point in $\Delta^{\mathcal{I}_1}$, $r \in \mathbf{N}_C \cup \mathbf{N}_C^-$, satisfying a certain set of concepts remains the same. In contrast, if \mathcal{L} does not contain qualified number restrictions, then we introduce infinitely many copies d^i of any r -successor. The reason is that later we want to amalgamate the unravellings of \mathcal{I}_1 and \mathcal{I}_2 , and, therefore, need the same number of r -successors satisfying the same \mathcal{L}_S -concepts in both unravellings. Construct \mathcal{J}_2 from \mathcal{I}_2 and the point e_1 in the same way.

We define an S -isomorphism ρ between \mathcal{J}_1 and \mathcal{J}_2 as the union of partial S -isomorphisms $\rho_0 \subseteq \rho_1 \subseteq \rho_2 \subseteq \dots$, where

- ρ_n is an S -isomorphism between the restrictions of \mathcal{J}_1 and \mathcal{J}_2 to the points of length not exceeding n ; i.e., points $(d_1, r_2, d_2^{i_2}, \dots, d_m^{i_m}) \in \Delta^{\mathcal{J}_1}$, and $(e_1, r_2, e_2^{i_2}, \dots, e_m^{i_m}) \in \Delta^{\mathcal{J}_2}$, where $m \leq n$;
- for each $w \in \text{dom}(\rho_n)$: $w \sim_S \rho_n(w)$.

The sequence of partial S -isomorphisms is inductively defined as follows. For the induction base, set $\rho_1((d_1)) = (e_1)$. Consider the induction step. Suppose that ρ_n has been defined. Assume $\rho_n(w_1) = w_2$, where

$$\begin{aligned} w_1 &= (d_1, r_2, d_2^{i_2}, \dots, r_n, d_n^{i_n}), \text{ and} \\ w_2 &= (e_1, s_2, e_2^{j_2}, \dots, s_n, e_n^{j_n}). \end{aligned}$$

Observe that $r_n \in S \cup S^-$ implies $s_n = r_n$ since ρ_n is a partial S -isomorphism. Now ρ_{n+1} is defined by adding, for each (w_1, w_2) , the following pairs to ρ_n :

- If $r \in S \cup S^-$ with $r_n \neq r^-$ and \mathcal{L} contains qualified number restrictions, we can take (by Point 4 above) an S -invariant bijection b between the $r^{\mathcal{I}_1}$ -successors of d_n and the $r^{\mathcal{I}_2}$ -successors of e_n and extend ρ_n with the set

$$\{((w_1, r, d^0), (w_2, r, b(d)^0) \mid (d_n, d) \in r^{\mathcal{I}_1}\}.$$

- If $r \in S \cup S^-$ with $r_n = r^-$ and \mathcal{L} -contains qualified number restrictions, consider the sets

$$B_1 = \{d \mid (d_n, d) \in r^{\mathcal{I}_1}\} \setminus \{d_{n-1}\} \text{ and } B_2 = \{e \mid (e_n, e) \in r^{\mathcal{I}_2}\} \setminus \{e_{n-1}\}.$$

We have $d_{n-1} \sim_S e_{n-1}$. Hence, by Point 4 above, there exists an S -invariant bijection b between B_1 and B_2 . Extend ρ_n with the set

$$\{((w_1, r, d^0), (w_2, r, b(d)^0) \mid d \in B_1\}.$$

- If $r \in S \cup S^-$ and \mathcal{L} does not contain qualified number restrictions, we find (by Points 2 and 3) a bijection b between the sets

$$B_1 = \bigcup_{i \geq 0} \{d^i \mid (d_n, d) \in r^{\mathcal{I}_1}\} \text{ and } B_2 = \bigcup_{i \geq 0} \{e^i \mid (e_n, e) \in r^{\mathcal{I}_2}\}$$

such that $b(d^i) = e^j$ implies $d \sim_S e$. Extend ρ_n with the set

$$\{(w_1, r, d), (w_2, r, b(d)) \mid d \in B_1\}.$$

- Finally, take an S -invariant bijection b between the remaining points $(w_1, r_{n+1}, d_{n+1}^i)$ of $\Delta^{\mathcal{J}_1}$ and $(w_2, s_{n+1}, e_{n+1}^i)$ of $\Delta^{\mathcal{J}_2}$. This is possible, because of Point 1 above and the introduction of the dummy relation symbol δ , which ensures that, for each $d \in \Delta^{\mathcal{I}_1}$, there are infinitely many $(w_1, \delta, d^i) \in \Delta^{\mathcal{J}_1}$ with $d \sim_{\xi}^{\mathcal{L}} (w_1, \delta, d^i)$, and, correspondingly, for $e \in \Delta^{\mathcal{J}_2}$. Add b to ρ_n .

The mapping $\rho = \bigcup_{n \geq 1} \rho_n$ is an S -isomorphism between \mathcal{J}_1 and \mathcal{J}_2 . The required model \mathcal{I} is now constructed by taking the model \mathcal{J}_1 but interpreting the $A \in \mathbf{N}_C \cap \text{sig}(\Psi_2)$ as $\rho^{-1}(A^{\mathcal{J}_2})$ and the $r \in \mathbf{N}_R \cap \text{sig}(\Psi_2)$ as $\rho^{-1}(r^{\mathcal{J}_2})$. \square

Proposition 9. *Let \mathcal{L} be any of the languages ALC , $ALCQ$, $ALCI$, $ALCQI$, $ALCU$, $ALCQU$, $ALCIU$, $ALCQIU$. Then $\mathcal{QL}_{\mathcal{L}}$ has interpolation.*

Proof. Let \mathcal{L} be any of the DLs listed in the proposition. Assume that there exists a set Ψ of $\mathcal{QL}_{\mathcal{L}}$ -sentences and \mathcal{L} -concepts C_0, D_0 with $\Psi \models C_0 \sqsubseteq D_0$ such that there does not exist an interpolant $I(\Psi, C_0 \sqsubseteq D_0)$. Let $S = \text{sig}(\Psi) \cap \text{sig}(C_0 \sqsubseteq D_0)$ and

$$\Psi_S = \{C \sqsubseteq D \mid \Psi \models C \sqsubseteq D, \text{sig}(C \sqsubseteq D) \subseteq S\}.$$

Then $\Psi \models \Psi_S$ and $\Psi_S \not\models C_0 \sqsubseteq D_0$, by compactness. Take a model \mathcal{I}'_2 of Ψ_S with $e_1 \in (C_0 \sqcap \neg D_0)^{\mathcal{I}'_2}$. Take a model \mathcal{I}'_1 of Ψ containing a point d_1 such that $(\mathcal{I}'_1, d_1) \sim_{\xi}^{\mathcal{L}} (\mathcal{I}'_2, e_1)$. The existence of such a model follows again by compactness. Now the proof splits into two parts.

(i) Assume first that \mathcal{L} contains the universal role. Then, for all \mathcal{L}_S -concepts C , $C^{\mathcal{I}'_2} = \emptyset$ iff $C^{\mathcal{I}'_1} = \emptyset$, because $d_1 \in (\exists u.C)^{\mathcal{I}'_1}$ iff $e_1 \in (\exists u.C)^{\mathcal{I}'_2}$. It follows that, for $\Psi_1 = \Psi$ and $\Psi_2 = \emptyset$, the models \mathcal{I}'_1 and \mathcal{I}'_2 and points d_1 and e_1 satisfy the conditions of Proposition 8. We obtain a model \mathcal{I} of Ψ such that $(C_0 \sqcap \neg D_0)^{\mathcal{I}} \neq \emptyset$. Hence $\Psi \not\models C_0 \sqsubseteq D_0$ and we have derived a contradiction.

(ii) Assume now that \mathcal{L} does not contain the universal role. Let

$$\mathcal{C} = \{C \mid C \text{ a } \mathcal{L}_S\text{-concept with } \Psi \not\models C \sqsubseteq \perp\}$$

and take for each $C \in \mathcal{C}$ a model \mathcal{I}_C of Ψ such that $C^{\mathcal{I}_C} \neq \emptyset$. Let \mathcal{I}_1 be the disjoint union of the models \mathcal{I}'_1 and \mathcal{I}_C , $C \in \mathcal{C}$. Correspondingly, let \mathcal{I}_2 be the disjoint union of the models \mathcal{I}'_2 and \mathcal{I}_C , $C \in \mathcal{C}$. Again, $\mathcal{I}_1, \mathcal{I}_2, d_1, e_1$ satisfy the conditions of Proposition 8 for $\Psi_1 = \Psi$ and $\Psi_2 = \emptyset$. Hence there exists a model \mathcal{I} of Ψ with $(C_0 \sqcap \neg D_0)^{\mathcal{I}} \neq \emptyset$ and we have derived a contradiction. \square

Proposition 10. *Let \mathcal{L} be any of the languages \mathcal{ALC} , \mathcal{ALCQ} , \mathcal{ALCI} , \mathcal{ALCQI} . Then $(\mathcal{L}, \mathcal{QL}_{\mathcal{L}})$ is robust under joins.*

Proof. Fix a DL \mathcal{L} from the proposition. Let \mathcal{O}_1 and \mathcal{O}_2 be \mathcal{L} -ontologies. Assume \mathcal{O}_1 and \mathcal{O}_2 are S -inseparable w.r.t. $\mathcal{QL}_{\mathcal{L}}$ with $\text{sig}(\mathcal{O}_1) \cap \text{sig}(\mathcal{O}_2) \subseteq S$. Let, for $i = 1, 2$,

$$\mathcal{C}_i = \{C \mid C \text{ a } \mathcal{L}_S\text{-concept with } \mathcal{O}_i \not\models C \sqsubseteq \perp\}.$$

By S -inseparability, $\mathcal{C}_1 = \mathcal{C}_2$. Take, for each $C \in \mathcal{C}_1$, a model \mathcal{I}_C^1 of \mathcal{O}_1 such that $C^{\mathcal{I}_C^1} \neq \emptyset$ and let \mathcal{I}_1 be the disjoint union of the models \mathcal{I}_C^1 , $C \in \mathcal{C}_1$. Similarly, take, for each $C \in \mathcal{C}_1$, a model \mathcal{I}_C^2 of \mathcal{O}_2 such that $C^{\mathcal{I}_C^2} \neq \emptyset$ and let \mathcal{I}_2 be the disjoint union of the models \mathcal{I}_C^2 , $C \in \mathcal{C}_1$. Then \mathcal{I}_1 and \mathcal{I}_2 satisfy the conditions of Proposition 8 for $\Psi_1 = \mathcal{O}_1$ and $\Psi_2 = \mathcal{O}_2$. Hence, there exists a model \mathcal{I} of $\mathcal{O}_1 \cup \mathcal{O}_2$ such that $C^{\mathcal{I}} \neq \emptyset$ whenever $C \in \mathcal{C}_1$. It follows that $\mathcal{O}_1 \cup \mathcal{O}_2$ and \mathcal{O}_i are S -inseparable w.r.t. $\mathcal{QL}_{\mathcal{L}}$, $i = 1, 2$. \square

Index

acyclic ontology, 32

conservative extension, 9

definitorial extension, 11

forgetting, 30

inseparability, 9

interpolation, 25

lightweight description logics, 32

Robinson Joint Consistency, 25

uniform interpolation, 30

variable elimination, 30