Journal of Artificial Intelligence Research 31 (2008) 273-318 Submitted 07/07; published 02/08

Modular Reuse of Ontologies: Theory and Practice

Bernardo Cuenca Grau BERG@COMLAB.OX.AC.UK
Ian Horrocks IAN.HORROCKS@COMLAB.OX.AC.UK
Yevgeny Kazakov YEVGENY.KAZAKOV@COMLAB.OX.AC.UK
Oxford University Computing Laboratory
Oxford, OX1 3QD, UK
Ulrike Sattler SATTLER@CS.MAN.AC.UK
School of Computer Science
The University of Manchester
Manchester, M13 9PL, UK

Abstract

In this paper, we propose a set of tasks that are relevant for the modular reuse of ontologies. In order to formalize these tasks as reasoning problems, we introduce the notions of *conservative extension*, *safety* and *module* for a very general class of logic-based ontology languages. We investigate the general properties of and relationships between these notions and study the relationships between the relevant reasoning problems we have previously identified. To study the computability of these problems, we consider, in particular, Description Logics (DLs), which provide the formal underpinning of the W3C Web Ontology Language (OWL), and show that all the problems we consider are undecidable or algorithmically unsolvable for the description logic underlying OWL DL. In order to achieve a practical solution, we identify conditions sufficient for an ontology to reuse a set of symbols "safely"—that is, without changing their meaning. We provide the notion of a *safety class*, which characterizes any sufficient condition for safety, and identify a family of safety classes–called locality—which enjoys a collection of desirable properties. We use the notion of a safety class to extract modules from ontologies, and we provide various modularization algorithms that are appropriate to the properties of the particular safety class in use. Finally, we show practical benefits of our safety checking and module extraction algorithms.

1. Motivation

Ontologies—conceptualizations of a domain shared by a community of users—play a major role in the Semantic Web, and are increasingly being used in knowledge management systems, e-Science, bio-informatics, and Grid applications (Staab & Studer, 2004).

The design, maintenance, reuse, and integration of ontologies are complex tasks. Like software engineers, ontology engineers need to be supported by tools and methodologies that help them to minimize the introduction of errors, i.e., to ensure that ontologies are consistent and do not have unexpected consequences. In order to develop this support, important notions from software engineering, such as *module*, *black-box behavior*, and *controlled interaction*, need to be adapted.

Recently, there has been growing interest in the topic of modularity in ontology engineering (Seidenberg & Rector, 2006; Noy, 2004a; Lutz, Walther, & Wolter, 2007; Cuenca Grau, Parsia, Sirin, & Kalyanpur, 2006b; Cuenca Grau, Horrocks, Kazakov, & Sattler, 2007), which has been motivated by the above-mentioned application needs. In this paper, we focus on the use of modularity to support the *partial reuse* of ontologies. In particular, we consider the scenario in which we are developing an ontology $\mathcal{P}$ and want to reuse a set $\mathbf{S}$ of symbols from a "foreign" ontology $\mathcal{Q}$ without changing their meaning.

For example, suppose that an ontology engineer is building an ontology about research projects, which specifies different types of projects according to the research topic they focus on. The ontology engineer in charge of the projects ontology $\mathcal{P}$ may use terms such as Cystic_Fibrosis and Genetic_Disorder in his descriptions of medical research projects. The ontology engineer is an expert on research projects; he may be unfamiliar, however, with most of the topics the projects cover and, in particular, with the terms Cystic_Fibrosis and Genetic_Disorder. In order to complete the projects ontology with suitable definitions for these medical terms, he decides to reuse the knowledge about these subjects from a well-established medical ontology $\mathcal{Q}$.

The most straightforward way to reuse these concepts is to construct the logical union $\mathcal{P} \cup \mathcal{Q}$ of the axioms in $\mathcal{P}$ and $\mathcal{Q}$. It is reasonable to assume that the additional knowledge about the medical terms used in both $\mathcal{P}$ and $\mathcal{Q}$ will have implications on the meaning of the projects defined in $\mathcal{P}$; indeed, the additional knowledge about the reused terms provides new information about medical research projects which are defined using these medical terms. Less intuitive is the fact that importing $\mathcal{Q}$ may also result in new entailments concerning the reused symbols, namely Cystic_Fibrosis and Genetic_Disorder. Since the ontology engineer of the projects ontology is not an expert in medicine and relies on the designers of $\mathcal{Q}$, it is to be expected that the meaning of the reused symbols is completely specified in $\mathcal{Q}$; that is, the fact that these symbols are used in the projects ontology $\mathcal{P}$ should not imply that their original "meaning" in $\mathcal{Q}$ changes. If $\mathcal{P}$ does not change the meaning of these symbols in $\mathcal{Q}$, we say that $\mathcal{P} \cup \mathcal{Q}$ is a *conservative extension* of $\mathcal{Q}$

In realistic application scenarios, it is often unreasonable to assume the foreign ontology $\mathcal{Q}$ to be fixed; that is, $\mathcal{Q}$ may evolve beyond the control of the modelers of $\mathcal{P}$. The ontology engineers in charge of $\mathcal{P}$ may not be authorized to access all the information in $\mathcal{Q}$ or, most importantly, they may decide at a later time to reuse the symbols Cystic_Fibrosis and Genetic_Disorder from a medical ontology other than $\mathcal{Q}$. For application scenarios in which the external ontology $\mathcal{Q}$ may change, it is reasonable to "abstract" from the particular $\mathcal{Q}$ under consideration. In particular, given a set $\mathbf{S}$ of "external symbols", the fact that the axioms in $\mathcal{P}$ do not change the meaning of any symbol in $\mathbf{S}$ should be independent of the particular meaning ascribed to these symbols by $\mathcal{Q}$. In that case, we will say that $\mathcal{P}$ is *safe* for $\mathbf{S}$.

Moreover, even if $\mathcal{P}$ "safely" reuses a set of symbols from an ontology $\mathcal{Q}$, it may still be the case that $\mathcal{Q}$ is a large ontology. In particular, in our example, the foreign medical ontology may be huge, and importing the whole ontology would make the consequences of the additional information costly to compute and difficult for our ontology engineers in charge of the projects ontology (who are not medical experts) to understand. In practice, therefore, one may need to extract a *module* $\mathcal{Q}_1$ of $\mathcal{Q}$ that includes only the relevant information. Ideally, this module should be as small as possible while still guarantee to capture the meaning of the terms used; that is, when answering queries against the research projects ontology, importing the module $\mathcal{Q}_1$ would give exactly the same answers as if the whole medical ontology $\mathcal{Q}$ had been imported. In this case, importing the module will have the same observable effect on the projects ontology as importing the entire ontology. Furthermore, the fact that $\mathcal{Q}_1$ is a module in $\mathcal{Q}$ should be independent of the particular $\mathcal{P}$ under consideration.

The contributions of this paper are as follows:

1. We propose a set of tasks that are relevant to ontology reuse and formalize them as reasoning problems. To this end, we introduce the notions of *conservative extension*, *safety* and *module* for a very general class of logic-based ontology languages.

2. We investigate the general properties of and relationships between the notions of conservative extension, safety, and module and use these properties to study the relationships between the relevant reasoning problems we have previously identified.

3. We consider Description Logics (DLs), which provide the formal underpinning of the W3C Web Ontology Language (OWL), and study the computability of our tasks. We show that all the tasks we consider are undecidable or algorithmically unsolvable for the description logic underlying OWL DL—the most expressive dialect of OWL that has a direct correspondence to description logics.

4. We consider the problem of deciding safety of an ontology for a signature. Given that this problem is undecidable for OWL DL, we identify sufficient conditions for safety, which are decidable for OWL DL—that is, if an ontology satisfies our conditions then it is safe; the converse, however, does not necessarily hold. We propose the notion of a *safety class*, which characterizes any sufficiency condition for safety, and identify a family of safety classes–called locality—which enjoys a collection of desirable properties.

5. We next apply the notion of a safety class to the task of extracting modules from ontologies; we provide various modularization algorithms that are appropriate to the properties of the particular safety class in use.

6. We present empirical evidence of the practical benefits of our techniques for safety checking and module extraction.

This paper extends the results in our previous work (Cuenca Grau, Horrocks, Kutz, & Sattler, 2006; Cuenca Grau et al., 2007; Cuenca Grau, Horrocks, Kazakov, & Sattler, 2007).

2. Preliminaries

In this section we introduce description logics (DLs) (Baader, Calvanese, McGuinness, Nardi, & Patel-Schneider, 2003), a family of knowledge representation formalisms which underlie modern ontology languages, such as OWL DL (Patel-Schneider, Hayes, & Horrocks, 2004). A hierarchy of commonly-used description logics is summarized in Table 1.

The *syntax* of a description logic $\mathcal{L}$ is given by a signature and a set of constructors. A *signature* (or *vocabulary*) **Sg** of a DL is the (disjoint) union of countably infinite sets **AC** of *atomic concepts* $(A, B, \dots)$ representing sets of elements, **AR** of *atomic roles* $(r, s, \dots)$ representing binary relations between elements, and **Ind** of *individuals* $(a, b, c, \dots)$ representing constants. We assume the signature to be fixed for every DL.

DLs	Constructors		Axioms [**Ax**]		
	Rol	**Con**	RBox	TBox	ABox
$\mathcal{EL}$	r	$\top$, A, $C_1 \sqcap C_2$, $\exists R.C$		$A \equiv C$, $C_1 \sqsubseteq C_2$	$a : C$, $r(a, b)$
$\mathcal{ALC}$	–"–	–"–, $\neg C$		–"–	–"–
$\mathcal{S}$	–"–	–"–	$\mathsf{Trans}(r)$	–"–	–"–
+ $\mathcal{I}$	r^-				
+ $\mathcal{H}$			$R_1 \sqsubseteq R_2$		
+ $\mathcal{F}$			$\mathsf{Funct}(R)$		
+ $\mathcal{N}$		$(\geqslant n\, S)$			
+ $\mathcal{Q}$		$(\geqslant n\, S.C)$			
+ $\mathcal{O}$		$\{i\}$			

Here $r \in \mathbf{AR}$, $A \in \mathbf{AC}$, $a, b \in \mathbf{Ind}$, $R_{(i)} \in \mathbf{Rol}$, $C_{(i)} \in \mathbf{Con}$, $n \geq 1$ and $S \in \mathbf{Rol}$ is a simple role (see (Horrocks & Sattler, 2005)).

Table 1: The hierarchy of standard description logics

Every DL provides *constructors* for defining the set **Rol** of (general) *roles* $(R, S, \dots)$, the set **Con** of (general) *concepts* $(C, D, \dots)$, and the set **Ax** of *axioms* $(\alpha, \beta, \dots)$ which is a union of *role axioms* (RBox), *terminological axioms* (TBox) and *assertions* (ABox).

$\mathcal{EL}$ (Baader, Brandt, & Lutz, 2005) is a simple description logic which allows one to construct complex concepts using *conjunction* $C_1 \sqcap C_2$ and *existential restriction* $\exists R.C$ starting from atomic concepts A, roles R and the *top concept* $\top$. $\mathcal{EL}$ provides no role constructors and no role axioms; thus, every role R in $\mathcal{EL}$ is atomic. The TBox axioms of $\mathcal{EL}$ can be either *concept definitions* $A \equiv C$ or *general concept inclusion axioms* (GCIs) $C_1 \sqsubseteq C_2$. $\mathcal{EL}$ assertions are either *concept assertions* $a : C$ or *role assertions* $r(a, b)$. In this paper we assume the concept definition $A \equiv C$ is an abbreviation for two GCIs $A \sqsubseteq C$ and $C \sqsubseteq A$.

The basic description logic $\mathcal{ALC}$ (Schmidt-Schauß & Smolka, 1991) is obtained from $\mathcal{EL}$ by adding the *concept negation* constructor $\neg C$. We introduce some additional constructors as abbreviations: the *bottom concept* $\bot$ is a shortcut for $\neg\top$, the *concept disjunction* $C_1 \sqcup C_2$ stands for $\neg(\neg C_1 \sqcap \neg C_2)$, and the *value restriction* $\forall R.C$ stands for $\neg(\exists R.\neg C)$. In contrast to $\mathcal{EL}$, $\mathcal{ALC}$ can express *contradiction axioms* like $\top \sqsubseteq \bot$. The logic $\mathcal{S}$ is an extension of $\mathcal{ALC}$ where, additionally, some atomic roles can be declared to be *transitive* using a role axiom $\mathsf{Trans}(r)$.

Further extensions of description logics add features such as *inverse roles* r^- (indicated by appending a letter $\mathcal{I}$ to the name of the logic), *role inclusion axioms* (RIs) $R_1 \sqsubseteq R_2$ $(+\mathcal{H})$, *functional roles* $\mathsf{Funct}(R)$ $(+\mathcal{F})$, *number restrictions* $(\geqslant n\, S)$, with $n \geq 1$, $(+\mathcal{N})$, *qualified number restrictions* $(\geqslant n\, S.C)$, with $n \geq 1$, $(+\mathcal{Q})$[1], and *nominals* $\{a\}$ $(+\mathcal{O})$. Nominals make it possible to construct a concept representing a singleton set $\{a\}$ (a *nominal* concept) from an individual a. These extensions can be used in different combinations; for example $\mathcal{ALCO}$ is an extension of $\mathcal{ALC}$ with nominals; $\mathcal{SHIQ}$ is an extension of $\mathcal{S}$ with role hierarchies,

1. the dual constructors $(\leqslant n\, S)$ and $(\leqslant n\, S.C)$ are abbreviations for $\neg(\geqslant n + 1\, S)$ and $\neg(\geqslant n + 1\, S.\neg C)$, respectively

inverse roles and qualified number restrictions; and $\mathcal{SHOIQ}$ is the DL that uses all the constructors and axiom types we have presented.

Modern ontology languages, such as OWL, are based on description logics and, to a certain extent, are syntactic variants thereof. In particular, OWL DL corresponds to $\mathcal{SHOIN}$ (Horrocks, Patel-Schneider, & van Harmelen, 2003). In this paper, we assume an *ontology $\mathcal{O}$ based on a description logic $\mathcal{L}$* to be a finite set of axioms in $\mathcal{L}$. The *signature of an ontology $\mathcal{O}$ (of an axiom α)* is the set $\mathsf{Sig}(\mathcal{O})$ ($\mathsf{Sig}(\alpha)$) of atomic concepts, atomic roles and individuals that occur in $\mathcal{O}$ (respectively in α).

The main reasoning task for ontologies is *entailment*: given an ontology $\mathcal{O}$ and an axiom α, check if $\mathcal{O}$ implies α. The logical entailment $\models$ is defined using the *usual Tarski-style set-theoretic semantics* for description logics as follows. An *interpretation* $\mathcal{I}$ is a pair $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where $\Delta^\mathcal{I}$ is a non-empty set, called the *domain* of the interpretation, and $\cdot^\mathcal{I}$ is the *interpretation function* that assigns: to every $A \in \mathbf{AC}$ a subset $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$, to every $r \in \mathbf{AR}$ a binary relation $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$, and to every $a \in \mathbf{Ind}$ an element $a^\mathcal{I} \in \Delta^\mathcal{I}$. Note that the sets $\mathbf{AC}$, $\mathbf{AR}$ and $\mathbf{Ind}$ are not defined by the interpretation $\mathcal{I}$ but assumed to be fixed for the ontology language (DL).

The interpretation function $\cdot^\mathcal{I}$ is extended to complex roles and concepts via DL-constructors as follows:

$$
\begin{array}{rcl}
(\top)^\mathcal{I} & = & \Delta \\
(C \sqcap D)^\mathcal{I} & = & C^\mathcal{I} \cap D^\mathcal{I} \\
(\exists R.C)^\mathcal{I} & = & \{x \in \Delta^\mathcal{I} \mid \exists y.\langle x,y\rangle \in R^\mathcal{I} \wedge y \in C^\mathcal{I}\} \\
(\neg C)^\mathcal{I} & = & \Delta^\mathcal{I} \setminus C^\mathcal{I} \\
(r^-)^\mathcal{I} & = & \{\langle x,y\rangle \mid \langle y,x\rangle \in r^\mathcal{I}\} \\
(\geqslant n\, R)^\mathcal{I} & = & \{\, x \in \Delta^\mathcal{I} \mid\ \sharp\{y \in \Delta^\mathcal{I} \mid \langle x,y\rangle \in R^\mathcal{I}\} \geq n\,\} \\
(\geqslant n\, R.C)^\mathcal{I} & = & \{\, x \in \Delta^\mathcal{I} \mid\ \sharp\{y \in \Delta^\mathcal{I} \mid \langle x,y\rangle \in R^\mathcal{I} \wedge y \in C^\mathcal{I}\} \geq n\,\} \\
\{a\}^\mathcal{I} & = & \{a^\mathcal{I}\}
\end{array}
$$

The *satisfaction* relation $\mathcal{I} \models \alpha$ between an interpretation $\mathcal{I}$ and a DL axiom α (read as $\mathcal{I}$ *satisfies* α, or $\mathcal{I}$ is a *model* of α) is defined as follows:

$$
\begin{array}{ll}
\mathcal{I} \models C_1 \sqsubseteq C_2 \ \text{ iff } \ C_1^\mathcal{I} \subseteq C_2^\mathcal{I}; & \mathcal{I} \models a : C \ \text{ iff } \ a^\mathcal{I} \in C^\mathcal{I}; \\
\mathcal{I} \models R_1 \sqsubseteq R_2 \ \text{ iff } \ R_1^\mathcal{I} \subseteq R_2^\mathcal{I}\ ; & \mathcal{I} \models r(a,b) \ \text{ iff } \ \langle a^\mathcal{I}, b^\mathcal{I}\rangle \in r^\mathcal{I};
\end{array}
$$

$$\mathcal{I} \models \mathsf{Trans}(r) \ \text{ iff } \ \forall x,y,z \in \Delta^\mathcal{I}[\,\langle x,y\rangle \in r^\mathcal{I} \wedge \langle y,z\rangle \in r^\mathcal{I} \Rightarrow \langle x,z\rangle \in r^\mathcal{I}\,];$$

$$\mathcal{I} \models \mathsf{Funct}(R) \ \text{ iff } \ \forall x,y,z \in \Delta^\mathcal{I}[\,\langle x,y\rangle \in R^\mathcal{I} \wedge \langle x,z\rangle \in R^\mathcal{I} \Rightarrow y = z\,];$$

An interpretation $\mathcal{I}$ is a *model* of an ontology $\mathcal{O}$ if $\mathcal{I}$ satisfies all axioms in $\mathcal{O}$. An ontology $\mathcal{O}$ *implies* an axiom α (written $\mathcal{O} \models \alpha$) if $\mathcal{I} \models \alpha$ for every model $\mathcal{I}$ of $\mathcal{O}$. Given a set $\mathbf{I}$ of interpretations, we say that an axiom α (an ontology $\mathcal{O}$) is *valid in* $\mathbf{I}$ if every interpretation $\mathcal{I} \in \mathbf{I}$ is a model of α (respectively $\mathcal{O}$). An axiom α is a *tautology* if it is valid in the set of all interpretations (or, equivalently, is implied by the empty ontology).

We say that two interpretations $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ and $\mathcal{J} = (\Delta^\mathcal{J}, \cdot^\mathcal{J})$ *coincide on the subset* $\mathbf{S}$ *of the signature* (notation: $\mathcal{I}|_\mathbf{S} = \mathcal{J}|_\mathbf{S}$) if $\Delta^\mathcal{I} = \Delta^\mathcal{J}$ and $X^\mathcal{I} = X^\mathcal{J}$ for every $X \in \mathbf{S}$. We say that two sets of interpretations $\mathbf{I}$ and $\mathbf{J}$ *are equal modulo* $\mathbf{S}$ (notation: $\mathbf{I}|_\mathbf{S} = \mathbf{J}|_\mathbf{S}$) if for every $\mathcal{I} \in \mathbf{I}$ there exits $\mathcal{J} \in \mathbf{J}$ such that $\mathcal{J}|_\mathbf{S} = \mathcal{I}|_\mathbf{S}$ and for every $\mathcal{J} \in \mathbf{J}$ there exists $\mathcal{I} \in \mathbf{I}$ such that $\mathcal{I}|_\mathbf{S} = \mathcal{J}|_\mathbf{S}$.

	Ontology of medical research projects $\mathcal{P}$:
P1	$\mathsf{Genetic\_Disorder\_Project} \equiv \mathsf{Project} \sqcap \exists \mathsf{has\_Focus}.\underline{\mathsf{Genetic\_Disorder}}$
P2	$\mathsf{Cystic\_Fibrosis\_EUProject} \equiv \mathsf{EUProject} \sqcap \exists \mathsf{has\_Focus}.\underline{\mathsf{Cystic\_Fibrosis}}$
P3	$\mathsf{EUProject} \sqsubseteq \mathsf{Project}$
P4	$\exists \mathsf{has\_Focus}.\top \sqsubseteq \mathsf{Project}$
E1	$\mathsf{Project} \sqcap (\underline{\mathsf{Genetic\_Disorder}} \sqcap \underline{\mathsf{Cystic\_Fibrosis}}) \sqsubseteq \bot$
E2	$\forall \mathsf{has\_Focus}.\underline{\mathsf{Cystic\_Fibrosis}} \sqsubseteq \exists \mathsf{has\_Focus}.\underline{\mathsf{Genetic\_Disorder}}$

	Ontology of medical terms $\mathcal{Q}$:
M1	$\underline{\mathsf{Cystic\_Fibrosis}} \equiv \mathsf{Fibrosis} \sqcap \exists \mathsf{located\_In}.\mathsf{Pancreas} \sqcap \exists \mathsf{has\_Origin}.\mathsf{Genetic\_Origin}$
M2	$\mathsf{Genetic\_Fibrosis} \equiv \mathsf{Fibrosis} \sqcap \exists \mathsf{has\_Origin}.\mathsf{Genetic\_Origin}$
M3	$\mathsf{Fibrosis} \sqcap \exists \mathsf{located\_In}.\mathsf{Pancreas} \sqsubseteq \mathsf{Genetic\_Fibrosis}$
M4	$\mathsf{Genetic\_Fibrosis} \sqsubseteq \underline{\mathsf{Genetic\_Disorder}}$
M5	$\mathsf{DEFBI\_Gene} \sqsubseteq \mathsf{Immuno\_Protein\_Gene} \sqcap \exists \mathsf{associated\_With}.\underline{\mathsf{Cystic\_Fibrosis}}$

Figure 1: Reusing medical terminology in an ontology on research projects

3. Ontology Integration and Knowledge Reuse

In this section, we elaborate on the ontology reuse scenario sketched in Section 1. Based on this application scenario, we motivate and define the reasoning tasks to be investigated in the remainder of the paper. In particular, our tasks are based on the notions of a *conservative extension* (Section 3.2), *safety* (Sections 3.2 and 3.3) and *module* (Section 3.4). These notions are defined relative to a language $\mathcal{L}$. Within this section, we assume that $\mathcal{L}$ is an ontology language based on a description logic; in Section 3.6, we will define formally the class of ontology languages for which the given definitions of conservative extensions, safety and modules apply. For the convenience of the reader, all the tasks we consider in this paper are summarized in Table 2.

3.1 A Motivating Example

Suppose that an ontology engineer is in charge of a $\mathcal{SHOIQ}$ ontology on research projects, which specifies different types of projects according to the research topic they are concerned with. Assume that the ontology engineer defines two concepts—Genetic_Disorder_Project and Cystic_Fibrosis_EUProject—in his ontology $\mathcal{P}$. The first one describes projects about genetic disorders; the second one describes European projects about cystic fibrosis, as given by the axioms P1 and P2 in Figure 1. The ontology engineer is an expert on research projects: he knows, for example, that every instance of EUProject must be an instance of Project (the concept-inclusion axiom P3) and that the role has_Focus can be applied only to instances of Project (the domain axiom P4). He may be unfamiliar, however, with most of the topics the projects cover and, in particular, with the terms Cystic_Fibrosis and Genetic_Disorder mentioned in P1 and P2. In order to complete the projects ontology with suitable definitions for these medical terms, he decides to reuse the knowledge about these subjects from a well-established and widely-used medical ontology.

Suppose that Cystic_Fibrosis and Genetic_Disorder are described in an ontology $\mathcal{Q}$ containing the axioms M1-M5 in Figure 1. The most straightforward way to reuse these concepts is to import in $\mathcal{P}$ the ontology $\mathcal{Q}$—that is, to add the axioms from $\mathcal{Q}$ to the axioms in $\mathcal{P}$ and work with the extended ontology $\mathcal{P} \cup \mathcal{Q}$. Importing additional axioms into an ontology may result in new logical consequences. For example, it is easy to see that axioms M1–M4 in $\mathcal{Q}$ imply that every instance of Cystic_Fibrosis is an instance of Genetic_Disorder:

$$\mathcal{Q} \models \alpha := (\mathsf{Cystic\_Fibrosis} \sqsubseteq \mathsf{Genetic\_Disorder}) \quad (1)$$

Indeed, the concept inclusion $\alpha_1 := (\mathsf{Cystic\_Fibrosis} \sqsubseteq \mathsf{Genetic\_Fibrosis})$ follows from axioms M1 and M2 as well as from axioms M1 and M3; α follows from axioms α_1 and M4.

Using inclusion α from (1) and axioms P1–P3 from ontology $\mathcal{P}$ we can now prove that every instance of Cystic_Fibrosis_EUProject is also an instance of Genetic_Disorder_Project:

$$\mathcal{P} \cup \mathcal{Q} \models \beta := (\mathsf{Cystic\_Fibrosis\_EUProject} \sqsubseteq \mathsf{Genetic\_Disorder\_Project}) \quad (2)$$

This inclusion β, however, does not follow from $\mathcal{P}$ alone—that is, $\mathcal{P} \not\models \beta$. The ontology engineer might be not aware of Entailment (2), even though it concerns the terms of primary scope in his projects ontology $\mathcal{P}$.

It is natural to expect that entailments like α in (1) from an imported ontology $\mathcal{Q}$ result in new logical consequences, like β, in (2), over the terms defined in the main ontology $\mathcal{P}$. One would not expect, however, that the meaning of the terms defined in $\mathcal{Q}$ changes as a consequence of the import since these terms are supposed to be completely specified within $\mathcal{Q}$. Such a side effect would be highly undesirable for the modeling of ontology $\mathcal{P}$ since the ontology engineer of $\mathcal{P}$ might not be an expert on the subject of $\mathcal{Q}$ and is not supposed to alter the meaning of the terms defined in $\mathcal{Q}$ not even implicitly.

The meaning of the reused terms, however, might change during the import, perhaps due to modeling errors. In order to illustrate such a situation, suppose that the ontology engineer has learned about the concepts Genetic_Disorder and Cystic_Fibrosis from the ontology $\mathcal{Q}$ (including the inclusion (1)) and has decided to introduce additional axioms formalizing the following statements:

"Every instance of Project *is different from every instance of* Genetic_Disorder *and every instance of* Cystic_Fibrosis*."* (3)

"Every project that has_Focus *on* Cystic_Fibrosis*, also* has_Focus *on* Genetic_Disorder*"* (4)

Note that the statements (3) and (4) can be thought of as adding new information about projects and, intuitively, they should not change or constrain the meaning of the medical terms.

Suppose the ontology engineer has formalized the statements (3) and (4) in ontology $\mathcal{P}$ using axioms E1 and E2 respectively. At this point, the ontology engineer has introduced modeling errors and, as a consequence, axioms E1 and E2 do not correspond to (3) and (4): E1 actually formalizes the following statement: *"Every instance of* Project *is different from every common instance of* Genetic_Disorder *and* Cystic_Fibrosis*"*, and E2 expresses that *"Every object that either* $\mathsf{has\_Focus}$ *on nothing, or* $\mathsf{has\_Focus}$ *only on* $\mathsf{Cystic\_Fibrosis}$*, also* $\mathsf{has\_Focus}$ *on* $\mathsf{Genetic\_Disorder}$*"*. These kinds of modeling errors are difficult to detect, especially when they do not cause inconsistencies in the ontology.

Note that, although axiom E1 does not correspond to fact (3), it is still a consequence of (3) which means that it should not constrain the meaning of the medical terms. On the other hand, E2 is not a consequence of (4) and, in fact, it constrains the meaning of medical terms. Indeed, the axioms E1 and E2 together with axioms P1-P4 from $\mathcal{P}$ imply new axioms about the concepts $\mathsf{Cystic\_Fibrosis}$ and $\mathsf{Genetic\_Disorder}$, namely their disjointness:

$$\mathcal{P} \models \gamma := (\mathsf{Genetic\_Disorder} \sqcap \mathsf{Cystic\_Fibrosis} \sqsubseteq \bot) \tag{5}$$

The entailment (5) can be proved using axiom E2 which is equivalent to:

$$\top \sqsubseteq \exists \mathsf{has\_Focus}.(\mathsf{Genetic\_Disorder} \sqcup \neg \mathsf{Cystic\_Fibrosis}) \tag{6}$$

The inclusion (6) and P4 imply that every element in the domain must be a project—that is, $\mathcal{P} \models (\top \ \sqsubseteq \ \mathsf{Project})$. Now, together with axiom E1, this implies (5).

The axioms E1 and E2 not only imply new statements about the medical terms, but also cause inconsistencies when used together with the imported axioms from $\mathcal{Q}$. Indeed, from (1) and (5) we obtain $\mathcal{P} \cup \mathcal{Q} \models \delta := (\mathsf{Cystic\_Fibrosis} \sqsubseteq \bot)$, which expresses the inconsistency of the concept $\mathsf{Cystic\_Fibrosis}$.

To summarize, we have seen that importing an external ontology can lead to undesirable side effects in our knowledge reuse scenario, like the entailment of new axioms or even inconsistencies involving the reused vocabulary. In the next section we discuss how to formalize the effects we consider undesirable.

3.2 Conservative Extensions and Safety for an Ontology

As argued in the previous section, an important requirement for the reuse of an ontology $\mathcal{Q}$ within an ontology $\mathcal{P}$ should be that $\mathcal{P} \cup \mathcal{Q}$ produces exactly the same logical consequences over the vocabulary of $\mathcal{Q}$ as $\mathcal{Q}$ alone does. This requirement can be naturally formulated using the well-known notion of a conservative extension, which has recently been investigated in the context of ontologies (Ghilardi, Lutz, & Wolter, 2006; Lutz et al., 2007).

Definition 1 (Deductive Conservative Extension). Let $\mathcal{O}$ and $\mathcal{O}_1 \subseteq \mathcal{O}$ be two $\mathcal{L}$-ontologies, and $\mathbf{S}$ a signature over $\mathcal{L}$. We say that $\mathcal{O}$ is a *deductive* $\mathbf{S}$*-conservative extension* of $\mathcal{O}_1$ w.r.t. $\mathcal{L}$, if for every axiom α over $\mathcal{L}$ with $\mathsf{Sig}(\alpha) \subseteq \mathbf{S}$, we have $\mathcal{O} \models \alpha$ iff $\mathcal{O}_1 \models \alpha$. We say that $\mathcal{O}$ is a *deductive conservative extension* of $\mathcal{O}_1$ w.r.t. $\mathcal{L}$ if $\mathcal{O}$ is a deductive $\mathbf{S}$-conservative extension of $\mathcal{O}_1$ w.r.t. $\mathcal{L}$ for $\mathbf{S} = \mathsf{Sig}(\mathcal{O}_1)$. ◊

In other words, an ontology $\mathcal{O}$ is a deductive $\mathbf{S}$-conservative extension of $\mathcal{O}_1$ for a signature $\mathbf{S}$ and language $\mathcal{L}$ if and only if every logical consequence α of $\mathcal{O}$ constructed using the language $\mathcal{L}$ and symbols only from $\mathbf{S}$, is already a logical consequence of $\mathcal{O}_1$; that is, the additional axioms in $\mathcal{O} \setminus \mathcal{O}_1$ do not result into new logical consequences over the vocabulary $\mathbf{S}$. Note that if $\mathcal{O}$ is a deductive $\mathbf{S}$-conservative extension of $\mathcal{O}_1$ w.r.t. $\mathcal{L}$, then $\mathcal{O}$ is a deductive $\mathbf{S}_1$-conservative extension of $\mathcal{O}_1$ w.r.t. $\mathcal{L}$ for every $\mathbf{S}_1 \subseteq \mathbf{S}$.

The notion of a deductive conservative extension can be directly applied to our ontology reuse scenario.

Definition 2 (Safety for an Ontology). Given $\mathcal{L}$-ontologies $\mathcal{O}$ and $\mathcal{O}'$, we say that $\mathcal{O}$ is *safe* for $\mathcal{O}'$ (or $\mathcal{O}$ imports $\mathcal{O}'$ *in a safe way*) w.r.t. $\mathcal{L}$ if $\mathcal{O} \cup \mathcal{O}'$ is a deductive conservative extension of $\mathcal{O}'$ w.r.t. $\mathcal{L}$. $\Diamond$

Hence, the first reasoning task relevant for our ontology reuse scenario can be formulated as follows:

T1. given $\mathcal{L}$-ontologies $\mathcal{O}$ and $\mathcal{O}'$, determine if $\mathcal{O}$ is safe for $\mathcal{O}'$ w.r.t. $\mathcal{L}$.

We have shown in Section 3.1 that, given $\mathcal{P}$ consisting of axioms P1–P4, E1, E2, and $\mathcal{Q}$ consisting of axioms M1–M5 from Figure 1, there exists an axiom $\delta = (\mathsf{Cystic\_Fibrosis} \sqsubseteq \bot)$ that uses only symbols in $\mathsf{Sig}(\mathcal{Q})$ such that $\mathcal{Q} \not\models \delta$ but $\mathcal{P} \cup \mathcal{Q} \models \delta$. According to Definition 1, this means that $\mathcal{P} \cup \mathcal{Q}$ is not a deductive conservative extension of $\mathcal{Q}$ w.r.t. any language $\mathcal{L}$ in which δ can be expressed (e.g. $\mathcal{L} = \mathcal{ALC}$). It is possible, however, to show that if the axiom E2 is removed from $\mathcal{P}$ then for the resulting ontology $\mathcal{P}_1 = \mathcal{P} \setminus \{\text{E2}\}$, $\mathcal{P}_1 \cup \mathcal{Q}$ is a deductive conservative extension of $\mathcal{Q}$. The following notion is useful for proving deductive conservative extensions:

Definition 3 (Model Conservative Extension, Lutz et al., 2007).
Let $\mathcal{O}$ and $\mathcal{O}_1 \subseteq \mathcal{O}$ be two $\mathcal{L}$-ontologies and $\mathbf{S}$ a signature over $\mathcal{L}$. We say that $\mathcal{O}$ is a *model* $\mathbf{S}$-*conservative extension* of $\mathcal{O}_1$, if for every model $\mathcal{I}$ of $\mathcal{O}_1$, there exists a model $\mathcal{J}$ of $\mathcal{O}$ such that $\mathcal{I}|_{\mathbf{S}} = \mathcal{J}|_{\mathbf{S}}$. We say that $\mathcal{O}$ is a *model conservative extension* of $\mathcal{O}_1$ if $\mathcal{O}$ is a model $\mathbf{S}$-conservative extension of $\mathcal{O}_1$ for $\mathbf{S} = \mathsf{Sig}(\mathcal{O}_1)$. $\Diamond$

The notion of model conservative extension in Definition 3 can be seen as a semantic counterpart for the notion of deductive conservative extension in Definition 1: the latter is defined in terms of logical entailment, whereas the former is defined in terms of models. Intuitively, an ontology $\mathcal{O}$ is a model $\mathbf{S}$-conservative extension of $\mathcal{O}_1$ if for every model of $\mathcal{O}_1$ one can find a model of $\mathcal{O}$ over the same domain which interprets the symbols from $\mathbf{S}$ in the same way. The notion of model conservative extension, however, does not provide a complete characterization of deductive conservative extensions, as given in Definition 1; that is, this notion can be used for proving that an ontology is a deductive conservative extension of another, but not vice versa:

Proposition 4 (Model vs. Deductive Conservative Extensions, Lutz et al., 2007)

1. *For every two $\mathcal{L}$-ontologies $\mathcal{O}$, $\mathcal{O}_1 \subseteq \mathcal{O}$, and a signature $\mathbf{S}$ over $\mathcal{L}$, if $\mathcal{O}$ is a model $\mathbf{S}$-conservative extension of $\mathcal{O}_1$ then $\mathcal{O}$ is a deductive $\mathbf{S}$-conservative extension of $\mathcal{O}_1$ w.r.t. $\mathcal{L}$;*

2. *There exist two $\mathcal{ALC}$ ontologies $\mathcal{O}$ and $\mathcal{O}_1 \subseteq \mathcal{O}$ such that $\mathcal{O}$ is a deductive conservative extension of $\mathcal{O}_1$ w.r.t. $\mathcal{ALC}$, but $\mathcal{O}$ is not a model conservative extension of $\mathcal{O}_1$.*

Example 5 Consider an ontology $\mathcal{P}_1$ consisting of axioms P1–P4, and E1 and an ontology $\mathcal{Q}$ consisting of axioms M1–M5 from Figure 1. We demonstrate that $\mathcal{P}_1 \cup \mathcal{Q}$ is a deductive conservative extension of $\mathcal{Q}$. According to proposition 4, it is sufficient to show that $\mathcal{P}_1 \cup \mathcal{Q}$ is a model conservative extension of $\mathcal{Q}$; that is, for every model $\mathcal{I}$ of $\mathcal{Q}$ there exists a model $\mathcal{J}$ of $\mathcal{P}_1 \cup \mathcal{Q}$ such that $\mathcal{I}|_{\mathbf{S}} = \mathcal{J}|_{\mathbf{S}}$ for $\mathbf{S} = \mathsf{Sig}(\mathcal{Q})$.

The required model $\mathcal{J}$ can be constructed from $\mathcal{I}$ as follows: take $\mathcal{J}$ to be identical to $\mathcal{I}$ except for the interpretations of the atomic concepts $\mathsf{Genetic\_Disorder\_Project}$, $\mathsf{Cystic\_Fibrosis\_EUProject}$, $\mathsf{Project}$, $\mathsf{EUProject}$ and the atomic role $\mathsf{has\_Focus}$, all of which we interpret in $\mathcal{J}$ as the empty set. It is easy to check that all the axioms P1–P4, E1 are satisfied in $\mathcal{J}$and hence $\mathcal{J} \models \mathcal{P}_1$. Moreover, since the interpretation of the symbols in $\mathcal{Q}$ remains unchanged, we have $\mathcal{I}|_{\mathsf{Sig}(\mathcal{Q})} = \mathcal{J}|_{\mathsf{Sig}(\mathcal{Q})}$ and $\mathcal{J} \models \mathcal{Q}$. Hence, $\mathcal{P}_1 \cup \mathcal{Q}$ is a model conservative extension of $\mathcal{Q}$.

For an example of Statement 2 of the Proposition, we refer the interested reader to the literature (Lutz et al., 2007, p. 455). ♢

3.3 Safety of an Ontology for a Signature

So far, in our ontology reuse scenario we have assumed that the reused ontology $\mathcal{Q}$ is fixed and the axioms of $\mathcal{Q}$ are just copied into $\mathcal{P}$ during the import. In practice, however, it is often convenient to keep $\mathcal{Q}$ separate from $\mathcal{P}$ and make its axioms available on demand via a reference. This makes it possible to continue developing $\mathcal{P}$ and $\mathcal{Q}$ independently. For example, the ontology engineers of the project ontology $\mathcal{P}$ may not be willing to depend on a particular version of $\mathcal{Q}$, or may even decide at a later time to reuse the medical terms ($\mathsf{Cystic\_Fibrosis}$ and $\mathsf{Genetic\_Disorder}$) from another medical ontology instead. Therefore, for many application scenarios it is important to develop a stronger safety condition for $\mathcal{P}$ which depends as little as possible on the particular ontology $\mathcal{Q}$ to be reused. In order to formulate such condition, we abstract from the particular ontology $\mathcal{Q}$ to be imported and focus instead on the symbols from $\mathcal{Q}$ that are to be reused:

Definition 6 (Safety for a Signature). Let $\mathcal{O}$ be a $\mathcal{L}$-ontology and $\mathbf{S}$ a signature over $\mathcal{L}$. We say that $\mathcal{O}$ *is safe for* $\mathbf{S}$ *w.r.t.* $\mathcal{L}$, if for every $\mathcal{L}$-ontology $\mathcal{O}'$ with $\mathsf{Sig}(\mathcal{O}) \cap \mathsf{Sig}(\mathcal{O}') \subseteq \mathbf{S}$, we have that $\mathcal{O}$ is safe for $\mathcal{O}'$ w.r.t. $\mathcal{L}$; that is, $\mathcal{O} \cup \mathcal{O}'$ is a deductive conservative extension of $\mathcal{O}'$ w.r.t. $\mathcal{L}$. ♢

Intuitively, in our knowledge reuse scenario, an ontology $\mathcal{O}$ is safe for a signature $\mathbf{S}$ w.r.t. a language $\mathcal{L}$ if and only if it imports in a safe way any ontology $\mathcal{O}'$ written in $\mathcal{L}$ that shares only symbols from $\mathbf{S}$ with $\mathcal{O}$. The associated reasoning problem is then formulated in the following way:

T2. given an $\mathcal{L}$-ontology $\mathcal{O}$ and a signature $\mathbf{S}$ over $\mathcal{L}$,
determine if $\mathcal{O}$ is safe for $\mathbf{S}$ w.r.t. $\mathcal{L}$.

As seen in Section 3.2, the ontology $\mathcal{P}$ consisting of axioms P1–P4, E1, E2 from Figure 1, does not import $\mathcal{Q}$ consisting of axioms M1–M5 from Figure 1 in a safe way for $\mathcal{L} = \mathcal{ALC}$. According to Definition 6, since $\mathsf{Sig}(\mathcal{P}) \cap \mathsf{Sig}(\mathcal{Q}) \subseteq \mathbf{S} = \{\mathsf{Cystic\_Fibrosis}, \mathsf{Genetic\_Disorder}\}$, the ontology $\mathcal{P}$ is not safe for $\mathbf{S}$ w.r.t. $\mathcal{L}$.

In fact, it is possible to show a stronger result, namely, that any ontology $\mathcal{O}$ containing axiom E2 is not safe for $\mathbf{S} = \{\mathsf{Cystic\_Fibrosis}, \mathsf{Genetic\_Disorder}\}$ w.r.t. $\mathcal{L} = \mathcal{ALC}$. Consider an ontology $\mathcal{O}' = \{\beta_1, \beta_2\}$, where $\beta_1 = (\top \sqsubseteq \mathsf{Cystic\_Fibrosis})$ and $\beta_2 = (\mathsf{Genetic\_Disorder} \sqsubseteq \bot)$. Since E2 is equivalent to axiom (6), it is easy to see that $\mathcal{O} \cup \mathcal{O}'$ is inconsistent. Indeed E2, β_1 and β_2 imply the contradiction $\alpha = (\top \sqsubseteq \bot)$, which is not entailed by $\mathcal{O}'$. Hence, $\mathcal{O} \cup \mathcal{O}'$ is not a deductive conservative extension of $\mathcal{O}'$. By Definition 6, since $\mathsf{Sig}(\mathcal{O}) \cap \mathsf{Sig}(\mathcal{O}') \subseteq \mathbf{S}$, this means that $\mathcal{O}$ is not safe for $\mathbf{S}$.

It is clear how one could prove that an ontology $\mathcal{O}$ is not safe for a signature $\mathbf{S}$: simply find an ontology $\mathcal{O}'$ with $\mathsf{Sig}(\mathcal{O}) \cap \mathsf{Sig}(\mathcal{O}') \subseteq \mathbf{S}$, such that $\mathcal{O} \cup \mathcal{O}'$ is not a deductive conservative extension of $\mathcal{O}'$. It is not so clear, however, how one could prove that $\mathcal{O}$ is safe for $\mathbf{S}$. It turns out that the notion of model conservative extensions can also be used for this purpose. The following lemma introduces a property which relates the notion of model conservative extension with the notion of safety for a signature. Intuitively, it says that the notion of model conservative extension is stable under expansion with new axioms provided they share only symbols from $\mathbf{S}$ with the original ontologies.

Lemma 7 *Let $\mathcal{O}$, $\mathcal{O}_1 \subseteq \mathcal{O}$, and $\mathcal{O}'$ be $\mathcal{L}$-ontologies and $\mathbf{S}$ a signature over $\mathcal{L}$ such that $\mathcal{O}$ is a model $\mathbf{S}$-conservative extension of $\mathcal{O}_1$ and $\mathsf{Sig}(\mathcal{O}) \cap \mathsf{Sig}(\mathcal{O}') \subseteq \mathbf{S}$. Then $\mathcal{O} \cup \mathcal{O}'$ is a model $\mathbf{S}'$-conservative extension of $\mathcal{O}_1 \cup \mathcal{O}'$ for $\mathbf{S}' = \mathbf{S} \cup \mathsf{Sig}(\mathcal{O}')$.*

Proof. In order to show that $\mathcal{O}\cup\mathcal{O}'$ is a model $\mathbf{S}'$-conservative extension of $\mathcal{O}_1\cup\mathcal{O}'$ according to Definition 3, let $\mathcal{I}$ be a model of $\mathcal{O}_1 \cup \mathcal{O}'$. We just construct a model $\mathcal{J}$ of $\mathcal{O} \cup \mathcal{O}'$ such that $\mathcal{I}|_{\mathbf{S}'} = \mathcal{J}|_{\mathbf{S}'}$. (♯)

Since $\mathcal{O}$ is a model $\mathbf{S}$-conservative extension of $\mathcal{O}_1$, and $\mathcal{I}$ is a model of $\mathcal{O}_1$, by Definition 3 there exists a model $\mathcal{J}_1$ of $\mathcal{O}$ such that $\mathcal{I}|_{\mathbf{S}} = \mathcal{J}_1|_{\mathbf{S}}$. Let $\mathcal{J}$ be an interpretation such that $\mathcal{J}|_{\mathsf{Sig}(\mathcal{O})} = \mathcal{J}_1|_{\mathsf{Sig}(\mathcal{O})}$ and $\mathcal{J}|_{\mathbf{S}'} = \mathcal{I}|_{\mathbf{S}'}$. Since $\mathsf{Sig}(\mathcal{O}) \cap \mathbf{S}' = \mathsf{Sig}(\mathcal{O}) \cap (\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}')) \subseteq \mathbf{S} \cup (\mathsf{Sig}(\mathcal{O}) \cap \mathsf{Sig}(\mathcal{O}')) \subseteq \mathbf{S}$ and $\mathcal{I}|_{\mathbf{S}} = \mathcal{J}_1|_{\mathbf{S}}$, such interpretation $\mathcal{J}$ always exists. Since $\mathcal{J}|_{\mathsf{Sig}(\mathcal{O})} = \mathcal{J}_1|_{\mathsf{Sig}(\mathcal{O})}$ and $\mathcal{J}_1 \models \mathcal{O}$, we have $\mathcal{J} \models \mathcal{O}$; since $\mathcal{J}|_{\mathbf{S}'} = \mathcal{I}|_{\mathbf{S}'}$, $\mathsf{Sig}(\mathcal{O}') \subseteq \mathbf{S}'$, and $\mathcal{I} \models \mathcal{O}'$, we have $\mathcal{J} \models \mathcal{O}'$. Hence $\mathcal{J} \models \mathcal{O} \cup \mathcal{O}'$ and $\mathcal{I}|_{\mathbf{S}'} = \mathcal{J}|_{\mathbf{S}'}$, which proves (♯). □

Lemma 7 allows us to identify a condition sufficient to ensure the safety of an ontology for a signature:

Proposition 8 (Safety for a Signature vs. Model Conservative Extensions) *Let $\mathcal{O}$ be an $\mathcal{L}$-ontology and $\mathbf{S}$ a signature over $\mathcal{L}$ such that $\mathcal{O}$ is a model $\mathbf{S}$-conservative extension of the empty ontology $\mathcal{O}_1 = \emptyset$; that is, for every interpretation $\mathcal{I}$ there exists a model $\mathcal{J}$ of $\mathcal{O}$ such that $\mathcal{J}|_{\mathbf{S}} = \mathcal{I}|_{\mathbf{S}}$. Then $\mathcal{O}$ is safe for $\mathbf{S}$ w.r.t. $\mathcal{L}$.*

Proof. In order to prove that $\mathcal{O}$ is safe for $\mathbf{S}$ w.r.t. $\mathcal{L}$ according to Definition 6, take any $\mathcal{SHOIQ}$ ontology $\mathcal{O}'$ such that $\mathsf{Sig}(\mathcal{O}) \cap \mathsf{Sig}(\mathcal{O}') \subseteq \mathbf{S}$. We need to demonstrate that $\mathcal{O} \cup \mathcal{O}'$ is a deductive conservative extension of $\mathcal{O}'$ w.r.t. $\mathcal{L}$. (♯)

Indeed, by Lemma 7, since $\mathcal{O}$ is a model $\mathbf{S}$-conservative extension of $\mathcal{O}_1 = \emptyset$, and $\mathsf{Sig}(\mathcal{O})\cap\mathsf{Sig}(\mathcal{O}') \subseteq \mathbf{S}$, we have that $\mathcal{O}\cup\mathcal{O}'$ is a model $\mathbf{S}'$-conservative extension of $\mathcal{O}_1\cup\mathcal{O}' = \mathcal{O}'$ for $\mathbf{S}' = \mathbf{S} \cup \mathsf{Sig}(\mathcal{O}')$. In particular, since $\mathsf{Sig}(\mathcal{O}') \subseteq \mathbf{S}'$, we have that $\mathcal{O} \cup \mathcal{O}'$ is a deductive conservative extension of $\mathcal{O}'$, as was required to prove (♯). □

Example 9 Let $\mathcal{P}_1$ be the ontology consisting of axioms P1–P4, and E1 from Figure 1. We show that $\mathcal{P}_1$ is safe for $\mathbf{S} = \{\mathsf{Cystic\_Fibrosis}, \mathsf{Genetic\_Disorder}\}$ w.r.t. $\mathcal{L} = \mathcal{SHOIQ}$. By Proposition 8, in order to prove safety for $\mathcal{P}_1$ it is sufficient to demonstrate that $\mathcal{P}_1$ is a model $\mathbf{S}$-conservative extension of the empty ontology, that is, for every $\mathbf{S}$-interpretation $\mathcal{I}$ there exists a model $\mathcal{J}$ of $\mathcal{P}_1$ such that $\mathcal{I}|_{\mathbf{S}} = \mathcal{J}|_{\mathbf{S}}$.

Consider the model $\mathcal{J}$ obtained from $\mathcal{I}$ as in Example 5. As shown in Example 5, $\mathcal{J}$ is a model of $\mathcal{P}_1$ and $\mathcal{I}|_{\mathsf{Sig}(\mathcal{Q})} = \mathcal{J}|_{\mathsf{Sig}(\mathcal{Q})}$ where $\mathcal{Q}$ consists of axioms M1–M5 from Figure 1. In particular, since $\mathbf{S} \subseteq \mathsf{Sig}(\mathcal{Q})$, we have $\mathcal{I}|_{\mathbf{S}} = \mathcal{J}|_{\mathbf{S}}$. ♢

3.4 Extraction of Modules from Ontologies

In our example from Figure 1 the medical ontology $\mathcal{Q}$ is very small. Well established medical ontologies, however, can be very large and may describe subject matters in which the designer of $\mathcal{P}$ is not interested. For example, the medical ontology $\mathcal{Q}$ could contain information about genes, anatomy, surgical techniques, etc.

Even if $\mathcal{P}$ imports $\mathcal{Q}$ without changing the meaning of the reused symbols, processing—that is, browsing, reasoning over, etc—the resulting ontology $\mathcal{P} \cup \mathcal{Q}$ may be considerably harder than processing $\mathcal{P}$ alone. Ideally, one would like to extract a (hopefully small) fragment $\mathcal{Q}_1$ of the external medical ontology—a *module*—that describes just the concepts that are reused in $\mathcal{P}$.

Intuitively, when answering an arbitrary query over the signature of $\mathcal{P}$, importing the module $\mathcal{Q}_1$ should give exactly the same answers as if the whole ontology $\mathcal{Q}$ had been imported.

Definition 10 (Module for an Ontology). Let $\mathcal{O}$, $\mathcal{O}'$ and $\mathcal{O}'_1 \subseteq \mathcal{O}'$ be $\mathcal{L}$-ontologies. We say that $\mathcal{O}'_1$ is a *module for* $\mathcal{O}$ *in* $\mathcal{O}'$ *w.r.t.* $\mathcal{L}$, if $\mathcal{O} \cup \mathcal{O}'$ is a deductive $\mathbf{S}$-conservative extension of $\mathcal{O} \cup \mathcal{O}'_1$ for $\mathbf{S} = \mathsf{Sig}(\mathcal{O})$ w.r.t. $\mathcal{L}$. ♢

The task of extracting modules from imported ontologies in our ontology reuse scenario can be thus formulated as follows:

T3. given $\mathcal{L}$-ontologies $\mathcal{O}$, $\mathcal{O}'$,
compute a module $\mathcal{O}'_1$ for $\mathcal{O}$ in $\mathcal{O}'$ w.r.t. $\mathcal{L}$.

Example 11 Consider the ontology $\mathcal{P}_1$ consisting of axioms P1–P4, E1 and the ontology $\mathcal{Q}$ consisting of axioms M1–M5 from Figure 1. Recall that the axiom α from (1) is a consequence of axioms M1, M2 and M4 as well as of axioms M1, M3 and M4 from ontology $\mathcal{Q}$. In fact, these sets of axioms are actually minimal subsets of $\mathcal{Q}$ that imply α. In particular, for the subset $\mathcal{Q}_0$ of $\mathcal{Q}$ consisting of axioms M2, M3, M4 and M5, we have $\mathcal{Q}_0 \not\models \alpha$. It can be demonstrated that $\mathcal{P}_1 \cup \mathcal{Q}_0$ is a deductive conservative extension of $\mathcal{Q}_0$. In particular $\mathcal{P}_1 \cup \mathcal{Q}_0 \not\models \alpha$. But then, according to Definition 10, $\mathcal{Q}_0$ is not a module for $\mathcal{P}_1$ in $\mathcal{Q}$ w.r.t. $\mathcal{L} = \mathcal{ALC}$ or its extensions, since $\mathcal{P}_1 \cup \mathcal{Q}$ is not an deductive $\mathbf{S}$-conservative extension of $\mathcal{P}_1 \cup \mathcal{Q}_0$ w.r.t. $\mathcal{L} = \mathcal{ALC}$ for $\mathbf{S} = \mathsf{Sig}(\mathcal{P}_1)$. Indeed, $\mathcal{P}_1 \cup \mathcal{Q} \models \alpha$, $\mathsf{Sig}(\alpha) \subseteq \mathbf{S}$ but $\mathcal{P}_1 \cup \mathcal{Q}_0 \not\models \alpha$. Similarly, one can show that any subset of $\mathcal{Q}$ that does not imply α is not a module in $\mathcal{Q}$ w.r.t. $\mathcal{P}_1$.

On the other hand, it is possible to show that both subsets $\mathcal{Q}_1 = \{\text{M1, M2, M4}\}$ and $\mathcal{Q}_2 = \{\text{M1, M3, M4}\}$ of $\mathcal{Q}$ are modules for $\mathcal{P}_1$ in $\mathcal{Q}$ w.r.t. $\mathcal{L} = \mathcal{SHOIQ}$. To do so, Definition 10, we need to demonstrate that $\mathcal{P}_1 \cup \mathcal{Q}$ is a deductive $\mathbf{S}$-conservative extension of both $\mathcal{P}_1 \cup \mathcal{Q}_1$ and $\mathcal{P}_1 \cup \mathcal{Q}_2$ for $\mathbf{S} = \mathsf{Sig}(\mathcal{P}_1)$ w.r.t. $\mathcal{L}$. As usual, we demonstrate a stronger fact, that $\mathcal{P}_1 \cup \mathcal{Q}$ is a model $\mathbf{S}$-conservative extension of $\mathcal{P}_1 \cup \mathcal{Q}_1$ and of $\mathcal{P}_1 \cup \mathcal{Q}_2$ for $\mathbf{S}$ which is sufficient by Claim 1 of Proposition 4.

In order to show that $\mathcal{P}_1 \cup \mathcal{Q}$ is a model $\mathbf{S}$-conservative extension of $\mathcal{P}_1 \cup \mathcal{Q}_1$ for $\mathbf{S} = \mathsf{Sig}(\mathcal{P}_1)$, consider any model $\mathcal{I}$ of $\mathcal{P}_1 \cup \mathcal{Q}_1$. We need to construct a model $\mathcal{J}$ of $\mathcal{P}_1 \cup \mathcal{Q}$ such that $\mathcal{I}|_{\mathbf{S}} = \mathcal{J}|_{\mathbf{S}}$. Let $\mathcal{J}$ be defined exactly as $\mathcal{I}$ except that the interpretations of the atomic concepts $\mathsf{Fibrosis}$, $\mathsf{Pancreas}$, $\mathsf{Genetic\_Fibrosis}$, and $\mathsf{Genetic\_Origin}$ are defined as the interpretation of $\mathsf{Cystic\_Fibrosis}$ in $\mathcal{I}$, and the interpretations of atomic roles $\mathsf{located\_In}$ and has_Origin are defined as the identity relation. It is easy to see that axioms M1–M3 and M5 are satisfied in $\mathcal{J}$. Since we do not modify the interpretation of the symbols in $\mathcal{P}_1$, $\mathcal{J}$ also satisfies all axioms from $\mathcal{P}_1$. Moreover, $\mathcal{J}$ is a model of M4, because Genetic_Fibrosis and Genetic_Disorder are interpreted in $\mathcal{J}$ like Cystic_Fibrosis and Genetic_Disorder in $\mathcal{I}$, and $\mathcal{I}$ is a model of $\mathcal{Q}_1$, which implies the concept inclusion $\alpha = (\mathsf{Cystic\_Fibrosis} \sqsubseteq \mathsf{Genetic\_Disorder})$. Hence we have constructed a model $\mathcal{J}$ of $\mathcal{P}_1 \cup \mathcal{Q}$ such that $\mathcal{I}|_{\mathbf{S}} = \mathcal{J}|_{\mathbf{S}}$, thus $\mathcal{P}_1 \cup \mathcal{Q}$ is a model $\mathbf{S}$-conservative extension of $\mathcal{P}_1 \cup \mathcal{Q}_1$.

In fact, the construction above works if we replace $\mathcal{Q}_1$ with any subset of $\mathcal{Q}$ that implies α. In particular, $\mathcal{P}_1 \cup \mathcal{Q}$ is also a model $\mathbf{S}$-conservative extension of $\mathcal{P}_1 \cup \mathcal{Q}_2$. In this way, we have demonstrated that the modules for $\mathcal{P}$ in $\mathcal{Q}$ are exactly those subsets of $\mathcal{Q}$ that imply α. ◊

An algorithm implementing the task T3 can be used for extracting a module from an ontology $\mathcal{Q}$ imported into $\mathcal{P}$ prior to performing reasoning over the terms in $\mathcal{P}$. However, when the ontology $\mathcal{P}$ is modified, the module has to be extracted again since a module $\mathcal{Q}_1$ for $\mathcal{P}$ in $\mathcal{Q}$ might not necessarily be a module for the modified ontology. Since the extraction of modules is potentially an expensive operation, it would be more convenient to extract a module only once and reuse it for any version of the ontology $\mathcal{P}$ that reuses the specified symbols from $\mathcal{Q}$. This idea motivates the following definition:

Definition 12 (Module for a Signature). Let $\mathcal{O}'$ and $\mathcal{O}'_1 \subseteq \mathcal{O}'$ be $\mathcal{L}$-ontologies and $\mathbf{S}$ a signature over $\mathcal{L}$. We say that $\mathcal{O}'_1$ *is a module for* $\mathbf{S}$ *in* $\mathcal{O}'$ (or an $\mathbf{S}$-*module in* $\mathcal{O}'$) w.r.t. $\mathcal{L}$, if for every $\mathcal{L}$-ontology $\mathcal{O}$ with $\mathsf{Sig}(\mathcal{O}) \cap \mathsf{Sig}(\mathcal{O}') \subseteq \mathbf{S}$, we have that $\mathcal{O}'_1$ is a module for $\mathcal{O}$ in $\mathcal{O}'$ w.r.t. $\mathcal{L}$. ◊

Intuitively, the notion of a module for a signature is a uniform analog of the notion of a module for an ontology, in a similar way as the notion of safety for a signature is a uniform analog of safety for an ontology. The reasoning task corresponding to Definition 12 can be formulated as follows:

T4. given a $\mathcal{L}$-ontology $\mathcal{O}'$ and a signature $\mathbf{S}$ over $\mathcal{L}$,
compute a module $\mathcal{O}'_1$ for $\mathbf{S}$ in $\mathcal{O}'$.

Continuing Example 11, it is possible to demonstrate that any subset of $\mathcal{Q}$ that implies axiom α, is in fact a module for $\mathbf{S} = \{\mathsf{Cystic\_Fibrosis}, \mathsf{Genetic\_Disorder}\}$ in $\mathcal{Q}$, that is, it can be imported instead of $\mathcal{Q}$ into every ontology $\mathcal{O}$ that shares with $\mathcal{Q}$ only symbols from $\mathbf{S}$. In order to prove this, we use the following sufficient condition based on the notion of model conservative extension:

Proposition 13 (Modules for a Signature vs. Model Conservative Extensions) *Let $\mathcal{O}'$ and $\mathcal{O}'_1 \subseteq \mathcal{O}'$ be $\mathcal{L}$-ontologies and $\mathbf{S}$ a signature over $\mathcal{L}$ such that $\mathcal{O}'$ is a model $\mathbf{S}$-conservative extension of $\mathcal{O}'_1$. Then $\mathcal{O}'_1$ is a module for $\mathbf{S}$ in $\mathcal{O}'$ w.r.t. $\mathcal{L}$.*

Proof. In order to prove that $\mathcal{O}'_1$ is a module for $\mathbf{S}$ in $\mathcal{O}'$ w.r.t. $\mathcal{L}$ according to Definition 12, take any $\mathcal{SHOIQ}$ ontology $\mathcal{O}$ such that $\mathsf{Sig}(\mathcal{O}) \cap \mathsf{Sig}(\mathcal{O}') \subseteq \mathbf{S}$. We need to demonstrate that $\mathcal{O}'_1$ is a module for $\mathcal{O}$ in $\mathcal{O}'$ w.r.t. $\mathcal{L}$, that is, according to Definition 10, $\mathcal{O} \cup \mathcal{O}'$ is a deductive $\mathbf{S}'$-conservative extension of $\mathcal{O} \cup \mathcal{O}'_1$ w.r.t. $\mathcal{L}$ for $\mathbf{S}' = \mathsf{Sig}(\mathcal{O})$. (♯)

Indeed, by Lemma 7, since $\mathcal{O}'$ is a model $\mathbf{S}$-conservative extension of $\mathcal{O}'_1$, and $\mathsf{Sig}(\mathcal{O}') \cap \mathsf{Sig}(\mathcal{O}) \subseteq \mathbf{S}$, we have that $\mathcal{O}' \cup \mathcal{O}$ is a model $\mathbf{S}''$-conservative extension of $\mathcal{O}'_1 \cup \mathcal{O}$ for $\mathbf{S}'' = \mathbf{S} \cup \mathsf{Sig}(\mathcal{O})$. In particular, since $\mathbf{S}' = \mathsf{Sig}(\mathcal{O}) \subseteq \mathbf{S}''$, we have that $\mathcal{O}' \cup \mathcal{O}$ is a deductive $\mathbf{S}'$-conservative extension of $\mathcal{O}'_1 \cup \mathcal{O}$ w.r.t. $\mathcal{L}$, as was required to prove ($\sharp$). $\square$

Example 14 Let $\mathcal{Q}$ and α be as in Example 11. We demonstrate that any subset $\mathcal{Q}_1$ of $\mathcal{Q}$ which implies α is a module for $\mathbf{S} = \{\mathsf{Cystic\_Fibrosis}, \mathsf{Genetic\_Disorder}\}$ in $\mathcal{Q}$. According to Proposition 13, it is sufficient to demonstrate that $\mathcal{Q}$ is a model $\mathbf{S}$-conservative extension of $\mathcal{Q}_1$, that is, for every model $\mathcal{I}$ of $\mathcal{Q}_1$ there exists a model $\mathcal{J}$ of $\mathcal{Q}$ such that $\mathcal{I}|_{\mathbf{S}} = \mathcal{J}|_{\mathbf{S}}$. It is easy to see that if the model $\mathcal{J}$ is constructed from $\mathcal{I}$ as in Example 11, then the required property holds. $\diamondsuit$

Note that a module for a signature $\mathbf{S}$ in $\mathcal{Q}$ does not necessarily contain all the axioms that contain symbols from $\mathbf{S}$. For example, the module $\mathcal{Q}_1$ consisting of the axiom M1, M2 and M4 from $\mathcal{Q}$ does not contain axiom M5 which mentions the atomic concept $\mathsf{Cystic\_Fibrosis}$ from $\mathbf{S}$. Also note that even in a minimal module like $\mathcal{Q}_1$ there might still be some axioms like M2 that do not mention symbols from $\mathbf{S}$ at all.

3.5 Minimal Modules and Essential Axioms

One is usually not interested in extracting arbitrary modules from a reused ontology, but in extracting modules that are easy to process afterwards. Ideally, the extracted modules should be as small as possible. Hence it is reasonable to consider the problem of extracting *minimal modules*; that is, modules that contain no other module as a subset.

Examples 11 and 14 demonstrate that a minimal module for an ontology or a signature is not necessarily unique: the ontology $\mathcal{Q}$ consisting of axioms M1–M5 has two minimal modules $\mathcal{Q}_1 = \{\mathrm{M1}, \mathrm{M2}, \mathrm{M4}\}$, and $\mathcal{Q}_2 = \{\mathrm{M1}, \mathrm{M3}, \mathrm{M4}\}$, for the ontology $\mathcal{P}_1 = \{\mathrm{P1}, \mathrm{P2}, \mathrm{P3}, \mathrm{E1}\}$ as well as for the signature $\mathbf{S} = \{\mathsf{Cystic\_Fibrosis}, \mathsf{Genetic\_Disorder}\}$, since these are the minimal sets of axioms that imply axiom $\alpha = (\mathsf{Cystic\_Fibrosis} \sqsubseteq \mathsf{Genetic\_Disorder})$. Depending on the application scenario, one can consider several variations of tasks T3 and T4 for computing minimal modules. In some applications it might be necessary to extract all minimal modules, whereas in others any minimal module suffices.

Axioms that do not occur in a minimal module in $\mathcal{Q}$ are not essential for $\mathcal{P}$ because they can always be removed from every module of $\mathcal{Q}$, thus they never need not be imported into $\mathcal{P}$. This is not true for the axioms that occur in minimal modules of $\mathcal{Q}$. It might be necessary to import such axioms into $\mathcal{P}$ in order not to lose essential information from $\mathcal{Q}$. These arguments motivate the following notion:

Definition 15 (Essential Axiom). Let $\mathcal{O}$ and $\mathcal{O}'$ be $\mathcal{L}$-ontologies, $\mathbf{S}$ a signature and α an axiom over $\mathcal{L}$. We say that α is *essential for* $\mathcal{O}$ *in* $\mathcal{O}'$ *w.r.t.* $\mathcal{L}$ if α is contained in any minimal module in $\mathcal{O}'$ for $\mathcal{O}$ w.r.t. $\mathcal{L}$. We say that α is an *essential axiom for* $\mathbf{S}$ *in* $\mathcal{O}'$ *w.r.t.* $\mathcal{L}$ (or $\mathbf{S}$-*essential in* $\mathcal{O}'$) if α is contained in some minimal module for $\mathbf{S}$ in $\mathcal{O}'$ w.r.t. $\mathcal{L}$. $\diamondsuit$

In our example above the axioms M1–M4 from $\mathcal{Q}$ are essential for the ontology $\mathcal{P}_1$ and for the signature $\mathbf{S} = \{\mathsf{Cystic\_Fibrosis}, \mathsf{Genetic\_Disorder}\}$, but the axiom M5 is not essential.

In certain situations one might be interested in computing the set of essential axioms of an ontology, which can be done by computing the union of all minimal modules. Note that

Notation	Input	Task
	Checking Safety:	
T1	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Check if $\mathcal{O}$ is safe for $\mathcal{O}'$ w.r.t. $\mathcal{L}$
T2	$\mathcal{O}$, $\mathbf{S}$, $\mathcal{L}$	Check if $\mathcal{O}$ is safe for $\mathbf{S}$ w.r.t. $\mathcal{L}$
	Extracting of [all / some / union of] [minimal] module(s):	
T3[a,s,u][m]	$\mathcal{O}$, $\mathcal{O}'$, $\mathcal{L}$	Extract modules in $\mathcal{O}'$ w.r.t. $\mathcal{O}$ and $\mathcal{L}$
T4[a,s,u][m]	$\mathcal{O}'$, $\mathbf{S}$, $\mathcal{L}$	Extract modules for $\mathbf{S}$ in $\mathcal{O}'$ w.r.t. $\mathcal{L}$
	where $\mathcal{O}$, $\mathcal{O}'$ are ontologies and $\mathbf{S}$ a signature over $\mathcal{L}$	

Table 2: Summary of reasoning tasks relevant for ontology integration and reuse

computing the union of minimal modules might be easier than computing all the minimal modules since one does not need to identify which axiom belongs to which minimal module.

In Table 2 we have summarized the reasoning tasks we found to be potentially relevant for ontology reuse scenarios and have included the variants T3am, T3sm, and T3um of the task T3 and T4am, T4sm, and T4um of the task T4 for computation of minimal modules discussed in this section.

Further variants of the tasks T3 and T4 could be considered relevant to ontology reuse. For example, instead of computing minimal modules, one might be interested in computing modules with the smallest number of axioms, or modules of the smallest size measured in the number of symbols, or in any other complexity measure of the ontology. The theoretical results that we will present in this paper can easily be extended to many such reasoning tasks.

3.6 Safety and Modules for General Ontology Languages

All the notions introduced in Section 3 are defined with respect to an "ontology language". So far, however, we have implicitly assumed that the ontology languages are the description logics defined in Section 2—that is, the fragments of the DL $\mathcal{SHOIQ}$. The notions considered in Section 3 can be applied, however, to a much broader class of ontology languages. Our definitions apply to any ontology language with a notion of entailment of axioms from ontologies, and a mechanism for identifying their signatures.

Definition 16. An *ontology language* is a tuple $\mathcal{L} = (\mathbf{Sg}, \mathbf{Ax}, \mathsf{Sig}, \models)$, where $\mathbf{Sg}$ is a set of *signature elements (or vocabulary)* of $\mathcal{L}$, $\mathbf{Ax}$ is a set of *axioms* in $\mathcal{L}$, Sig is a function that assigns to every axiom $\alpha \in \mathbf{Ax}$ a finite set $\mathsf{Sig}(\alpha) \subseteq \mathbf{Sg}$ called the *signature of* α, and $\models$ is the *entailment relation* between sets of axioms $\mathcal{O} \subseteq \mathbf{Ax}$ and axioms $\alpha \in \mathbf{Ax}$, written $\mathcal{O} \models \alpha$. An *ontology over* $\mathcal{L}$ is a finite set of axioms $\mathcal{O} \subseteq \mathbf{Ax}$. We extend the function Sig to ontologies $\mathcal{O}$ as follows: $\mathsf{Sig}(\mathcal{O}) := \bigcup_{\alpha \in \mathcal{O}} \mathsf{Sig}(\alpha)$. ◊

Definition 16 provides a very general notion of an ontology language. A language $\mathcal{L}$ is given by a set of symbols (a signature), a set of formulae (axioms) that can be constructed over those symbols, a function that assigns to each formula its signature, and an entailment relation between sets of axioms. The ontology language OWL DL as well as all description logics defined in Section 2 are examples of ontology languages in accordance to Definition 16. Other examples of ontology languages are First Order Logic, Second Order Logic, and Logic Programs.

It is easy to see that the notions of deductive conservative extension (Definition 1), safety (Definitions 2 and 6) and modules (Definitions 10 and 12), as well as all the reasoning tasks from Table 2, are well-defined for every ontology language $\mathcal{L}$ given by Definition 16. The definition of model conservative extension (Definition 3) and the propositions involving model conservative extensions (Propositions 4, 8, and 13) can be also extended to other languages with a standard Tarski model-theoretic semantics, such as Higher Order Logic. To simplify the presentation, however, we will not formulate general requirements on the semantics of ontology languages, and assume that we deal with sublanguages of $\mathcal{SHOIQ}$ whenever semantics is taken into account.

In the remainder of this section, we establish the relationships between the different notions of safety and modules for arbitrary ontology languages.

Proposition 17 (Safety vs. Modules for an Ontology) *Let $\mathcal{L}$ be an ontology language, and let $\mathcal{O}$, $\mathcal{O}'$, and $\mathcal{O}'_1 \subseteq \mathcal{O}'$ be ontologies over $\mathcal{L}$. Then:*

1. *$\mathcal{O}'$ is safe for $\mathcal{O}$ w.r.t. $\mathcal{L}$ iff the empty ontology is a module for $\mathcal{O}$ in $\mathcal{O}'$ w.r.t. $\mathcal{L}$.*

2. *If $\mathcal{O}' \setminus \mathcal{O}'_1$ is safe for $\mathcal{O} \cup \mathcal{O}'_1$ then $\mathcal{O}'_1$ is a module for $\mathcal{O}$ in $\mathcal{O}'$ w.r.t. $\mathcal{L}$.*

Proof. 1. By Definition 2, $\mathcal{O}'$ is safe for $\mathcal{O}$ w.r.t. $\mathcal{L}$ iff (a) $\mathcal{O} \cup \mathcal{O}'$ is a deductive conservative extension of $\mathcal{O}$ w.r.t. $\mathcal{L}$. By Definition 10, the empty ontology $\mathcal{O}'_0 = \emptyset$ is a module for $\mathcal{O}$ in $\mathcal{O}'$ w.r.t. $\mathcal{L}$ iff (b) $\mathcal{O} \cup \mathcal{O}'$ is a deductive $\mathbf{S}$-conservative extension of $\mathcal{O} \cup \mathcal{O}'_0 = \mathcal{O}$ w.r.t. $\mathcal{L}$ for $\mathbf{S} = \mathsf{Sig}(\mathcal{O})$. It is easy to see that (a) is the same as (b).

2. By Definition 2, $\mathcal{O}' \setminus \mathcal{O}'_1$ is safe for $\mathcal{O} \cup \mathcal{O}'_1$ w.r.t. $\mathcal{L}$ iff (c) $\mathcal{O} \cup \mathcal{O}'_1 \cup (\mathcal{O}' \setminus \mathcal{O}'_1) = \mathcal{O} \cup \mathcal{O}'$ is a deductive conservative extension of $\mathcal{O} \cup \mathcal{O}'_1$ w.r.t. $\mathcal{L}$. In particular, $\mathcal{O} \cup \mathcal{O}'$ is a deductive $\mathbf{S}$-conservative extension of $\mathcal{O} \cup \mathcal{O}'_1$ w.r.t. $\mathcal{L}$ for $\mathbf{S} = \mathsf{Sig}(\mathcal{O})$, which implies, by Definition 10, that $\mathcal{O}'_1$ is a module for $\mathcal{O}$ in $\mathcal{O}'$ w.r.t. $\mathcal{L}$. $\square$

We also provide the analog of Proposition 17 for the notions of safety and modules for a signature:

Proposition 18 (Safety vs. Modules for a Signature) *Let $\mathcal{L}$ be an ontology language, $\mathcal{O}'$ and $\mathcal{O}'_1 \subseteq \mathcal{O}'$, ontologies over $\mathcal{L}$, and $\mathbf{S}$ a subset of the signature of $\mathcal{L}$. Then:*

1. *$\mathcal{O}'$ is safe for $\mathbf{S}$ w.r.t. $\mathcal{L}$ iff the empty ontology $\mathcal{O}'_0 = \emptyset$ is an $\mathbf{S}$-module in $\mathcal{O}'$ w.r.t. $\mathcal{L}$.*

2. *If $\mathcal{O}' \setminus \mathcal{O}'_1$ is safe for $\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}'_1)$ w.r.t. $\mathcal{L}$, then $\mathcal{O}'_1$ is an $\mathbf{S}$-module in $\mathcal{O}'$ w.r.t. $\mathcal{L}$.*

Proof. 1. By Definition 6, $\mathcal{O}'$ is safe for $\mathbf{S}$ w.r.t. $\mathcal{L}$ iff (a) for every $\mathcal{O}$ with $\mathsf{Sig}(\mathcal{O}') \cap \mathsf{Sig}(\mathcal{O}) \subseteq \mathbf{S}$, it is the case that $\mathcal{O}'$ is safe for $\mathcal{O}$ w.r.t. $\mathcal{L}$. By Definition 12, $\mathcal{O}'_0 = \emptyset$ is an $\mathbf{S}$-module in $\mathcal{O}'$ w.r.t. $\mathcal{L}$ iff (b) for every $\mathcal{O}$ with $\mathsf{Sig}(\mathcal{O}') \cap \mathsf{Sig}(\mathcal{O}) \subseteq \mathbf{S}$, it is the case that $\mathcal{O}'_0 = \emptyset$ is a module for $\mathcal{O}$ in $\mathcal{O}'$ w.r.t. $\mathcal{L}$. By Claim 1 of Proposition 17, it is easy to see that (a) is equivalent to (b).

2. By Definition 6, $\mathcal{O}' \setminus \mathcal{O}'_1$ is safe for $\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}'_1)$ w.r.t. $\mathcal{L}$ iff (c) for every $\mathcal{O}$, with $\mathsf{Sig}(\mathcal{O}' \setminus \mathcal{O}'_1) \cap \mathsf{Sig}(\mathcal{O}) \subseteq \mathbf{S} \cup \mathsf{Sig}(\mathcal{O}'_1)$, we have that $\mathcal{O}' \setminus \mathcal{O}'_1$ is safe for $\mathcal{O}$ w.r.t. $\mathcal{L}$. By

Definition 12, $\mathcal{O}'_1$ is an $\mathbf{S}$-module in $\mathcal{O}'$ w.r.t. $\mathcal{L}$ iff (d) for every $\mathcal{O}$ with $\mathsf{Sig}(\mathcal{O}')\cap\mathsf{Sig}(\mathcal{O}) \subseteq \mathbf{S}$, we have that $\mathcal{O}'_1$ is a module for $\mathcal{O}$ in $\mathcal{O}'$ w.r.t. $\mathcal{L}$.

In order to prove that (c) implies (d), let $\mathcal{O}$ be such that $\mathsf{Sig}(\mathcal{O}')\cap\mathsf{Sig}(\mathcal{O}) \subseteq \mathbf{S}$. We need to demonstrate that $\mathcal{O}'_1$ is a module for $\mathcal{O}$ in $\mathcal{O}'$ w.r.t. $\mathcal{L}$. $(\star)$

Let $\tilde{\mathcal{O}} := \mathcal{O}\cup\mathcal{O}'_1$. Note that $\mathsf{Sig}(\mathcal{O}'\setminus\mathcal{O}'_1)\cap\mathsf{Sig}(\tilde{\mathcal{O}}) = \mathsf{Sig}(\mathcal{O}'\setminus\mathcal{O}'_1)\cap(\mathsf{Sig}(\mathcal{O})\cup\mathsf{Sig}(\mathcal{O}'_1)) \subseteq (\mathsf{Sig}(\mathcal{O}'\setminus\mathcal{O}'_1)\cap\mathsf{Sig}(\mathcal{O}))\cup\mathsf{Sig}(\mathcal{O}'_1) \subseteq \mathbf{S}\cup\mathsf{Sig}(\mathcal{O}'_1)$. Hence, by (c) we have that $\mathcal{O}'\setminus\mathcal{O}'_1$ is safe for $\tilde{\mathcal{O}} = \mathcal{O}\cup\mathcal{O}'_1$ w.r.t. $\mathcal{L}$ which implies by Claim 2 of Proposition 17 that $\mathcal{O}'_1$ is a module for $\mathcal{O}$ in $\mathcal{O}'$ w.r.t. $\mathcal{L}$ $(\star)$. $\square$

4. Undecidability and Complexity Results

In this section we study the computational properties of the tasks in Table 2 for ontology languages that correspond to fragments of the description logic $\mathcal{SHOIQ}$. We demonstrate that most of these reasoning tasks are algorithmically unsolvable even for relatively inexpressive DLs, and are computationally hard for simple DLs.

Since the notions of modules and safety are defined in Section 3 using the notion of deductive conservative extension, it is reasonable to identify which (un)decidability and complexity results for conservative extensions are applicable to the reasoning tasks in Table 2. The computational properties of conservative extensions have recently been studied in the context of description logics. Given $\mathcal{O}_1 \subseteq \mathcal{O}$ over a language $\mathcal{L}$, the problem of deciding whether $\mathcal{O}$ is a deductive conservative extension of $\mathcal{O}_1$ w.r.t. $\mathcal{L}$ is 2-EXPTIME-complete for $\mathcal{L} = \mathcal{ALC}$ (Ghilardi et al., 2006). This result was extended by Lutz et al. (2007), who showed that the problem is 2-EXPTIME-complete for $\mathcal{L} = \mathcal{ALCIQ}$ and undecidable for $\mathcal{L} = \mathcal{ALCIQO}$. Recently, the problem has also been studied for simple DLs; it has been shown that deciding deductive conservative extensions is EXPTIME-complete for $\mathcal{L} = \mathcal{EL}$(Lutz & Wolter, 2007). These results can be immediately applied to the notions of safety and modules for an ontology:

Proposition 19 *Given ontologies $\mathcal{O}$ and $\mathcal{O}'$ over $\mathcal{L}$, the problem of determining whether $\mathcal{O}$ is safe for $\mathcal{O}'$ w.r.t. $\mathcal{L}$ is EXPTIME-complete for $\mathcal{L} = \mathcal{EL}$, 2-EXPTIME-complete for $\mathcal{L} = \mathcal{ALC}$ and $\mathcal{L} = \mathcal{ALCIQ}$, and undecidable for $\mathcal{L} = \mathcal{ALCIQO}$. Given ontologies $\mathcal{O}$, $\mathcal{O}'$, and $\mathcal{O}'_1 \subseteq \mathcal{O}'$ over $\mathcal{L}$, the problem of determining whether $\mathcal{O}'_1$ is a module for $\mathcal{O}$ in $\mathcal{O}'$ is EXPTIME-complete for $\mathcal{L} = \mathcal{EL}$, 2-EXPTIME complete for $\mathcal{L} = \mathcal{ALC}$ and $\mathcal{L} = \mathcal{ALCIQ}$, and undecidable for $\mathcal{L} = \mathcal{ALCIQO}$.*

Proof. By Definition 2, an ontology $\mathcal{O}$ is safe for $\mathcal{O}'$ w.r.t. $\mathcal{L}$ iff $\mathcal{O}\cup\mathcal{O}'$ is a deductive conservative extension of $\mathcal{O}'$ w.r.t. $\mathcal{L}$. By Definition 2, an ontology $\mathcal{O}'_1$ is a module for $\mathcal{O}$ in $\mathcal{O}'$ w.r.t. $\mathcal{L}$ if $\mathcal{O}\cup\mathcal{O}'$ is a deductive $\mathbf{S}$-conservative extension of $\mathcal{O}\cup\mathcal{O}'_1$ for $\mathbf{S} = \mathsf{Sig}(\mathcal{O})$ w.r.t. $\mathcal{L}$. Hence, any algorithm for checking deductive conservative extensions can be reused for checking safety and modules.

Conversely, we demonstrate that any algorithm for checking safety or modules can be used for checking deductive conservative extensions. Indeed, $\mathcal{O}$ is a deductive conservative extension of $\mathcal{O}_1 \subseteq \mathcal{O}$ w.r.t. $\mathcal{L}$ iff $\mathcal{O}\setminus\mathcal{O}_1$ is safe for $\mathcal{O}_1$ w.r.t. $\mathcal{L}$ iff, by Claim 1 of Proposition 17, $\mathcal{O}_0 = \emptyset$ is a module for $\mathcal{O}_1$ in $\mathcal{O}\setminus\mathcal{O}_1$ w.r.t. $\mathcal{L}$. $\square$

Corollary 20 *There exist algorithms performing tasks T1, and T3[a,s,u]m from Table 2 for $\mathcal{L} = \mathcal{EL}$ and $\mathcal{L} = \mathcal{ALCIQ}$ that run in EXPTIME and 2-EXPTIME respectively. There is no algorithm performing tasks T1, or T3[a,s,u]m from Table 2 for $\mathcal{L} = \mathcal{ALCIQO}$.*

Proof. The task T1 corresponds directly to the problem of checking the safety of an ontology, as given in Definition 2.

Suppose that we have a 2-EXPTIME algorithm that, given ontologies $\mathcal{O}$, $\mathcal{O}'$ and $\mathcal{O}'_1 \subseteq \mathcal{O}'$, determines whether $\mathcal{O}'_1$ is a module for $\mathcal{O}$ in $\mathcal{O}'$ w.r.t. $\mathcal{L} = \mathcal{ALCIQ}$. We demonstrate that this algorithm can be used to solve the reasoning tasks T3am, T3sm and T3um for $\mathcal{L} = \mathcal{ALCIQ}$ in 2-EXPTIME. Indeed, given ontologies $\mathcal{O}$ and $\mathcal{O}'$, one can enumerate all subsets of $\mathcal{O}'$ and check in 2-EXPTIME which of these subsets are modules for $\mathcal{O}$ in $\mathcal{O}'$ w.r.t. $\mathcal{L}$. Then we can determine which of these modules are minimal and return all of them, one of them, or the union of them depending on the reasoning task.

Finally, we prove that solving any of the reasoning tasks T3am, T3sm and T3um is not easier than checking the safety of an ontology. Indeed, by Claim 1 of Proposition 17, an ontology $\mathcal{O}$ is safe for $\mathcal{O}'$ w.r.t. $\mathcal{L}$ iff $\mathcal{O}_0 = \emptyset$ is a module for $\mathcal{O}'$ in $\mathcal{O}$ w.r.t. $\mathcal{L}$. Note that the empty ontology $\mathcal{O}_0 = \emptyset$ is a module for $\mathcal{O}'$ in $\mathcal{O}$ w.r.t. $\mathcal{L}$ iff $\mathcal{O}_0 = \emptyset$ is the only minimal module for $\mathcal{O}'$ in $\mathcal{O}$ w.r.t. $\mathcal{L}$. □

We have demonstrated that the reasoning tasks T1 and T3[a,s,u]m are computationally unsolvable for DLs that are as expressive as $\mathcal{ALCQIO}$, and are 2-EXPTIME-hard for $\mathcal{ALC}$. In the remainder of this section, we focus on the computational properties of the reasoning tasks T2 and T4[a,s,u]m related to the notions of safety and modules for a signature. We demonstrate that all of these reasoning tasks are undecidable for DLs that are as expressive as $\mathcal{ALCO}$.

Theorem 21 (Undecidability of Safety for a Signature) *The problem of checking whether an ontology $\mathcal{O}$ consisting of a single $\mathcal{ALC}$-axiom is safe for a signature* $\mathbf{S}$ *is undecidable w.r.t. $\mathcal{L} = \mathcal{ALCO}$.*

Proof. The proof is a variation of the construction for undecidability of deductive conservative extensions in $\mathcal{ALCQIO}$ (Lutz et al., 2007), based on a reduction to a domino tiling problem.

A domino system is a triple $D = (T, H, V)$ where $T = \{1, \ldots, k\}$ is a finite set of *tiles* and $H, V \subseteq T \times T$ are *horizontal* and *vertical matching relations*. A *solution* for a domino system D is a mapping $t_{i,j}$ that assigns to every pair of integers $i, j \geq 1$ an element of T, such that $\langle t_{i,j}, t_{i,j+1} \rangle \in H$ and $\langle t_{i,j}, t_{i+1,j} \rangle \in V$. A *periodic solution* for a domino system D is a solution $t_{i,j}$ for which there exist integers $m \geq 1$, $n \geq 1$ called *periods* such that $t_{i+m,j} = t_{i,j}$ and $t_{i,j+n} = t_{i,j}$ for every $i, j \geq 1$.

Let $\mathcal{D}$ be the set of all domino systems, $\mathcal{D}_s$ be the subset of $\mathcal{D}$ that admit a solution and $\mathcal{D}_{ps}$ be the subset of $\mathcal{D}_s$ that admit a periodic solution. It is well-known (Börger, Grädel, & Gurevich, 1997, Theorem 3.1.7) that the sets $\mathcal{D} \setminus \mathcal{D}_s$ and $\mathcal{D}_{ps}$ are *recursively inseparable*, that is, there is no recursive (i.e. decidable) subset $\mathcal{D}' \subseteq \mathcal{D}$ of domino systems such that $\mathcal{D}_{ps} \subseteq \mathcal{D}' \subseteq \mathcal{D}_s$.

We use this property in our reduction. For each domino system D, we construct a signature $\mathbf{S} = \mathbf{S}(D)$, and an ontology $\mathcal{O} = \mathcal{O}(D)$ which consists of a single $\mathcal{ALC}$-axiom such that:

(q_1)	$\top \sqsubseteq A_1 \sqcup \cdots \sqcup A_k$	where $T = \{1, \ldots, k\}$
(q_2)	$A_t \sqcap A_{t'} \sqsubseteq \bot$	$1 \leq t < t' \leq k$
(q_3)	$A_t \sqsubseteq \exists r_H.(\bigsqcup_{\langle t,t'\rangle \in H} A_{t'})$	$1 \leq t \leq k$
(q_4)	$A_t \sqsubseteq \exists r_V.(\bigsqcup_{\langle t,t'\rangle \in V} A_{t'})$	$1 \leq t \leq k$

Figure 2: An ontology $\mathcal{O}_{\text{tile}} = \mathcal{O}_{\text{tile}}(D)$ expressing tiling conditions for a domino system D

(a) if D does not have a solution then $\mathcal{O} = \mathcal{O}(D)$ is safe for $\mathbf{S} = \mathbf{S}(D)$ w.r.t. $\mathcal{L} = \mathcal{ALCO}$, and

(b) if D has a periodic solution then $\mathcal{O} = \mathcal{O}(D)$ is not safe for $\mathbf{S} = \mathbf{S}(D)$ w.r.t. $\mathcal{L} = \mathcal{ALCO}$.

In other words, for the set $\mathcal{D}'$ consisting of the domino systems D such that $\mathcal{O} = \mathcal{O}(D)$ is not safe for $\mathbf{S} = \mathbf{S}(D)$ w.r.t. $\mathcal{L} = \mathcal{ALCO}$, we have $\mathcal{D}_{ps} \subseteq \mathcal{D}' \subseteq \mathcal{D}_s$. Since $\mathcal{D} \setminus \mathcal{D}_s$ and $\mathcal{D}_{ps}$ are recursively inseparable, this implies undecidability for $\mathcal{D}'$ and hence for the problem of checking if $\mathcal{O}$ is $\mathbf{S}$-safe w.r.t. $\mathcal{L} = \mathcal{ALCO}$, because otherwise one can use this problem for deciding membership in $\mathcal{D}'$.

The signature $\mathbf{S} = \mathbf{S}(D)$, and the ontology $\mathcal{O} = \mathcal{O}(D)$ are constructed as follows. Given a domino system $D = (T, H, V)$, let $\mathbf{S}$ consist of fresh atomic concepts A_t for every $t \in T$ and two atomic roles r_H and r_V. Consider an ontology $\mathcal{O}_{\text{tile}}$ in Figure 2 constructed for D. Note that $\mathsf{Sig}(\mathcal{O}_{\text{tile}}) = \mathbf{S}$. The axioms of $\mathcal{O}_{\text{tile}}$ express the tiling conditions for a domino system D, namely (q_1) and (q_2) express that every domain element is assigned with a unique tile $t \in T$; (q_3) and (q_4) express that every domain element has horizontal and vertical matching successors.

Now let s be an atomic role and B an atomic concept such that $s, B \notin \mathbf{S}$. Let $\mathcal{O} := \{\beta\}$ where:

$$\beta := \quad \top \sqsubseteq \exists s.\left[\bigsqcup_{(C_i \sqsubseteq D_i) \in \mathcal{O}_{\text{tile}}} (C_i \sqcap \neg D_i) \sqcup (\exists r_H.\exists r_V.B \sqcap \exists r_V.\exists r_H.\neg B)\right]$$

We say that r_H and r_V *commute in an interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ if for all domain elements a, b, c, d_1 and d_2 from $\Delta^{\mathcal{I}}$ such that $\langle a, b\rangle \in {r_H}^{\mathcal{I}}$, $\langle b, d_1\rangle \in {r_V}^{\mathcal{I}}$, $\langle a, c\rangle \in {r_V}^{\mathcal{I}}$, and $\langle c, d_2\rangle \in {r_H}^{\mathcal{I}}$, we have $d_1 = d_2$. The following claims can be easily proved:

Claim 1. If $\mathcal{O}_{\text{tile}}(D)$ has a model $\mathcal{I}$ in which r_H and r_V commute, then D has a solution.

Indeed a model $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$ of $\mathcal{O}_{\text{tile}}(D)$ can be used to guide the construction of a solution $t_{i,j}$ for D as follows. For every $i, j \geq 1$, we construct $t_{i,j}$ inductively together with elements $a_{i,j} \in \Delta^{\mathcal{I}}$ such that $\langle a_{i,j}, a_{i,j+1}\rangle \in {r_V}^{\mathcal{I}}$ and $\langle a_{i,j}, a_{i+1,j}\rangle \in {r_H}^{\mathcal{I}}$. We set $a_{1,1}$ to any element from $\Delta^{\mathcal{I}}$.

Now suppose $a_{i,j}$ with $i, j \geq 1$ is constructed. Since $\mathcal{I}$ is a model of axioms (q_1) and (q_2) from Figure 2, we have a unique A_t with $1 \leq t \leq k$ such that $a_{i,j} \in {A_t}^{\mathcal{I}}$. Then we set $t_{i,j} := t$. Since $\mathcal{I}$ is a model of axioms (q_3) and (q_4) and $a_{i,j} \in {A_t}^{\mathcal{I}}$ there exist $b, c \in \Delta^{\mathcal{I}}$ and $t', t'' \in T$ such that $\langle a_{i,j}, b\rangle \in {r_H}^{\mathcal{I}}$, $\langle a_{i,j}, c\rangle \in {r_V}^{\mathcal{I}}$, $\langle t, t'\rangle \in H$, $\langle t, t''\rangle \in V$, $b \in {A_{t'}}^{\mathcal{I}}$, and $c \in {A_{t''}}^{\mathcal{I}}$. In this case we assign $a_{i,j+1} := b$, $a_{i+1,j} := c$, $t_{i,j+1} := t'$, and $t_{i+1,j} := t''$. Note that some values of $a_{i,j}$ and $t_{i,j}$ can be assigned two times: $a_{i+1,j+1}$ and $t_{i+1,j+1}$ are constructed for $a_{i,j+1}$ and $a_{i,j+1}$. However, since r_V and r_H commute in $\mathcal{I}$, the value for

$a_{i+1,j+1}$ is unique, and because of (q_2), the value for $t_{i+1,j+1}$ is unique. It is easy to see that $t_{i,j}$ is a solution for D.

Claim 2. If $\mathcal{I}$ is a model of $\mathcal{O}_{\text{tile}} \cup \mathcal{O}$, then r_H and r_V do not commute in $\mathcal{I}$.

Indeed, it is easy to see that $\mathcal{O}_{\text{tile}} \cup \mathcal{O} \models (\top \sqsubseteq \exists s.[\exists r_H.\exists r_V.B \sqcap \exists r_V.\exists r_H.\neg B])$. Hence, if $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is a model of $\mathcal{O}_{\text{tile}} \cup \mathcal{O}$, then there exist a, b, c, d_1 and d_2 such that $\langle x, a\rangle \in s^{\mathcal{I}}$ for every $x \in \Delta^{\mathcal{I}}$, $\langle a, b\rangle \in {r_H}^{\mathcal{I}}$, $\langle b, d_1\rangle \in {r_V}^{\mathcal{I}}$, $d_1 \in B^{\mathcal{I}}$, $\langle a, c\rangle \in {r_V}^{\mathcal{I}}$, $\langle c, d_2\rangle \in {r_H}^{\mathcal{I}}$, and $d_2 \in (\neg B)^{\mathcal{I}}$. This implies that $d_1 \neq d_2$, and so, r_h and r_V do not commute in $\mathcal{I}$.

Finally, we demonstrate that $\mathcal{O} = \mathcal{O}(D)$ satisfies properties (a) and (b).

In order to prove property (a) we use the sufficient condition for safety given in Proposition 8 and demonstrate that if D has no solution then for every interpretation $\mathcal{I}$ there exists a model $\mathcal{J}$ of $\mathcal{O}$ such that $\mathcal{J}|_{\mathbf{S}} = \mathcal{I}|_{\mathbf{S}}$. By Proposition 8, this will imply that $\mathcal{O}$ is safe for $\mathbf{S}$ w.r.t. $\mathcal{L}$.

Let $\mathcal{I}$ be an arbitrary interpretation. Since D has no solution, then by the contra-positive of Claim 1 either (1) $\mathcal{I}$ is not a model of $\mathcal{O}_{\text{tile}}$, or (2) r_H and r_V do not commute in $\mathcal{I}$. We demonstrate for both of these cases how to construct the required model $\mathcal{J}$ of $\mathcal{O}$ such that $\mathcal{J}|_{\mathbf{S}} = \mathcal{I}|_{\mathbf{S}}$.

Case (1). If $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is not a model of $\mathcal{O}_{\text{tile}}$ then there exists an axiom $(C_i \sqsubseteq D_i) \in \mathcal{O}_{\text{tile}}$ such that $\mathcal{I} \not\models (C_i \sqsubseteq D_i)$. That is, there exists a domain element $a \in \Delta^{\mathcal{I}}$ such that $a \in C_i^{\mathcal{I}}$ but $a \notin D_i^{\mathcal{I}}$. Let us define $\mathcal{J}$ to be identical to $\mathcal{I}$ except for the interpretation of the atomic role s which we define in $\mathcal{J}$ as $s^{\mathcal{J}} = \{\langle x, a\rangle \mid x \in \Delta\}$. Since the interpretations of the symbols in $\mathbf{S}$ have remained unchanged, we have $a \in C_i^{\mathcal{J}}$, $a \in \neg D_i^{\mathcal{J}}$, and so $\mathcal{J} \models (\top \sqsubseteq \exists s.[C_i \sqcap \neg D_j])$. This implies that $\mathcal{J} \models \beta$, and so, we have constructed a model $\mathcal{J}$ of $\mathcal{O}$ such that $\mathcal{J}|_{\mathbf{S}} = \mathcal{I}|_{\mathbf{S}}$.

Case (2). Suppose that r_H and r_V do not commute in $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$. This means that there exist domain elements a, b, c, d_1 and d_2 from $\Delta^{\mathcal{I}}$ with $\langle a, b\rangle \in {r_H}^{\mathcal{I}}$, $\langle b, d_1\rangle \in {r_V}^{\mathcal{I}}$, $\langle a, c\rangle \in {r_V}^{\mathcal{I}}$, and $\langle c, d_2\rangle \in {r_H}^{\mathcal{I}}$, such that $d_1 \neq d_2$. Let us define $\mathcal{J}$ to be identical to $\mathcal{I}$ except for the interpretation of the atomic role s and the atomic concept B. We interpret s in $\mathcal{J}$ as $s^{\mathcal{J}} = \{\langle x, a\rangle \mid x \in \Delta\}$. We interpret B in $\mathcal{J}$ as $B^{\mathcal{J}} = \{d_1\}$. Note that $a \in (\exists r_H.\exists r_V.B)^{\mathcal{J}}$ and $a \in (\exists r_V.\exists r_H.\neg B)^{\mathcal{J}}$ since $d_1 \neq d_2$. So, we have $\mathcal{J} \models (\top \sqsubseteq \exists s.[\exists r_H.\exists r_V.B \sqcap \exists r_V.\exists r_H.\neg B])$ which implies that $\mathcal{J} \models \beta$, and thus, we have constructed a model $\mathcal{J}$ of $\mathcal{O}$ such that $\mathcal{J}|_{\mathbf{S}} = \mathcal{I}|_{\mathbf{S}}$.

In order to prove property (b), assume that D has a periodic solution $t_{i,j}$ with the periods $m, n \geq 1$. We demonstrate that $\mathcal{O}$ is not $\mathbf{S}$-safe w.r.t. $\mathcal{L}$. For this purpose we construct an $\mathcal{ALCO}$-ontology $\mathcal{O}'$ with $\mathsf{Sig}(\mathcal{O}) \cap \mathsf{Sig}(\mathcal{O}') \subseteq \mathbf{S}$ such that $\mathcal{O} \cup \mathcal{O}' \models (\top \sqsubseteq \bot)$, but $\mathcal{O}' \not\models (\top \sqsubseteq \bot)$. This will imply that $\mathcal{O}$ is not safe for $\mathcal{O}'$ w.r.t. $\mathcal{L} = \mathcal{ALCO}$, and, hence, is not safe for $\mathbf{S}$ w.r.t. $\mathcal{L} = \mathcal{ALCO}$.

We define $\mathcal{O}'$ such that every model of $\mathcal{O}'$ is a finite encoding of the periodic solution $t_{i,j}$ with the periods m and n. For every pair (i, j) with $1 \leq i \leq m$ and $1 \leq j \leq n$ we introduce a fresh individual $a_{i,j}$ and define $\mathcal{O}'$ to be an extension of $\mathcal{O}_{\text{tile}}$ with the following axioms:

$$
\begin{array}{lll}
(p_1)\ \{a_{i_1,j}\} \sqsubseteq \exists r_V.\{a_{i_2,j}\} & (p_2)\ \{a_{i_1,j}\} \sqsubseteq \forall r_V.\{a_{i_2,j}\}, & i_2 = i_1 + 1 \mod m \\
(p_3)\ \{a_{i,j_1}\} \sqsubseteq \exists r_H.\{a_{i,j_2}\} & (p_4)\ \{a_{i,j_1}\} \sqsubseteq \forall r_H.\{a_{i,j_2}\}, & j_2 = j_1 + 1 \mod n \\
 & (p_5)\ \top \sqsubseteq \bigsqcup_{1 \leq i \leq m,\, 1 \leq j \leq n} \{a_{i,j}\} &
\end{array}
$$

The purpose of axioms (p_1)–(p_5) is to ensure that r_H and r_V commute in every model of $\mathcal{O}'$. It is easy to see that $\mathcal{O}'$ has a model corresponding to every periodic solution for D with periods m and n. Hence $\mathcal{O}' \not\models (\top \sqsubseteq \bot)$. On the other hand, by Claim 2, since $\mathcal{O}'$ contains $\mathcal{O}_{\text{tile}}$, then in every model of $\mathcal{O}' \cup \mathcal{O}$, r_H and r_V do not commute. This is only possible if $\mathcal{O}' \cup \mathcal{O}$ have no models, so $\mathcal{O}' \cup \mathcal{O} \models (\top \sqsubseteq \bot)$. □

A direct consequence of Theorem 21 and Proposition 18 is the undecidability of the problem of checking whether a subset of an ontology is a module for a signature:

Corollary 22 *Given a signature* $\mathbf{S}$ *and* $\mathcal{ALC}$*-ontologies* $\mathcal{O}'$ *and* $\mathcal{O}'_1 \subseteq \mathcal{O}'$*, the problem of determining whether* $\mathcal{O}'_1$ *is an* $\mathbf{S}$*-module in* $\mathcal{O}'$ *w.r.t.* $\mathcal{L} = \mathcal{ALCO}$ *is undecidable.*

Proof. By Claim 1 of Proposition 18, $\mathcal{O}$ is $\mathbf{S}$-safe w.r.t. $\mathcal{L}$ if $\mathcal{O}_0 = \emptyset$ is an $\mathbf{S}$-module in $\mathcal{O}$ w.r.t. $\mathcal{L}$. Hence any algorithm for recognizing modules for a signature in $\mathcal{L}$ can be used for checking if an ontology is safe for a signature in $\mathcal{L}$. □

Corollary 23 *There is no algorithm that can perform tasks T2, or T4[a,s,u]m for* $\mathcal{L} = \mathcal{ALCO}$*.*

Proof. Theorem 21 directly implies that there is no algorithm for task T2, since this task corresponds to the problem of checking safety for a signature.

Solving any of the reasoning tasks T4am, T4sm, or T4um for $\mathcal{L}$ is at least as hard as checking the safety of an ontology, since, by Claim 1 of Proposition 18, an ontology $\mathcal{O}$ is $\mathbf{S}$-safe w.r.t. $\mathcal{L}$ iff $\mathcal{O}_0 = \emptyset$ is (the only minimal) $\mathbf{S}$-module in $\mathcal{O}$ w.r.t. $\mathcal{L}$. □

5. Sufficient Conditions for Safety

Theorem 21 establishes the undecidability of checking whether an ontology expressed in OWL DL is safe w.r.t. a signature. This undecidability result is discouraging and leaves us with two alternatives: First, we could focus on simple DLs for which this problem is decidable. Alternatively, we could look for sufficient conditions for the notion of safety—that is, if an ontology satisfies our conditions, then we can guarantee that it is safe; the converse, however, does not necessarily hold.

The remainder of this paper focuses on the latter approach. Before we go any further, however, it is worth noting that Theorem 21 still leaves room for investigating the former approach. Indeed, safety may still be decidable for weaker description logics, such as $\mathcal{EL}$, or even for very expressive logics such as $\mathcal{SHIQ}$. In the case of $\mathcal{SHIQ}$, however, existing results (Lutz et al., 2007) strongly indicate that checking safety is likely to be exponentially harder than reasoning and practical algorithms may be hard to design. This said, in what follows we will focus on defining sufficient conditions for safety that we can use in practice and restrict ourselves to OWL DL—that is, $\mathcal{SHOIQ}$—ontologies.

5.1 Safety Classes

In general, any sufficient condition for safety can be defined by giving, for each signature $\mathbf{S}$, the set of ontologies over a language that satisfy the condition for that signature. These ontologies are then guaranteed to be safe for the signature under consideration. These intuitions lead to the notion of a safety class.

Definition 24 (Class of Ontologies, Safety Class). A *class of ontologies for a language* $\mathcal{L}$ is a function $\mathbf{O}(\cdot)$ that assigns to every subset $\mathbf{S}$ of the signature of $\mathcal{L}$, a subset $\mathbf{O}(\mathbf{S})$ of ontologies in $\mathcal{L}$. A class $\mathbf{O}(\cdot)$ is *anti-monotonic* if $\mathbf{S}_1 \subseteq \mathbf{S}_2$ implies $\mathbf{O}(\mathbf{S}_2) \subseteq \mathbf{O}(\mathbf{S}_1)$; it is *compact* when $\mathcal{O} \in \mathbf{O}(\mathbf{S} \cap \mathsf{Sig}(\mathcal{O}))$ for each $\mathcal{O} \in \mathbf{O}(\mathbf{S})$; and it is *subset-closed* if $\mathcal{O}_1 \subseteq \mathcal{O}_2$ and $\mathcal{O}_2 \in \mathbf{O}(\mathbf{S})$ implies $\mathcal{O}_1 \in \mathbf{O}(\mathbf{S})$; it is *union-closed* if $\mathcal{O}_1 \in \mathbf{O}(\mathbf{S})$ and $\mathcal{O}_2 \in \mathbf{O}(\mathbf{S})$ implies $(\mathcal{O}_1 \cup \mathcal{O}_2) \in \mathbf{O}(\mathbf{S})$.

A *safety class* (also called a *sufficient condition to safety*) for an ontology language $\mathcal{L}$ is a class of ontologies $\mathbf{O}(\cdot)$ for $\mathcal{L}$ such that for each $\mathbf{S}$ it is the case that *(i)* $\emptyset \in \mathbf{O}(\mathbf{S})$, and *(ii)* each ontology $\mathcal{O} \in \mathbf{O}(\mathbf{S})$ is $\mathbf{S}$-safe for $\mathcal{L}$. $\diamondsuit$

Intuitively, a class of ontologies is a collection of sets of ontologies parameterized by a signature. A safety class represents a sufficient condition for safety: each ontology in $\mathbf{O}(\mathbf{S})$ should be safe for $\mathbf{S}$. Also, w.l.o.g., we assume that the the empty ontology $\emptyset$ belongs to every safety class for every signature. In what follows, whenever an ontology $\mathcal{O}$ belongs to a safety class for a given signature and the safety class is clear from the context, we will sometimes say that $\mathcal{O}$ passes the safety test for $\mathbf{S}$.

Safety classes may admit many natural properties, as given in Definition 24. *Anti-monotonicity* intuitively means that if an ontology $\mathcal{O}$ can be proved to be safe w.r.t. $\mathbf{S}$ using the sufficient condition, then $\mathcal{O}$ can be proved to be safe w.r.t. every subset of $\mathbf{S}$. *Compactness* means that it is sufficient to consider only common elements of $\mathsf{Sig}(\mathcal{O})$ and $\mathbf{S}$ for checking safety. *Subset-closure* (*union closure*) means that if $\mathcal{O}$ ($\mathcal{O}_1$ and $\mathcal{O}_2$) satisfy the sufficient condition for safety, then every subset of $\mathcal{O}$ (the union of $\mathcal{O}_1$ and $\mathcal{O}_2$) also satisfies this condition.

5.2 Locality

In this section we introduce a family of safety classes for $\mathcal{L} = \mathcal{SHOIQ}$ that is based on the semantic properties underlying the notion of model conservative extensions. In Section 3, we have seen that, according to Proposition 8, one way to prove that $\mathcal{O}$ is $\mathbf{S}$-safe is to show that $\mathcal{O}$ is a model $\mathbf{S}$-conservative extension of the empty ontology.

The following definition formalizes the classes of ontologies, called local ontologies, for which safety can be proved using Proposition 8.

Definition 25 (Class of Interpretations, Locality). Given a $\mathcal{SHOIQ}$ signature $\mathbf{S}$, we say that a set of interpretations $\mathbf{I}$ is *local w.r.t.* $\mathbf{S}$ if for every $\mathcal{SHOIQ}$-interpretation $\mathcal{I}$ there exists an interpretation $\mathcal{J} \in \mathbf{I}$ such that $\mathcal{I}|_{\mathbf{S}} = \mathcal{J}|_{\mathbf{S}}$.

A *class of interpretations* is a function $\mathbf{I}(\cdot)$ that given a $\mathcal{SHOIQ}$ signature $\mathbf{S}$ returns a set of interpretations $\mathbf{I}(\mathbf{S})$; it is *local* if $\mathbf{I}(\mathbf{S})$ is local w.r.t. $\mathbf{S}$ for every $\mathbf{S}$; it is *monotonic* if $\mathbf{S}_1 \subseteq \mathbf{S}_2$ implies $\mathbf{I}(\mathbf{S}_1) \subseteq \mathbf{I}(\mathbf{S}_2)$; it is *compact* if for every $\mathbf{S}_1$, $\mathbf{S}_2$ and $\mathbf{S}$ such that $(\mathbf{S}_1 \vartriangle \mathbf{S}_2) \cap \mathbf{S} = \emptyset$ we have that $\mathbf{I}(\mathbf{S}_1)|_{\mathbf{S}} = \mathbf{I}(\mathbf{S}_2)|_{\mathbf{S}}$, where $\mathbf{S}_1 \vartriangle \mathbf{S}_2$ is the *symmetric difference* of sets $\mathbf{S}_1$ and $\mathbf{S}_2$ defined by $\mathbf{S}_1 \vartriangle \mathbf{S}_2 := \mathbf{S}_1 \setminus \mathbf{S}_2 \cup \mathbf{S}_2 \setminus \mathbf{S}_1$.

Given a class of interpretations $\mathbf{I}(\cdot)$, we say that $\mathbf{O}(\cdot)$ is the class of ontologies $\mathbf{O}(\cdot)$ *based on* $\mathbf{I}(\cdot)$ if for every $\mathbf{S}$, $\mathbf{O}(\mathbf{S})$ is the set of ontologies that are valid in $\mathbf{I}(\mathbf{S})$; if $\mathbf{I}(\cdot)$ is local then we say that $\mathbf{O}(\cdot)$ is a *class of local ontologies*, and for every $\mathbf{S}$ and $\mathcal{O} \in \mathbf{O}(\mathbf{S})$ and every $\alpha \in \mathcal{O}$, we say that $\mathcal{O}$ and α are *local* (based on $\mathbf{I}(\cdot)$). $\diamondsuit$

Example 26 Let $\mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\cdot)$ be a class of $\mathcal{SHOIQ}$ interpretations defined as follows. Given a signature $\mathbf{S}$, the set $\mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$ consists of interpretations $\mathcal{J}$ such that $r^{\mathcal{J}} = \emptyset$ for every atomic role $r \notin \mathbf{S}$ and $A^{\mathcal{J}} = \emptyset$ for every atomic concept $A \notin \mathbf{S}$. It is easy to show that $\mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$ is local for every $\mathbf{S}$, since for every interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ and the interpretation $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ defined by $\Delta^{\mathcal{J}} := \Delta^{\mathcal{I}}$, $r^{\mathcal{J}} = \emptyset$ for $r \notin \mathbf{S}$, $A^{\mathcal{J}} = \emptyset$ for $A \notin \mathbf{S}$, and $X^{\mathcal{J}} := X^{\mathcal{I}}$ for the remaining symbols X, then $\mathcal{J} \in \mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$ and $\mathcal{I}|_{\mathbf{S}} = \mathcal{J}|_{\mathbf{S}}$. Since $\mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S}_1) \subseteq \mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S}_2)$ for every $\mathbf{S}_1 \subseteq \mathbf{S}_2$, it is the case that $\mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\cdot)$ is monotonic; $\mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\cdot)$ is also compact, since for every $\mathbf{S}_1$ and $\mathbf{S}_2$ the sets of interpretations $\mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S}_1)$ and $\mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S}_2)$ are defined differently only for elements in $\mathbf{S}_1 \vartriangle \mathbf{S}_2$.

Given a signature $\mathbf{S}$, the set $\mathbf{Ax}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$ of axioms that are local w.r.t. $\mathbf{S}$ based on $\mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$ consists of all axioms α such for every $\mathcal{J} \in \mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$, it is the case that $\mathcal{J} \models \alpha$. Then the class of local ontologies based on $\mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\cdot)$ is defined by $\mathcal{O} \in \mathbf{O}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$ iff $\mathcal{O} \subseteq \mathbf{Ax}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$. $\diamondsuit$

Proposition 27 (Locality Implies Safety) *Let $\mathbf{O}(\cdot)$ be a class of ontologies for $\mathcal{SHOIQ}$ based on a local class of interpretations $\mathbf{I}(\cdot)$. Then $\mathbf{O}(\cdot)$ is a subset-closed and union-closed safety class for $\mathcal{L} = \mathcal{SHOIQ}$. Additionally, if $\mathbf{I}(\cdot)$ is monotonic, then $\mathbf{O}(\cdot)$ is anti-monotonic, and if $\mathbf{I}(\cdot)$ is compact then $\mathbf{O}(\cdot)$ is also compact.*

Proof. Assume that $\mathbf{O}(\cdot)$ is a class of ontologies based on $\mathbf{I}(\cdot)$. Then by Definition 25 for every $\mathcal{SHOIQ}$ signature $\mathbf{S}$, we have $\mathcal{O} \in \mathbf{O}(\mathbf{S})$ iff $\mathcal{O}$ is valid in $\mathbf{I}(\mathbf{S})$ iff $\mathcal{J} \models \mathcal{O}$ for every interpretation $\mathcal{J} \in \mathbf{I}(\mathbf{S})$. Since $\mathbf{I}(\cdot)$ is a local class of interpretations, we have that for every $\mathcal{SHOIQ}$-interpretation $\mathcal{I}$ there exists $\mathcal{J} \in \mathbf{I}(\mathbf{S})$ such that $\mathcal{J}|_{\mathbf{S}} = \mathcal{I}|_{\mathbf{S}}$. Hence for every $\mathcal{O} \in \mathbf{I}(\mathbf{S})$ and every $\mathcal{SHOIQ}$ interpretation $\mathcal{I}$ there is a model $\mathcal{J} \in \mathbf{I}(\mathbf{S})$ such that $\mathcal{J}|_{\mathbf{S}} = \mathcal{I}|_{\mathbf{S}}$, which implies by Proposition 8 that $\mathcal{O}$ is safe for $\mathbf{S}$ w.r.t. $\mathcal{L} = \mathcal{SHOIQ}$. Thus $\mathbf{O}(\cdot)$ is a safety class.

The fact that $\mathbf{O}(\cdot)$ is subset-closed and union-closed follows directly from Definition 25 since $(\mathcal{O}_1 \cup \mathcal{O}_2) \in \mathbf{O}(\mathbf{S})$ iff $(\mathcal{O}_1 \cup \mathcal{O}_2)$ is valid in $\mathbf{I}(\mathbf{S})$ iff $\mathcal{O}_1$ and $\mathcal{O}_2$ are valid in $\mathbf{I}(\mathbf{S})$ iff $\mathcal{O}_1 \in \mathbf{O}(\mathbf{S})$ and $\mathcal{O}_2 \in \mathbf{O}(\mathbf{S})$. If $\mathbf{I}(\cdot)$ is monotonic then $\mathbf{I}(\mathbf{S}_1) \subseteq \mathbf{I}(\mathbf{S}_2)$ for every $\mathbf{S}_1 \subseteq \mathbf{S}_2$, and so $\mathcal{O} \in \mathbf{O}(\mathbf{S}_2)$ implies that $\mathcal{O}$ is valid in $\mathbf{I}(\mathbf{S}_2)$ which implies that $\mathcal{O}$ is valid in $\mathbf{I}(\mathbf{S}_1)$ which implies that $\mathcal{O} \in \mathbf{O}(\mathbf{S}_1)$. Hence $\mathbf{O}(\cdot)$ is anti-monotonic.

If $\mathbf{I}(\cdot)$ is compact then for every $\mathbf{S}_1$, $\mathbf{S}_2$ and $\mathbf{S}$ such that $(\mathbf{S}_1 \vartriangle \mathbf{S}_2) \cap \mathbf{S} = \emptyset$ we have that $\mathbf{I}(\mathbf{S}_1)|_{\mathbf{S}} = \mathbf{I}(\mathbf{S}_2)|_{\mathbf{S}}$, hence for every $\mathcal{O}$ with $\mathsf{Sig}(\mathcal{O}) \subseteq \mathbf{S}$ we have that $\mathcal{O}$ is valid in $\mathbf{I}(\mathbf{S}_1)$ iff $\mathcal{O}$ is valid in $\mathbf{I}(\mathbf{S}_2)$, and so, $\mathcal{O} \in \mathbf{O}(\mathbf{S}_1)$ iff $\mathcal{O} \in \mathbf{O}(\mathbf{S}_2)$. In particular, $\mathcal{O} \in \mathbf{O}(\mathbf{S})$ iff $\mathcal{O} \in \mathbf{O}(\mathbf{S} \cap \mathsf{Sig}(\mathcal{O}))$ since $(\mathbf{S} \vartriangle (\mathbf{S} \cap \mathsf{Sig}(\mathcal{O}))) \cap \mathsf{Sig}(\mathcal{O}) = (\mathbf{S} \setminus \mathsf{Sig}(\mathcal{O})) \cap \mathsf{Sig}(\mathcal{O}) = \emptyset$. Hence, $\mathbf{O}(\cdot)$ is compact. $\square$

Corollary 28 *The class of ontologies $\mathbf{O}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$ defined in Example 26 is an anti-monotonic compact subset-closed and union-closed safety class.*

Example 29 Recall that in Example 5 from Section 3, we demonstrated that the ontology $\mathcal{P}_1 \cup \mathcal{Q}$ given in Figure 1 with $\mathcal{P}_1 = \{\text{P1}, \ldots, \text{P4}, \text{E1}\}$ is a deductive conservative extension of $\mathcal{Q}$ for $\mathbf{S} = \{\mathsf{Cystic\_Fibrosis}, \mathsf{Genetic\_Disorder}\}$. This has been done by showing that every $\mathbf{S}$-interpretation $\mathcal{I}$ can be expanded to a model $\mathcal{J}$ of axioms P1–P4, E1 by interpreting the symbols in $\mathsf{Sig}(\mathcal{P}_1) \setminus \mathbf{S}$ as the empty set. In terms of Example 26 this means that

$\mathcal{P}_1 \in \mathbf{O}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$. Since $\mathbf{O}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\cdot)$ is a class of local ontologies, by Proposition 27, the ontology $\mathcal{P}_1$ is safe for $\mathbf{S}$ w.r.t. $\mathcal{L} = \mathcal{SHOIQ}$. ◊

Proposition 27 and Example 29 suggest a particular way of proving the safety of ontologies. Given a $\mathcal{SHOIQ}$ ontology $\mathcal{O}$ and a signature $\mathbf{S}$ it is sufficient to check whether $\mathcal{O} \in \mathbf{O}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$; that is, whether every axiom α in $\mathcal{O}$ is satisfied by every interpretation from $\mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$. If this property holds, then $\mathcal{O}$ must be safe for $\mathbf{S}$ according to Proposition 27. It turns out that this notion provides a powerful sufficiency test for safety which works surprisingly well for many real-world ontologies, as will be shown in Section 8. In the next section we discuss how to perform this test in practice.

5.3 Testing Locality

In this section, we focus in more detail on the safety class $\mathbf{O}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\cdot)$, introduced in Example 26. When ambiguity does not arise, we refer to this safety class simply as locality.[2]

From the definition of $\mathbf{Ax}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$ given in Example 26 it is easy to see that an axiom α is local w.r.t. $\mathbf{S}$ (based on $\mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$) if α is satisfied in every interpretation that fixes the interpretation of all atomic roles and concepts outside $\mathbf{S}$ to the empty set. Note that for defining locality we do not fix the interpretation of the individuals outside $\mathbf{S}$, but in principle, this could be done. The reason is that there is no elegant way to describe such interpretations. Namely, every individual needs to be interpreted as an element of the domain, and there is no "canonical" element of every domain to choose.

In order to test the locality of α w.r.t. $\mathbf{S}$, it is sufficient to interpret every atomic concept and atomic role not in $\mathbf{S}$ as the empty set and then check if α is satisfied in all interpretations of the remaining symbols. This observation suggests the following test for locality:

Proposition 30 (Testing Locality) *Given a $\mathcal{SHOIQ}$ signature $\mathbf{S}$, concept C, axiom α and ontology $\mathcal{O}$ let $\tau(C,\mathbf{S})$, $\tau(\alpha,\mathbf{S})$ and $\tau(\mathcal{O},\mathbf{S})$ be defined recursively as follows:*

$$
\begin{array}{lllr}
\tau(C,\mathbf{S}) ::= & \tau(\top,\mathbf{S}) & = \top; & (a)\\
& \mid \tau(A,\mathbf{S}) & = \bot \textit{ if } A \notin \mathbf{S} \textit{ and otherwise } = A; & (b)\\
& \mid \tau(\{a\},\mathbf{S}) & = \{a\}; & (c)\\
& \mid \tau(C_1 \sqcap C_2,\mathbf{S}) & = \tau(C_1,\mathbf{S}) \sqcap \tau(C_2,\mathbf{S}); & (d)\\
& \mid \tau(\neg C_1,\mathbf{S}) & = \neg\tau(C_1,\mathbf{S}); & (e)\\
& \mid \tau(\exists R.C_1,\mathbf{S}) & = \bot \textit{ if } \mathsf{Sig}(R) \nsubseteq \mathbf{S} \textit{ and otherwise} = \exists R.\tau(C_1,\mathbf{S}); & (f)\\
& \mid \tau(\geqslant n\, R.C_1,\mathbf{S}) & = \bot \textit{ if } \mathsf{Sig}(R) \nsubseteq \mathbf{S} \textit{ and otherwise} = (\geqslant n\, R.\tau(C_1,\mathbf{S})). & (g)\\
\tau(\alpha,\mathbf{S}) ::= & \tau(C_1 \sqsubseteq C_2,\mathbf{S}) & = (\tau(C_1,\mathbf{S}) \sqsubseteq \tau(C_2,\mathbf{S})); & (h)\\
& \mid \tau(R_1 \sqsubseteq R_2,\mathbf{S}) & = (\bot \sqsubseteq \bot) \textit{ if } \mathsf{Sig}(R_1) \nsubseteq \mathbf{S},\ \textit{otherwise} & \\
& & = \exists R_1.\top \sqsubseteq \bot \textit{ if } \mathsf{Sig}(R_2) \nsubseteq \mathbf{S},\ \textit{otherwise } = (R_1 \sqsubseteq R_2); & (i)\\
& \mid \tau(a : C,\mathbf{S}) & = a : \tau(C,\mathbf{S}); & (j)\\
& \mid \tau(r(a,b),\mathbf{S}) & = \top \sqsubseteq \bot \textit{ if } r \notin \mathbf{S} \textit{ and otherwise } = r(a,b); & (k)\\
& \mid \tau(\mathsf{Trans}(r),\mathbf{S}) & = \bot \sqsubseteq \bot \textit{ if } r \notin \mathbf{S} \textit{ and otherwise } = \mathsf{Trans}(r); & (l)\\
& \mid \tau(\mathsf{Funct}(R),\mathbf{S}) & = \bot \sqsubseteq \bot \textit{ if } \mathsf{Sig}(R) \nsubseteq \mathbf{S} \textit{ and otherwise } = \mathsf{Funct}(R). & (m)\\
\tau(\mathcal{O},\mathbf{S}) ::= & \bigcup_{\alpha\in\mathcal{O}} \tau(\alpha,\mathbf{S}) & & (n)
\end{array}
$$

2. This notion of locality is exactly the one we used in our previous work (Cuenca Grau et al., 2007).

Then, $\mathcal{O} \in \mathbf{O}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$ iff every axiom in $\tau(\mathcal{O}, \mathbf{S})$ is a tautology.

Proof. It is easy to check that for every atomic concept A and atomic role r from $\tau(C, \mathbf{S})$, we have $A \in \mathbf{S}$ and $r \in \mathbf{S}$, in other words, all atomic concepts and roles that are not in $\mathbf{S}$ are eliminated by the transformation.[3] It is also easy to show by induction that for every interpretation $\mathcal{I} \in \mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$, we have $C^{\mathcal{I}} = (\tau(C, \mathbf{S}))^{\mathcal{I}}$ and that $\mathcal{I} \models \alpha$ iff $\mathcal{I} \models \tau(\alpha, \mathbf{S})$. Hence an axiom α is local w.r.t. $\mathbf{S}$ iff $\mathcal{I} \models \alpha$ for every interpretation $\mathcal{I} \in \mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$ iff $\mathcal{I} \models \tau(\alpha, \mathbf{S})$ for every $\mathsf{Sig}(\alpha)$-interpretation $\mathcal{I} \in \mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$ iff $\tau(\alpha, \mathbf{S})$ is a tautology. $\square$

Example 31 Let $\mathcal{O} = \{\alpha\}$ be the ontology consisting of axiom $\alpha =$ M2 from Figure 1. We demonstrate using Proposition 30 that $\mathcal{O}$ is local w.r.t. $\mathbf{S}_1 = \{\mathsf{Fibrosis}, \mathsf{Genetic\_Origin}\}$, but not local w.r.t. $\mathbf{S}_2 = \{\mathsf{Genetic\_Fibrosis}, \mathsf{has\_Origin}\}$.

Indeed, according to Proposition 30, in order to check whether $\mathcal{O}$ is local w.r.t. $\mathbf{S}_1$ it is sufficient to perform the following replacements in α (the symbols from $\mathbf{S}_1$ are underlined):

$$\text{M2} \quad \overbrace{\mathsf{Genetic\_Fibrosis}}^{\bot\ [\text{by (b)}]} \equiv \underline{\mathsf{Fibrosis}} \sqcap \overbrace{\exists \mathsf{has\_Origin}.\underline{\mathsf{Genetic\_Origin}}}^{\bot\ [\text{by (f)}]} \tag{7}$$

Similarly, in order to check whether $\mathcal{O}$ is local w.r.t. $\mathbf{S}_2$, it is sufficient to perform the following replacements in α (the symbols from $\mathbf{S}_2$ are underlined):

$$\text{M2} \quad \underline{\mathsf{Genetic\_Fibrosis}} \equiv \overbrace{\mathsf{Fibrosis}}^{\bot\ [\text{by (b)}]} \sqcap \exists \underline{\mathsf{has\_Origin}}.\overbrace{\mathsf{Genetic\_Origin}}^{\bot\ [\text{by (b)}]} \tag{8}$$

In the first case we obtain $\tau(\text{M2}, \mathbf{S}_1) = (\bot \equiv \mathsf{Fibrosis} \sqcap \bot)$ which is a $\mathcal{SHOIQ}$-tautology. Hence $\mathcal{O}$ is local w.r.t. $\mathbf{S}_1$ and hence by Proposition 8 is $\mathbf{S}_1$-safe w.r.t. $\mathcal{SHOIQ}$. In the second case $\tau(\text{M2}, \mathbf{S}_2) = (\mathsf{Genetic\_Fibrosis} \equiv \bot \sqcap \exists \mathsf{has\_Origin}.\bot)$ which is not a $\mathcal{SHOIQ}$ tautology, hence $\mathcal{O}$ is not local w.r.t. $\mathbf{S}_2$. $\Diamond$

5.4 A Tractable Approximation to Locality

One of the important conclusions of Proposition 30 is that one can use the standard capabilities of available DL-reasoners such as FaCT++ (Tsarkov & Horrocks, 2006), RACER (Möller & Haarslev, 2003), Pellet (Sirin & Parsia, 2004) or KAON2 (Motik, 2006) for testing locality since these reasoners, among other things, allow testing for DL-tautologies. Checking for tautologies in description logics is, theoretically, a difficult problem (e.g. for the DL $\mathcal{SHOIQ}$ it is known to be NEXPTIME-complete, Tobies, 2000). There are, however, several reasons to believe that the locality test would perform well in practice. The primary reason is that the sizes of the axioms which need to be tested for tautologies are usually relatively small compared to the sizes of ontologies. Secondly, modern DL reasoners are highly optimized for standard reasoning tasks and behave well for most realistic ontologies. In case reasoning is too costly, it is possible to formulate a tractable approximation to the locality conditions for $\mathcal{SHOIQ}$:

3. Recall that the constructors $\bot$, $C_1 \sqcup C_2$, $\forall R.C$, and $(\leqslant n\, R.C)$ are assumed to be expressed using $\top$, $C_1 \sqcap C_2$, $\exists R.C$ and $(\geqslant n\, R.C)$, hence, in particular, every role $R^\emptyset$ with $\mathsf{Sig}(R^\emptyset) \nsubseteq \mathbf{S}$ occurs in $\mathcal{O}$ as either $\exists R^\emptyset.C$, $(\geqslant n\, R^\emptyset.C)$, $R^\emptyset \sqsubseteq R$, $R \sqsubseteq R^\emptyset$, $\mathsf{Trans}(R^\emptyset)$, or $\mathsf{Funct}(R^\emptyset)$, and hence will be eliminated. All atomic concepts $A^\emptyset \notin \mathbf{S}$ will be eliminated likewise. Note that it is not necessarily the case that $\mathsf{Sig}(\tau(\alpha, \mathbf{S})) \subseteq \mathbf{S}$, since $\tau(\alpha, \mathbf{S})$ may still contain individuals that do not occur in $\mathbf{S}$.

Definition 32 (Syntactic Locality for $\mathcal{SHOIQ}$). Let $\mathbf{S}$ be a signature. The following grammar recursively defines two sets of concepts $\mathbf{Con}^{\emptyset}(\mathbf{S})$ and $\mathbf{Con}^{\Delta}(\mathbf{S})$ for a signature $\mathbf{S}$:

$$\mathbf{Con}^{\emptyset}(\mathbf{S}) ::= A^{\emptyset} \mid \neg C^{\Delta} \mid C \sqcap C^{\emptyset} \mid C^{\emptyset} \sqcap C \mid \exists R^{\emptyset}.C \mid \exists R.C^{\emptyset} \mid (\geqslant n\, R^{\emptyset}.C) \mid (\geqslant n\, R.C^{\emptyset})\,.$$
$$\mathbf{Con}^{\Delta}(\mathbf{S}) ::= \top \mid \neg C^{\emptyset} \mid C_1^{\Delta} \sqcap C_2^{\Delta}\,.$$

where $A^{\emptyset} \notin \mathbf{S}$ is an atomic concept, $R^{\emptyset}$ (possibly the inverse of) an atomic role $r^{\emptyset} \notin \mathbf{S}$, C any concept, R any role, $C^{\emptyset} \in \mathbf{Con}^{\emptyset}(\mathbf{S})$, $C_{(i)}^{\Delta} \in \mathbf{Con}^{\Delta}(\mathbf{S})$, and $i \in \{1, 2\}$.

An axiom α is *syntactically local w.r.t.* $\mathbf{S}$ if it is of one of the following forms: (1) $R^{\emptyset} \sqsubseteq R$, or (2) $\mathsf{Trans}(r^{\emptyset})$, or (3) $\mathsf{Funct}(R^{\emptyset})$, or (4) $C^{\emptyset} \sqsubseteq C$, or (5) $C \sqsubseteq C^{\Delta}$, or (6) $a : C^{\Delta}$. We denote by $\tilde{\mathbf{Ax}}{}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$ the set of all $\mathcal{SHOIQ}$-axioms that are syntactically local w.r.t. $\mathbf{S}$.

A $\mathcal{SHOIQ}$-ontology $\mathcal{O}$ is *syntactically local w.r.t.* $\mathbf{S}$ if $\mathcal{O} \subseteq \tilde{\mathbf{Ax}}{}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$. We denote by $\tilde{\mathbf{O}}{}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$ the set of all $\mathcal{SHOIQ}$ ontologies that are syntactically local w.r.t. $\mathbf{S}$. ◊

Intuitively, syntactic locality provides a simple syntactic test to ensure that an axiom is satisfied in every interpretation from $\mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$. It is easy to see from the inductive definitions of $\mathbf{Con}^{\emptyset}(\mathbf{S})$ and $\mathbf{Con}^{\Delta}(\mathbf{S})$ in Definition 32 that for every interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ from $\mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$ it is the case that $(C^{\emptyset})^{\mathcal{I}} = \emptyset$ and $(C^{\Delta})^{\mathcal{I}} = \Delta^{\mathcal{I}}$ for every $C^{\emptyset} \in \mathbf{Con}^{\emptyset}(\mathbf{S})$ and $C^{\Delta} \in \mathbf{Con}^{\Delta}(\mathbf{S})$. Hence, every syntactically local axiom is satisfied in every interpretation $\mathcal{I}$ from $\mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$, and so we obtain the following conclusion:

Proposition 33 $\tilde{\mathbf{Ax}}{}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S}) \subseteq \mathbf{Ax}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$.

Further, it can be shown that the safety class for $\mathcal{SHOIQ}$ based on syntactic locality enjoys all of the properties from Definition 24:

Proposition 34 *The class of syntactically local ontologies $\tilde{\mathbf{O}}{}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\cdot)$ given in Definition 32 is an anti-monotonic, compact, subset-closed, and union-closed safety class.*

Proof. $\tilde{\mathbf{O}}{}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\cdot)$ is a safety class by Proposition 33. Anti-monotonicity for $\tilde{\mathbf{O}}{}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\cdot)$ can be shown by induction, by proving that $\mathbf{Con}^{\emptyset}(\mathbf{S}_2) \subseteq \mathbf{Con}^{\emptyset}(\mathbf{S}_1)$, $\mathbf{Con}^{\Delta}(\mathbf{S}_2) \subseteq \mathbf{Con}^{\Delta}(\mathbf{S}_1)$ and $\tilde{\mathbf{Ax}}{}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S}_2) \subseteq \tilde{\mathbf{Ax}}{}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S}_1)$ when $\mathbf{S}_1 \subseteq \mathbf{S}_2$. Also one can show by induction that $\alpha \in \tilde{\mathbf{Ax}}{}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$ iff $\alpha \in \tilde{\mathbf{Ax}}{}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S} \cap \mathsf{Sig}(\alpha))$, so $\tilde{\mathbf{O}}{}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\cdot)$ is compact. Since $\mathcal{O} \in \tilde{\mathbf{O}}{}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$ iff $\mathcal{O} \subseteq \tilde{\mathbf{Ax}}{}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$, we have that $\tilde{\mathbf{O}}{}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\cdot)$ is subset-closed and union-closed. ☐

Example 35 (Example 31 continued) It is easy to see that axiom M2 from Figure 1 is syntactically local w.r.t. $\mathbf{S}_1 = \{\mathsf{Fibrosis}, \mathsf{Genetic\_Origin}\}$. Below we indicate sub-concepts in α from $\mathbf{Con}^{\emptyset}(\mathbf{S}_1)$:

$$\text{M2} \quad \overbrace{\mathsf{Genetic\_Fibrosis}}^{\in\, \mathbf{Con}^{\emptyset}(\mathbf{S}_1)\ [\text{matches } A^{\emptyset}]} \equiv \underbrace{\underline{\mathsf{Fibrosis}} \sqcap \overbrace{\exists \mathsf{has\_Origin}.\underline{\mathsf{Genetic\_Origin}}}^{\in\, \mathbf{Con}^{\emptyset}(\mathbf{S}_1)\ [\text{matches } \exists R^{\emptyset}.C]}}_{\in\, \mathbf{Con}^{\emptyset}(\mathbf{S}_1)\ [\text{matches } C \sqcap C^{\emptyset}]} \tag{9}$$

It is easy to show in a similar way that axioms P_1—P4, and E1 from Figure 1 are syntactically local w.r.t. $\mathbf{S} = \{\mathsf{Cystic\_Fibrosis}, \mathsf{Genetic\_Disorder}\}$. Hence the ontology $\mathcal{P}_1 = \{\text{P1}, \ldots, \text{P4}, \text{E1}\}$ considered in Example 29 is syntactically local w.r.t. $\mathbf{S}$. ◊

$\mathbf{I}^{r\leftarrow *}_{A\leftarrow *}(\mathbf{S})$	$r, A \notin \mathbf{S}$: $r^{\mathcal{J}}$	$A^{\mathcal{J}}$
$\mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$	$\emptyset$	$\emptyset$
$\mathbf{I}^{r\leftarrow\Delta\times\Delta}_{A\leftarrow\emptyset}(\mathbf{S})$	$\Delta^{\mathcal{J}} \times \Delta^{\mathcal{J}}$	$\emptyset$
$\mathbf{I}^{r\leftarrow\mathbf{id}}_{A\leftarrow\emptyset}(\mathbf{S})$	$\{\langle x, x\rangle \mid x \in \Delta^{\mathcal{J}}\}$	$\emptyset$

$\mathbf{I}^{r\leftarrow *}_{A\leftarrow *}(\mathbf{S})$	$r, A \notin \mathbf{S}$: $r^{\mathcal{J}}$	$A^{\mathcal{J}}$
$\mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\Delta}(\mathbf{S})$	$\emptyset$	$\Delta^{\mathcal{J}}$
$\mathbf{I}^{r\leftarrow\Delta\times\Delta}_{A\leftarrow\Delta}(\mathbf{S})$	$\Delta^{\mathcal{J}} \times \Delta^{\mathcal{J}}$	$\Delta^{\mathcal{J}}$
$\mathbf{I}^{r\leftarrow\mathbf{id}}_{A\leftarrow\Delta}(\mathbf{S})$	$\{\langle x, x\rangle \mid x \in \Delta^{\mathcal{J}}\}$	$\Delta^{\mathcal{J}}$

Table 3: Examples for Different Local Classes of Interpretations

The converse of Proposition 33 does not hold in general since there are semantically local axioms that are not syntactically local. For example, the axiom $\alpha = (A \sqsubseteq A \sqcup B)$ is local w.r.t. every $\mathbf{S}$ since it is a tautology (and hence true in every interpretation). On the other hand, it is easy to see that α is not syntactically local w.r.t. $\mathbf{S} = \{A, B\}$ according to Definition 32 since it involves symbols in $\mathbf{S}$ only. Another example, which is not a tautology, is a GCI $\alpha = (\exists r.\neg A \sqsubseteq \exists r.\neg B)$. The axiom α is semantically local w.r.t. $\mathbf{S} = \{r\}$, since $\tau(\alpha, \mathbf{S}) = (\exists r.\neg\bot \sqsubseteq \exists r.\neg\bot)$ is a tautology, but not syntactically local. These examples show that the limitation of our syntactic notion of locality is its inability to "compare" different occurrences of concepts from the given signature $\mathbf{S}$. As a result, syntactic locality does not detect all tautological axioms. It is reasonable to assume, however, that tautological axioms do not occur often in realistic ontologies. Furthermore, syntactic locality checking can be performed in polynomial time by matching an axiom according to Definition 32.

Proposition 36 *There exists an algorithm that given a $\mathcal{SHOIQ}$ ontology $\mathcal{O}$ and a signature $\mathbf{S}$, determines whether $\mathcal{O} \in \tilde{\mathbf{O}}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$, and whose running time is polynomial in $|\mathcal{O}| + |\mathbf{S}|$, where $|\mathcal{O}|$ and $|\mathbf{S}|$ are the number of symbols occurring in $\mathcal{O}$ and $\mathbf{S}$ respectively.*[4]

5.5 Other Locality Classes

The locality condition given in Example 26 based on a class of local interpretations $\mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\cdot)$ is just a particular example of locality which can be used for testing safety. Other classes of local interpretations can be constructed in a similar way by fixing the interpretations of elements outside $\mathbf{S}$ to different values. In Table 3 we have listed several such classes of local interpretations where we fix the interpretation of atomic roles outside $\mathbf{S}$ to be either the empty set $\emptyset$, the universal relation $\Delta \times \Delta$, or the identity relation $\mathbf{id}$ on Δ, and the interpretation of atomic concepts outside $\mathbf{S}$ to either the empty set $\emptyset$ or the set Δ of all elements.

Each local class of interpretations in Table 3 defines a corresponding class of local ontologies analogously to Example 26. In Table 4 we have listed all of these classes together with examples of typical types of axioms used in ontologies. These axioms are assumed to be an extension of our project ontology from Figure 1. We indicate which axioms are local w.r.t. $\mathbf{S}$ for which locality conditions assuming, as usual, that the symbols from $\mathbf{S}$ are underlined.

It can be seen from Table 4 that different types of locality conditions are appropriate for different types of axioms. The locality condition based on $\mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$ captures the domain axiom P4, definition P5, the disjointness axiom P6, and the functionality axiom P7 but

4. We assume that the numbers in number restrictions are written down using binary coding.

α	Axiom	? $\alpha \in \mathbf{Ax}$	$\boldsymbol{r\leftarrow\emptyset}$ $\boldsymbol{A\leftarrow\emptyset}$	$r\leftarrow\Delta\times\Delta$ $A\leftarrow\emptyset$	$r\leftarrow\mathbf{id}$ $A\leftarrow\emptyset$	$r\leftarrow\emptyset$ $A\leftarrow\Delta$	$r\leftarrow\Delta\times\Delta$ $A\leftarrow\Delta$	$r\leftarrow\mathbf{id}$ $A\leftarrow\Delta$
P4	$\exists$has_Focus.$\top \sqsubseteq$ Project		✓	✗	✗	✓	✓	✓
P5	BioMedical_Project $\equiv$ Project $\sqcap$ $\sqcap$ $\exists$has_Focus.$\underline{\text{Bio\_Medicine}}$		✓	✓	✓	✗	✗	✗
P6	Project $\sqcap$ $\underline{\text{Bio\_Medicine}}$ $\sqsubseteq \bot$		✓	✓	✓	✗	✗	✗
P7	Funct(has_Focus)		✓	✗	✓	✓	✗	✓
P8	*Human_Genome* : Project		✗	✗	✗	✓	✓	✓
P9	has_Focus(*Human_Genome*, $\underline{\textit{Gene}}$)		✗	✓	✗	✗	✓	✗
E2	$\forall$has_Focus.$\underline{\text{Cystic\_Fibrosis}}$ $\sqsubseteq$ $\sqsubseteq$ $\exists$has_Focus.$\underline{\text{Cystic\_Fibrosis}}$		✗	✗	✗	✗	✗	✗

Table 4: A Comparison between Different Types of Locality Conditions

neither of the assertions P8 or P9, since the individuals *Human_Genome* and *Gene* prevent us from interpreting the atomic role has_Focus and atomic concept Project with the empty set. The locality condition based on $\mathbf{I}^{r\leftarrow\Delta\times\Delta}_{A\leftarrow\Delta}(\mathbf{S})$, where the atomic roles and concepts outside $\mathbf{S}$ are interpreted as the largest possible sets, can capture such assertions but are generally poor for other types of axioms. For example, the functionality axiom P7 is not captured by this locality condition since the atomic role has_Focus is interpreted as the universal relation $\Delta\times\Delta$, which is not necessarily functional. In order to capture such functionality axioms, one can use locality based on $\mathbf{I}^{r\leftarrow\mathbf{id}}_{A\leftarrow\emptyset}(\mathbf{S})$ or $\mathbf{I}^{r\leftarrow\mathbf{id}}_{A\leftarrow\Delta}(\mathbf{S})$, where every atomic role outside $\mathbf{S}$ is interpreted with the identity relation $\mathbf{id}$ on the interpretation domain. Note that the modeling error E2 is not local for any of the given locality conditions. Note also that it is not possible to come up with a locality condition that captures all the axioms P4–P9, since in P6 and P8 together imply axiom $\beta = (\{\textit{Human\_Genome}\} \sqcap \text{Bio\_Medicine} \sqsubseteq \bot)$ which uses the symbols from $\mathbf{S}$ only. Hence, every subset of $\mathcal{P}$ containing P6 and P8 is not safe w.r.t. $\mathbf{S}$, and so cannot be local w.r.t. $\mathbf{S}$.

It might be possible to come up with algorithms for testing locality conditions for the classes of interpretation in Table 3 similar to the ones presented in Proposition 30. For example, locality based on the class $\mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\Delta}(\mathbf{S})$ can be tested as in Proposition 30, where the case (a) of the definition for $\tau(C,\mathbf{S})$ is replaced with the following:

$$\tau(A,\mathbf{S}) \qquad = \top \text{ if } A \notin \mathbf{S} \text{ and otherwise } = A \qquad (a')$$

For the remaining classes of interpretations, that is for $\mathbf{I}^{r\leftarrow\Delta\times\Delta}_{A\leftarrow *}(\mathbf{S})$ and $\mathbf{I}^{r\leftarrow\mathbf{id}}_{A\leftarrow *}(\mathbf{S})$, checking locality, however, is not that straightforward, since it is not clear how to eliminate the universal roles and identity roles from the axioms and preserve validity in the respective classes of interpretations.

Still, it is easy to come up with tractable syntactic approximations for all the locality conditions considered in this section in a similar manner to what has been done in Section 5.4. The idea is the same as that used in Definition 32, namely to define two sets $\mathbf{Con}^{\emptyset}(\mathbf{S})$ and $\mathbf{Con}^{\Delta}(\mathbf{S})$ of concepts for a signature $\mathbf{S}$ which are interpreted as the empty

$$
\begin{array}{lll}
\mathbf{Con}^{\emptyset}(\mathbf{S}) ::= & \neg C^{\Delta} \mid C \sqcap C^{\emptyset} \mid C^{\emptyset} \sqcap C \\
 & \mid \exists R.C^{\emptyset} \mid \geqslant n\, R.C^{\emptyset} \\
\text{for } \mathbf{I}^{r\leftarrow *}_{A\leftarrow \emptyset}(\cdot): & \mid A^{\emptyset} \\
\text{for } \mathbf{I}^{r\leftarrow \emptyset}_{A\leftarrow *}(\cdot): & \mid \exists R^{\emptyset}.C \mid \geqslant n\, R^{\emptyset}.C \\
\text{for } \mathbf{I}^{r\leftarrow \mathbf{id}}_{A\leftarrow *}(\cdot): & \mid (\geqslant m\, R^{\mathbf{id}}.C),\ m \geq 2\,.
\end{array}
\qquad
\begin{array}{lll}
\mathbf{Con}^{\Delta}(\mathbf{S}) ::= & \top \mid \neg C^{\emptyset} \mid C^{\Delta}_1 \sqcap C^{\Delta}_2 \\
\text{for } \mathbf{I}^{r\leftarrow *}_{A\leftarrow \Delta}(\cdot): & \mid A^{\Delta} \\
\text{for } \mathbf{I}^{r\leftarrow \Delta\times\Delta}_{A\leftarrow *}(\cdot): & \mid \exists R^{\Delta\times\Delta}.C^{\Delta} \mid (\geqslant n\, R^{\Delta\times\Delta}.C^{\Delta}) \\
\text{for } \mathbf{I}^{r\leftarrow \mathbf{id}}_{A\leftarrow *}(\cdot): & \mid \exists R^{\mathbf{id}}.C^{\Delta} \mid (\geqslant 1\, R^{\mathbf{id}}.C^{\Delta})\,.
\end{array}
$$

$$
\begin{array}{ll}
\tilde{\mathbf{Ax}}^{r\leftarrow *}_{A\leftarrow *}(\mathbf{S}) ::= & C^{\emptyset} \sqsubseteq C \mid C \sqsubseteq C^{\Delta} \mid a : C^{\Delta} \\
\text{for } \mathbf{I}^{r\leftarrow \emptyset}_{A\leftarrow *}(\cdot): & \mid R^{\emptyset} \sqsubseteq R \mid \mathsf{Trans}(r^{\emptyset}) \mid \mathsf{Funct}(R^{\emptyset}) \\
\text{for } \mathbf{I}^{r\leftarrow \Delta\times\Delta}_{A\leftarrow *}(\cdot): & \mid R \sqsubseteq R^{\Delta\times\Delta} \mid \mathsf{Trans}(r^{\Delta\times\Delta}) \mid r^{\Delta\times\Delta}(a,b) \\
\text{for } \mathbf{I}^{r\leftarrow \mathbf{id}}_{A\leftarrow *}(\cdot): & \mid \mathsf{Trans}(r^{\mathbf{id}}) \mid \mathsf{Funct}(R^{\mathbf{id}})
\end{array}
\qquad
\begin{array}{l}
\text{Where:} \\
A^{\emptyset}, A^{\Delta}, r^{\emptyset}, r^{\Delta\times\Delta}, r^{\mathbf{id}} \notin \mathbf{S}; \\
\mathsf{Sig}(R^{\emptyset}), \mathsf{Sig}(R^{\Delta\times\Delta}), \mathsf{Sig}(R^{\mathbf{id}}) \nsubseteq \mathbf{S}; \\
C^{\emptyset} \in \mathbf{Con}^{\emptyset}(\mathbf{S}),\ C^{\Delta}_{(i)} \in \mathbf{Con}^{\Delta}(\mathbf{S}); \\
C \text{ is any concept, } R \text{ is any role}
\end{array}
$$

Figure 3: Syntactic Locality Conditions for the Classes of Interpretations in Table 3

set and, respectively, as $\Delta^{\mathcal{I}}$ in every interpretation $\mathcal{I}$ from the class and see in which situations DL-constructors produce the elements from these sets. In Figure 3 we gave recursive definitions for syntactically local axioms $\tilde{\mathbf{Ax}}^{r\leftarrow *}_{A\leftarrow *}(\mathbf{S})$ that correspond to classes $\mathbf{I}^{r\leftarrow *}_{A\leftarrow *}(\mathbf{S})$ of interpretations from Table 3, where some cases in the recursive definitions are present only for the indicated classes of interpretations.

5.6 Combining and Extending Safety Classes

In the previous section we gave examples of several safety classes based on different local classes of interpretations and demonstrated that different classes are suitable for different types of axioms. In order to check safety of ontologies in practice, one may try to apply different sufficient tests and check if any of them succeeds. Obviously, this gives a more powerful sufficient condition for safety, which can be seen as the union of the safety classes used in the tests.

Formally, given two classes of ontologies $\mathbf{O_1}(\cdot)$ and $\mathbf{O_2}(\cdot)$, their *union* $(\mathbf{O_1} \cup \mathbf{O_2})(\cdot)$ is a class of ontologies defined by $(\mathbf{O_1} \cup \mathbf{O_2})(\mathbf{S}) = \mathbf{O_1}(\mathbf{S}) \cup \mathbf{O_2}(\mathbf{S})$. It is easy to see by Definition 24 that if both $\mathbf{O_1}(\cdot)$ and $\mathbf{O_2}(\cdot)$ are safety classes then their union $(\mathbf{O_1} \cup \mathbf{O_2})(\cdot)$ is a safety class. Moreover, if each of the safety classes is also anti-monotonic or subset-closed, then the union is anti-monotonic, or respectively, subset-closed as well. Unfortunately the union-closure property for safety classes is not preserved under unions, as demonstrated in the following example:

Example 37 Consider the union $(\mathbf{O}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset} \cup \mathbf{O}^{r\leftarrow\Delta\times\Delta}_{A\leftarrow\Delta})(\cdot)$ of two classes of local ontologies $\mathbf{O}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\cdot)$ and $\mathbf{O}^{r\leftarrow\Delta\times\Delta}_{A\leftarrow\Delta}(\cdot)$ defined in Section 5.5. This safety class is not union-closed since, for example, the ontology $\mathcal{O}_1$ consisting of axioms P4–P7 from Table 4 satisfies the first locality condition, the ontology $\mathcal{O}_2$ consisting of axioms P8–P9 satisfies the second locality condition, but their union $\mathcal{O}_1 \cup \mathcal{O}_2$ satisfies neither the first nor the second locality condition and, in fact, is not even safe for $\mathbf{S}$ as we have shown in Section 5.5. ◊

As shown in Proposition 33, every locality condition gives a union-closed safety class; however, as seen in Example 37, the union of such safety classes might be no longer union-closed. One may wonder if locality classes already provide the most powerful sufficient conditions for safety that satisfy all the desirable properties from Definition 24. Surprisingly this is the case to a certain extent for some locality classes considered in Section 5.5.

Definition 38 (Maximal Union-Closed Safety Class). A safety class $\mathbf{O_2}(\cdot)$ *extends* a safety class $\mathbf{O_1}(\cdot)$ if $\mathbf{O_1}(\mathbf{S}) \subseteq \mathbf{O_2}(\mathbf{S})$ for every $\mathbf{S}$. A safety class $\mathbf{O_1}(\cdot)$ is *maximal union-closed for a language* $\mathcal{L}$ if $\mathbf{O_1}(\cdot)$ is union-closed and for every union-closed safety class $\mathbf{O_2}(\cdot)$ that extends $\mathbf{O_1}(\cdot)$ and every $\mathcal{O}$ over $\mathcal{L}$ we have that $\mathcal{O} \in \mathbf{O_2}(\mathbf{S})$ implies $\mathcal{O} \in \mathbf{O_1}(\mathbf{S})$. $\quad\Diamond$

Proposition 39 *The classes of local ontologies* $\mathbf{O}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\cdot)$ *and* $\mathbf{O}^{r\leftarrow\emptyset}_{A\leftarrow\Delta}(\cdot)$ *defined in Section 5.5 are maximal union-closed safety classes for* $\mathcal{L} = \mathcal{SHIQ}$.

Proof. According to Definition 38, if a safety class $\mathbf{O}(\cdot)$ is not maximal union-closed for a language $\mathcal{L}$, then there exists a signature $\mathbf{S}$ and an ontology $\mathcal{O}$ over $\mathcal{L}$, such that (i) $\mathcal{O} \notin \mathbf{O}(\mathbf{S})$, (ii) $\mathcal{O}$ is safe w.r.t. $\mathbf{S}$ in $\mathcal{L}$, and (iii) for every $\mathcal{P} \in \mathbf{O}(\mathbf{S})$, it is the case that $\mathcal{O} \cup \mathcal{P}$ is safe w.r.t. $\mathbf{S}$ in $\mathcal{L}$; that is, for every ontology $\mathcal{Q}$ over $\mathcal{L}$ with $\mathsf{Sig}(\mathcal{Q}) \cap \mathsf{Sig}(\mathcal{O} \cup \mathcal{P}) \subseteq \mathbf{S}$ it is the case that $\mathcal{O} \cup \mathcal{P} \cup \mathcal{Q}$ is a deductive conservative extension of $\mathcal{Q}$. We demonstrate that this is not possible for $\mathbf{O}(\cdot) = \mathbf{O}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\cdot)$ or $\mathbf{O}(\cdot) = \mathbf{O}^{r\leftarrow\emptyset}_{A\leftarrow\Delta}(\cdot)$

We first consider the case $\mathbf{O}(\cdot) = \mathbf{O}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\cdot)$ and then show how to modify the proof for the case $\mathbf{O}(\cdot) = \mathbf{O}^{r\leftarrow\emptyset}_{A\leftarrow\Delta}(\cdot)$.

Let $\mathcal{O}$ be an ontology over $\mathcal{L}$ that satisfies conditions (i)–(iii) above. We define ontologies $\mathcal{P}$ and $\mathcal{Q}$ as follows. Take $\mathcal{P}$ to consist of the axioms $A \sqsubseteq \bot$ and $\exists r.\top \sqsubseteq \bot$ for every atomic concept A and atomic role r from $\mathsf{Sig}(\mathcal{O}) \setminus \mathbf{S}$. It is easy to see that $\mathcal{P} \in \mathbf{O}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$. Take $\mathcal{Q}$ to consist of all tautologies of form $\bot \sqsubseteq A$ and $\bot \sqsubseteq \exists r.\top$ for every $A, r \in \mathbf{S}$. Note that $\mathsf{Sig}(\mathcal{O} \cup \mathcal{P}) \cap \mathsf{Sig}(\mathcal{Q}) \subseteq \mathbf{S}$. We claim that $\mathcal{O} \cup \mathcal{P} \cup \mathcal{Q}$ is not a deductive $\mathsf{Sig}(\mathcal{Q})$-conservative extension of $\mathcal{Q}$. $\quad(\sharp)$

Intuitively, the ontology $\mathcal{P}$ is chosen in such a way that $\mathcal{P} \in \mathbf{O}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$ and $\mathcal{O} \cup \mathcal{P} \cup \mathcal{Q}$ has only models from $\mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$. $\mathcal{Q}$ is an ontology which implies nothing but tautologies and uses all the atomic concepts and roles from $\mathbf{S}$.

Since $\mathcal{O} \notin \mathbf{O}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$, there exists an axiom $\alpha \in \mathcal{O}$ such that $\alpha \notin \mathbf{Ax}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$. Let $\beta := \tau(\alpha, \mathbf{S})$ where $\tau(\cdot,\cdot)$ is defined as in Proposition 30. As has been shown in the proof of this proposition, $\mathcal{I} \models \alpha$ iff $\mathcal{I} \models \beta$ for every $\mathcal{I} \in \mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$. Now, since $\mathcal{O} \models \alpha$ and $\mathcal{O} \cup \mathcal{P} \cup \mathcal{Q}$ has only models from $\mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$, it is the case that $\mathcal{O} \cup \mathcal{P} \cup \mathcal{Q} \models \beta$. By Proposition 30, since β does not contain individuals, we have that $\mathsf{Sig}(\beta) \subseteq \mathbf{S} = \mathsf{Sig}(\mathcal{Q})$ and, since $\alpha \notin \mathbf{Ax}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$, β is not a tautology, thus $\mathcal{Q} \not\models \beta$. Hence, by Definition 1, $\mathcal{O} \cup \mathcal{P} \cup \mathcal{Q}$ is not a deductive $\mathsf{Sig}(\mathcal{Q})$-conservative extension of $\mathcal{Q}$ $(\sharp)$.

For $\mathbf{O}(\cdot) = \mathbf{O}^{r\leftarrow\emptyset}_{A\leftarrow\Delta}(\cdot)$ the proof can be repeated by taking $\mathcal{P}$ to consist of axioms $\top \sqsubseteq A$ and $\exists r.\top \sqsubseteq \bot$ for all $A, r \in \mathsf{Sig}(\mathcal{O}) \setminus \mathbf{S}$, and modifying $\tau(\cdot,\cdot)$ as has been discussed in Section 5.5. $\square$

There are some difficulties in extending the proof of Proposition 39 to other locality classes considered in Section 5.5. First, it is not clear how to force interpretations of roles to be the universal or identity relation using $\mathcal{SHOIQ}$ axioms. Second, it is not clear how to define the function $\tau(\cdot,\cdot)$ in these cases (see a related discussion in Section 5.5). Note also that the proof of Proposition 39 does not work in the presence of nominals, since this will not guarantee that $\beta = \tau(\alpha, \mathbf{S})$ contains symbols from $\mathbf{S}$ only (see Footnote 3 on p. 297). Hence there is probably room to extend the locality classes $\mathbf{O}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\cdot)$ and $\mathbf{O}^{r\leftarrow\emptyset}_{A\leftarrow\Delta}(\cdot)$ for $\mathcal{L} = \mathcal{SHOIQ}$ while preserving union-closure.

6. Extracting Modules Using Safety Classes

In this section we revisit the problem of extracting modules from ontologies. As shown in Corollary 23 in Section 4, there exists no general procedure that can recognize or extract all the (minimal) modules for a signature in an ontology in finite time.

The techniques described in Section 5, however, can be reused for extracting particular families of modules that satisfy certain sufficient conditions. Proposition 18 establishes the relationship between the notions of safety and module; more precisely, a subset $\mathcal{O}_1$ of $\mathcal{O}$ is an $\mathbf{S}$-module in $\mathcal{O}$ provided that $\mathcal{O} \setminus \mathcal{O}_1$ is safe for $\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_1)$. Therefore, any safety class $\mathbf{O}(\cdot)$ can provide a sufficient condition for testing modules—that is, in order to prove that $\mathcal{O}_1$ is a $\mathbf{S}$-module in $\mathcal{O}$, it is sufficient to show that $\mathcal{O} \setminus \mathcal{O}_1 \in \mathbf{O}(\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_1))$. A notion of modules based on this property can be defined as follows.

Definition 40 (Modules Based on Safety Class).
Let $\mathcal{L}$ be an ontology language and $\mathbf{O}(\cdot)$ be a safety class for $\mathcal{L}$. Given an ontology $\mathcal{O}$ and a signature $\mathbf{S}$ over $\mathcal{L}$, we say that $\mathcal{O}_m \subseteq \mathcal{O}$ *is an* $\mathbf{O}(\cdot)$*-based* $\mathbf{S}$*-module in* $\mathcal{O}$ if $\mathcal{O} \setminus \mathcal{O}_m \in \mathbf{O}(\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_m))$. $\Diamond$

Remark 41 Note that for every safety class $\mathbf{O}(\cdot)$, ontology $\mathcal{O}$ and signature $\mathbf{S}$, there exists at least one $\mathbf{O}(\cdot)$-based $\mathbf{S}$-module in $\mathcal{O}$, namely $\mathcal{O}$ itself; indeed, by Definition 24, the empty ontology $\emptyset = \mathcal{O} \setminus \mathcal{O}$ also belongs to $\mathbf{O}(\mathbf{S})$ for every $\mathbf{O}(\cdot)$ and $\mathbf{S}$.

Note also that it follows from Definition 40 that $\mathcal{O}_m$ is an $\mathbf{O}(\cdot)$-based $\mathbf{S}$-module in $\mathcal{O}$ iff $\mathcal{O}_m$ is an $\mathbf{O}(\cdot)$-based $\mathbf{S}'$-module in $\mathcal{O}$ for every $\mathbf{S}'$ with $\mathbf{S} \subseteq \mathbf{S}' \subseteq (\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_m))$. $\Diamond$

It is clear that, according to Definition 40, any procedure for checking membership of a safety class $\mathbf{O}(\cdot)$ can be used directly for checking whether $\mathcal{O}_m$ is a module based on $\mathbf{O}(\cdot)$. In order to extract an $\mathbf{O}(\cdot)$-module, it is sufficient to enumerate all possible subsets of the ontology and check if any of these subsets is a module based on $\mathbf{O}(\cdot)$.

In practice, however, it is possible to avoid checking all the possible subsets of the input ontology. Figure 4 presents an optimized version of the module-extraction algorithm. The procedure manipulates configurations of the form $\mathcal{O}_m \mid \mathcal{O}_u \mid \mathcal{O}_s$, which represent a partitioning of the ontology $\mathcal{O}$ into three disjoint subsets $\mathcal{O}_m$, $\mathcal{O}_u$ and $\mathcal{O}_s$. The set $\mathcal{O}_m$ accumulates the axioms of the extracted module; the set $\mathcal{O}_s$ is intended to be safe w.r.t. $\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_m)$. The set $\mathcal{O}_u$, which is initialized to $\mathcal{O}$, contains the unprocessed axioms. These axioms are distributed among $\mathcal{O}_m$ and $\mathcal{O}_s$ according to rules R1 and R2. Given an axiom α from $\mathcal{O}_u$, rule R1 moves α into $\mathcal{O}_s$ provided $\mathcal{O}_s$ remains safe w.r.t. $\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_m)$ according to the safety class $\mathbf{O}(\cdot)$. Otherwise, rule R2 moves α into $\mathcal{O}_m$ and moves all axioms from $\mathcal{O}_s$ back into $\mathcal{O}_u$, since $\mathsf{Sig}(\mathcal{O}_m)$ might expand and axioms from $\mathcal{O}_s$ might become no longer safe w.r.t. $\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_m)$. At the end of this process, when no axioms are left in in $\mathcal{O}_u$, the set $\mathcal{O}_m$ is an $\mathbf{O}(\cdot)$-based module in $\mathcal{O}$.

The rewrite rules R1 and R2 preserve invariants I1–I3 given in Figure 4. Invariant I1 states that the three sets $\mathcal{O}_m$, $\mathcal{O}_u$ and $\mathcal{O}_s$ form a partitioning of $\mathcal{O}$; I2 states that the set $\mathcal{O}_s$ satisfies the safety test for $\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_m)$ w.r.t. $\mathbf{O}(\cdot)$; finally, I3 establishes that the rewrite rules either add elements to $\mathcal{O}_m$, or they add elements to $\mathcal{O}_s$ without changing $\mathcal{O}_m$; in other words, the pair $(|\mathcal{O}_m|, |\mathcal{O}_s|)$ consisting of the sizes for these sets increases in lexicographical order.

Input:	an ontology $\mathcal{O}$, a signature $\mathbf{S}$, a safety class $\mathbf{O}(\cdot)$
Output:	a module $\mathcal{O}_m$ in $\mathcal{O}$ based on $\mathbf{O}(\cdot)$

$$\text{Configuration: } \underset{\substack{\uparrow \\ \textbf{m}\text{odule}}}{\mathcal{O}_m} \mid \overset{\substack{\textbf{u}\text{nprocessed} \\ \downarrow}}{\mathcal{O}_u} \mid \underset{\substack{\uparrow \\ \textbf{s}\text{afe}}}{\mathcal{O}_s};$$

Initial Configuration $= \quad \emptyset \mid \mathcal{O} \mid \emptyset$

Termination Condition: $\mathcal{O}_u = \emptyset$

Rewrite rules:

R1. $\mathcal{O}_m \mid \mathcal{O}_u \cup \{\alpha\} \mid \mathcal{O}_s \Longrightarrow \mathcal{O}_m \mid \mathcal{O}_u \mid \mathcal{O}_s \cup \{\alpha\}$ if $(\mathcal{O}_s \cup \{\alpha\}) \in \mathbf{O}(\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_m))$

R2. $\mathcal{O}_m \mid \mathcal{O}_u \cup \{\alpha\} \mid \mathcal{O}_s \Longrightarrow \mathcal{O}_m \cup \{\alpha\} \mid \mathcal{O}_u \cup \mathcal{O}_s \mid \emptyset$ if $(\mathcal{O}_s \cup \{\alpha\}) \notin \mathbf{O}(\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_m))$

Invariants for $\mathcal{O}_m \mid \mathcal{O}_u \mid \mathcal{O}_s$:

I1. $\mathcal{O} = \mathcal{O}_m \uplus \mathcal{O}_u \uplus \mathcal{O}_s$

I2. $\mathcal{O}_s \in \mathbf{O}(\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_m))$

Invariant for $\mathcal{O}_m \mid \mathcal{O}_u \mid \mathcal{O}_s \Longrightarrow \mathcal{O}'_m \mid \mathcal{O}'_u \mid \mathcal{O}'_s$:

I3. $(|\mathcal{O}_m|, |\mathcal{O}_s|) <_{lex} (|\mathcal{O}'_m|, |\mathcal{O}'_s|)$

Figure 4: A Procedure for Computing Modules Based on a Safety Class $\mathbf{O}(\cdot)$

Proposition 42 (Correctness of the Procedure from Figure 4) *Let $\mathbf{O}(\cdot)$ be a safety class for an ontology language $\mathcal{L}$, $\mathcal{O}$ an ontology over $\mathcal{L}$, and $\mathbf{S}$ a signature over $\mathcal{L}$. Then:*

(1) The procedure in Figure 4 with input $\mathcal{O}$, $\mathbf{S}$, $\mathbf{O}(\cdot)$ terminates and returns an $\mathbf{O}(\cdot)$-based $\mathbf{S}$-module $\mathcal{O}_m$ in $\mathcal{O}$; and

(2) If, additionally, $\mathbf{O}(\cdot)$ is anti-monotonic, subset-closed and union-closed, then there is a unique minimal $\mathbf{O}(\cdot)$-based $\mathbf{S}$-module in $\mathcal{O}$, and the procedure returns precisely this minimal module.

Proof. (1) The procedure based on the rewrite rules from Figure 4 always terminates for the following reasons: (i) for every configuration derived by the rewrite rules, the sets $\mathcal{O}_m$, $\mathcal{O}_u$ and $\mathcal{O}_s$ form a partitioning of $\mathcal{O}$ (see invariant I1 in Figure 4), and therefore the size of every set is bounded; (ii) at each rewrite step, $(|\mathcal{O}_m|, |\mathcal{O}_s|)$ increases in lexicographical order (see invariant I3 in Figure 4). Additionally, if $\mathcal{O}_u \neq \emptyset$ then it is always possible to apply one of the rewrite rules R1 or R2, and hence the procedure always terminates with $\mathcal{O}_u = \emptyset$. Upon termination, by invariant I1 from Figure 4, $\mathcal{O}$ is partitioned into $\mathcal{O}_m$ and $\mathcal{O}_s$ and by invariant I2, $\mathcal{O}_s \in \mathbf{O}(\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_m))$, which implies, by Definition 40, that $\mathcal{O}_m$ is an $\mathbf{O}(\cdot)$-based $\mathbf{S}$-module in $\mathcal{O}$.

(2) Now, suppose that, in addition, $\mathbf{O}(\cdot)$ is an anti-monotonic, subset-closed, and union-closed safety class, and suppose that $\mathcal{O}'_m$ is an $\mathbf{O}(\cdot)$-based $\mathbf{S}$-module in $\mathcal{O}$. We demonstrate by induction that for every configuration $\mathcal{O}_m \mid \mathcal{O}_u \mid \mathcal{O}_s$ derivable from $\emptyset \mid \mathcal{O} \mid \emptyset$ by rewrite rules R1 and R2, it is the case that $\mathcal{O}_m \subseteq \mathcal{O}'_m$. This will prove that the module computed by the procedure is a subset of every $\mathbf{O}(\cdot)$-based $\mathbf{S}$-module in $\mathcal{O}$, and hence, is the smallest $\mathbf{O}(\cdot)$-based $\mathbf{S}$-module in $\mathcal{O}$.

Indeed, for the base case we have $\mathcal{O}_m = \emptyset \subseteq \mathcal{O}'_m$. The rewrite rule R1 does not change the set $\mathcal{O}_m$. For the rewrite rule R2 we have: $\mathcal{O}_m \mid \mathcal{O}_u \cup \{\alpha\} \mid \mathcal{O}_s \Longrightarrow \mathcal{O}_m \cup \{\alpha\} \mid \mathcal{O}_u \cup \mathcal{O}_s \mid \emptyset$ if $(\mathcal{O}_s \cup \{\alpha\}) \notin \mathbf{O}(\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_m))$. $\quad (\sharp)$

Input: an ontology $\mathcal{O}$, a signature $\mathbf{S}$, a safety class $\mathbf{O}(\cdot)$	
Output: a module $\mathcal{O}_m$ in $\mathcal{O}$ based on $\mathbf{O}(\cdot)$	
Initial Configuration = $\emptyset \mid \mathcal{O} \mid \emptyset$	Termination Condition: $\mathcal{O}_u = \emptyset$
Rewrite rules:	
R1. $\mathcal{O}_m \mid \mathcal{O}_u \cup \{\alpha\} \mid \mathcal{O}_s \Longrightarrow \mathcal{O}_m \mid \mathcal{O}_u \mid \mathcal{O}_s \cup \{\alpha\}$	if $\{\alpha\} \in \mathbf{O}(\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_m))$
R2. $\mathcal{O}_m \mid \mathcal{O}_u \cup \{\alpha\} \mid \mathcal{O}_s^\alpha \cup \mathcal{O}_s \Longrightarrow \mathcal{O}_m \cup \{\alpha\} \mid \mathcal{O}_u \cup \mathcal{O}_s^\alpha \mid \mathcal{O}_s$	if $\{\alpha\} \notin \mathbf{O}(\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_m))$, and $\mathsf{Sig}(\mathcal{O}_s) \cap \mathsf{Sig}(\alpha) \subseteq \mathsf{Sig}(\mathcal{O}_m)$

Figure 5: An Optimized Procedure for Computing Modules Based on a Compact Subset-Closed Union-Closed Safety Class $\mathbf{O}(\cdot)$

Suppose, to the contrary, that $\mathcal{O}_m \subseteq \mathcal{O}'_m$ but $\mathcal{O}_m \cup \{\alpha\} \nsubseteq \mathcal{O}'_m$. Then $\alpha \in \mathcal{O}'_s := \mathcal{O} \setminus \mathcal{O}'_m$. Since $\mathcal{O}'_m$ is an $\mathbf{O}(\cdot)$-based $\mathbf{S}$-module in $\mathcal{O}$, we have that $\mathcal{O}'_s \in \mathbf{O}(\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}'_m))$. Since $\mathbf{O}(\cdot)$ is subset-closed, $\{\alpha\} \in \mathbf{O}(\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}'_m))$. Since $\mathcal{O}_m \subseteq \mathcal{O}'_m$ and $\mathbf{O}(\cdot)$ is anti-monotonic, we have $\{\alpha\} \in \mathbf{O}(\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_m))$. Since by invariant I2 from Figure 4, $\mathcal{O}_s \in \mathbf{O}(\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_m))$ and $\mathbf{O}(\cdot)$ is union-closed, $\mathcal{O}_s \cup \{\alpha\} \in \mathbf{O}(\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_m))$, which contradicts ($\sharp$). This contradiction implies that the rule R2 also preserves property $\mathcal{O}_m \subseteq \mathcal{O}'_m$. $\square$

Claim (1) of Proposition 42 establishes that the procedure from Figure 4 terminates for every input and produces a module based on a given safety class. Moreover, it is possible to show that this procedure runs in polynomial time assuming that the safety test can be also performed in polynomial time.

If the safety class $\mathbf{O}(\cdot)$ satisfies additional desirable properties, like those based on classes of local interpretations described in Section 5.2, the procedure, in fact, produces the smallest possible module based on the safety class, as stated in claim (2) of Proposition 42. In this case, it is possible to optimize the procedure shown in Figure 4. If $\mathbf{O}(\cdot)$ is union closed, then, instead of checking whether $(\mathcal{O}_s \cup \{\alpha\}) \in \mathbf{O}(\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_m))$ in the conditions of rules R1 and R2, it is sufficient to check if $\{\alpha\} \in \mathbf{O}(\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_m))$ since it is already known that $\mathcal{O}_s \in \mathbf{O}(\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_m))$. If $\mathbf{O}(\cdot)$ is compact and subset closed, then instead of moving all the axioms in $\mathcal{O}_s$ to $\mathcal{O}_u$ in rule R2, it is sufficient to move only those axioms $\mathcal{O}_s^\alpha$ that contain at least one symbol from α which did not occur in $\mathcal{O}_m$ before, since the set of remaining axioms will stay in $\mathbf{O}(\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_m))$. In Figure 5 we present an optimized version of the algorithm from Figure 4 for such locality classes.

Example 43 In Table 5 we present a trace of the algorithm from Figure 5 for the ontology $\mathcal{O}$ consisting of axioms M1–M5 from Figure 1, signature $\mathbf{S} = \{\mathsf{Cystic\_Fibrosis}, \mathsf{Genetic\_Disorder}\}$ and safety class $\mathbf{O}(\cdot) = \mathbf{O}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\cdot)$ defined in Example 26. The first column of the table lists the configurations obtained from the initial configuration $\emptyset \mid \mathcal{O} \mid \emptyset$ by applying the rewrite rules R1 and R2 from Figure 5; in each row, the underlined axiom α is the one that is being tested for safety. The second column of the table shows the elements of $\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_m)$ that have appeared for the current configuration but have not been present for the preceding configurations. In the last column we indicate whether the first conditions of the rules R1 and R2 are fulfilled for the selected axiom α from $\mathcal{O}_u$—that is, whether α is local for $\mathbf{I}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\cdot)$. The rewrite rule corresponding to the result of this test is applied to the configuration.

	$\mathcal{O}_m \mid \mathcal{O}_u, \underline{\alpha} \mid \mathcal{O}_s$	New elements in $\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_m)$	$\{\alpha\} \in \mathbf{O}(\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_m))$?
1	$- \mid$ M1, $\underline{\text{M2}}$, M3, M4, M5 $\mid -$	Cystic_Fibrosis, Genetic_Disorder	Yes $\Rightarrow$ R1
2	$- \mid$ M1, $\underline{\text{M3}}$, M4, M5 $\mid$ M2	–	Yes $\Rightarrow$ R1
3	$- \mid$ $\underline{\text{M1}}$, M4, M5 $\mid$ M2, M3	–	No $\Rightarrow$ R2
4	M1 $\mid$ M2, $\underline{\text{M3}}$, M4, M5 $\mid -$	Fibrosis, located_In, Pancreas, has_Origin, Genetic_Origin	No $\Rightarrow$ R2
5	M1, M3 $\mid$ M2, $\underline{\text{M4}}$, M5 $\mid -$	Genetic_Fibrosis	No $\Rightarrow$ R2
6	M1, M3, M4 $\mid$ M2, $\underline{\text{M5}}$ $\mid -$	–	Yes $\Rightarrow$ R1
7	M1, M3, M4 $\mid$ $\underline{\text{M2}}$ $\mid$ M5	–	No $\Rightarrow$ R2
8	M1, M2, M3, M4 $\mid - \mid$ M5	–	

Table 5: A trace of the Procedure in Figure 5 for the input $\mathcal{Q} = \{\text{M1}, \ldots, \text{M5}\}$ from Figure 1 and $\mathbf{S} = \{$Cystic_Fibrosis, Genetic_Disorder$\}$

Note that some axioms α are tested for safety several times for different configurations, because the set $\mathcal{O}_u$ may increase after applications of rule R2; for example, the axiom $\alpha =$ M2 is tested for safety in both configurations 1 and 7, and $\alpha =$ M3 in configurations 2 and 4. Note also that different results for the locality tests are obtained in these cases: both M2 and M3 were local w.r.t. $\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_m)$ when $\mathcal{O}_m = \emptyset$, but became non-local when new axioms were added to $\mathcal{O}_m$. It is also easy to see that, in our case, syntactic locality produces the same results for the tests.

In our example, the rewrite procedure produces a module $\mathcal{O}_m$ consisting of axioms M1–M4. Note that it is possible to apply the rewrite rules for different choices of the axiom α in $\mathcal{O}_u$, which results in a different computation. In other words, the procedure from Figure 5 has implicit non-determinism. According to Claim (2) of Proposition 42 all such computations should produce the same module $\mathcal{O}_m$, which is the smallest $\mathbf{O}^{r \leftarrow \emptyset}_{A \leftarrow \emptyset}(\cdot)$-based $\mathbf{S}$-module in $\mathcal{O}$; that is, the implicit non-determinism in the procedure from Figure 5 does not have any impact on the result of the procedure. However, alternative choices for α may result in shorter computations: in our example we could have selected axiom M1 in the first configuration instead of M2 which would have led to a shorter trace consisting of configurations 1, and 4–8 only. $\Diamond$

It is worth examining the connection between the $\mathbf{S}$-modules in an ontology $\mathcal{O}$ based on a particular safety class $\mathbf{O}(\cdot)$ and the actual minimal $\mathbf{S}$-modules in $\mathcal{O}$. It turns out that any $\mathbf{O}(\cdot)$-based module $\mathcal{O}_m$ is guaranteed to cover the set of minimal modules, provided that $\mathbf{O}(\cdot)$ is anti-monotonic and subset-closed. In other words, given $\mathcal{O}$ and $\mathbf{S}$, $\mathcal{O}_m$ contains all the $\mathbf{S}$-essential axioms in $\mathcal{O}$. The following Lemma provides the main technical argument underlying this result.

Lemma 44 *Let $\mathbf{O}(\cdot)$ be an anti-monotonic subset-closed safety class for an ontology language $\mathcal{L}$, $\mathcal{O}$ an ontology, and $\mathbf{S}$ a signature over $\mathcal{L}$. Let $\mathcal{O}_1$ be an $\mathbf{S}$-module in $\mathcal{O}$ w.r.t. $\mathcal{L}$ and $\mathcal{O}_m$ an $\mathbf{O}(\cdot)$-based $\mathbf{S}$-module in $\mathcal{O}$. Then $\mathcal{O}_2 := \mathcal{O}_1 \cap \mathcal{O}_m$ is an $\mathbf{S}$-module in $\mathcal{O}$ w.r.t. $\mathcal{L}$.*

Proof. By Definition 40, since $\mathcal{O}_m$ is an $\mathbf{O}(\cdot)$-based $\mathbf{S}$-module in $\mathcal{O}$, we have that $\mathcal{O} \setminus \mathcal{O}_m \in \mathbf{O}(\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_m))$. Since $\mathcal{O}_1 \setminus \mathcal{O}_2 = \mathcal{O}_1 \setminus \mathcal{O}_m \subseteq \mathcal{O} \setminus \mathcal{O}_m$ and $\mathbf{O}(\cdot)$ is subset-closed, it is the case that $\mathcal{O}_1 \setminus \mathcal{O}_2 \in \mathbf{O}(\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_m))$. Since $\mathbf{O}(\cdot)$ is anti-monotonic, and $\mathcal{O}_2 \subseteq \mathcal{O}_m$, we have that $\mathcal{O}_1 \setminus \mathcal{O}_2 \in \mathbf{O}(\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_2))$, hence, $\mathcal{O}_2$ is an $\mathbf{O}(\cdot)$-based $\mathbf{S}$-module in $\mathcal{O}_1$. In particular $\mathcal{O}_2$ is an $\mathbf{S}$-module in $\mathcal{O}_1$ w.r.t. $\mathcal{L}$. Since $\mathcal{O}_1$ is an $\mathbf{S}$-module in $\mathcal{O}$ w.r.t. $\mathcal{L}$, $\mathcal{O}_2$ is an $\mathbf{S}$-module in $\mathcal{O}$ w.r.t. $\mathcal{L}$. □

Corollary 45 *Let $\mathbf{O}(\cdot)$ be an anti-monotonic, subset-closed safety class for $\mathcal{L}$ and $\mathcal{O}_m$ be an $\mathbf{O}(\cdot)$-based $\mathbf{S}$-module in $\mathcal{O}$. Then $\mathcal{O}_m$ contains all $\mathbf{S}$-essential axioms in $\mathcal{O}$ w.r.t. $\mathcal{L}$.*

Proof. Let $\mathcal{O}_1$ be a minimal $\mathbf{S}$-module in $\mathcal{O}$ w.r.t. $\mathcal{L}$. We demonstrate that $\mathcal{O}_1 \subseteq \mathcal{O}_m$. Indeed, otherwise, by Lemma 44, $\mathcal{O}_1 \cap \mathcal{O}_m$ is an $\mathbf{S}$-module in $\mathcal{O}$ w.r.t. $\mathcal{L}$ which is strictly contained in $\mathcal{O}_1$. Hence $\mathcal{O}_m$ is a superset of every minimal $\mathbf{S}$-module in $\mathcal{O}$ and hence, contains all $\mathbf{S}$-essential axioms in $\mathcal{O}$ w.r.t. $\mathcal{L}$. □

As shown in Section 3.4, all the axioms M1–M4 are essential for the ontology $\mathcal{O}$ and signature $\mathbf{S}$ considered in Example 43. We have seen in this example that the locality-based $\mathbf{S}$-module extracted from $\mathcal{O}$ contains all of these axioms, in accordance to Corollary 45. In our case, the extracted module contains only essential axioms; in general, however, locality-based modules might contain non-essential axioms.

An interesting application of modules is the pruning of irrelevant axioms when checking if an axiom α is implied by an ontology $\mathcal{O}$. Indeed, in order to check whether $\mathcal{O} \models \alpha$ it suffices to retrieve the module for $\mathsf{Sig}(\alpha)$ and verify if the implication holds w.r.t. this module. In some cases, it is sufficient to extract a module for a subset of the signature of α which, in general, leads to smaller modules. In particular, in order to test subsumption between a pair of atomic concepts, if the safety class being used enjoys some nice properties then it suffices to extract a module for one of them, as given by the following proposition:

Proposition 46 *Let $\mathbf{O}(\cdot)$ be a compact union-closed safety class for an ontology language $\mathcal{L}$, $\mathcal{O}$ an ontology and A, B atomic concepts. Let $\mathcal{O}_A$ be an $\mathbf{O}(\cdot)$-based module in $\mathcal{O}$ for $\mathbf{S} = \{A\}$ in $\mathcal{O}$. Then:*

1 *If $\mathcal{O} \models \alpha := (A \sqsubseteq B)$ and $\{B \sqsubseteq \bot\} \in \mathbf{O}(\emptyset)$ then $\mathcal{O}_A \models \alpha$;*

2 *If $\mathcal{O} \models \alpha := (B \sqsubseteq A)$ and $\{\top \sqsubseteq B\} \in \mathbf{O}(\emptyset)$ then $\mathcal{O}_A \models \alpha$;*

Proof. 1. Consider two cases: (a) $B \in \mathsf{Sig}(\mathcal{O}_A)$ and (b) $B \notin \mathsf{Sig}(\mathcal{O}_A)$.

(a) By Remark 41 it is the case that $\mathcal{O}_A$ is an $\mathbf{O}(\cdot)$-based module in $\mathcal{O}$ for $\mathbf{S} = \{A, B\}$. Since $\mathsf{Sig}(\alpha) \subseteq \mathbf{S}$, by Definition 12 and Definition 1, it is the case that $\mathcal{O} \models \alpha$ implies $\mathcal{O}_A \models \alpha$.

(b) Consider $\mathcal{O}' = \mathcal{O} \cup \{B \sqsubseteq \bot\}$. Since $\mathsf{Sig}(B \sqsubseteq \bot) = \{B\}$ and $B \notin \mathsf{Sig}(\mathcal{O}_A)$, $\{B \sqsubseteq \bot\} \in \mathbf{O}(\emptyset)$ and $\mathbf{O}(\cdot)$ is compact, by Definition 24 it is the case that $\{B \sqsubseteq \bot\} \in \mathbf{O}(\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_A))$. Since $\mathcal{O}_A$ is an $\mathbf{O}(\cdot)$-based $\mathbf{S}$-module in $\mathcal{O}$, by Definition 40, then $\mathcal{O} \setminus \mathcal{O}_A \in \mathbf{O}(\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_A))$. Since $\mathbf{O}(\cdot)$ is union-closed, it is the case that $(\mathcal{O} \setminus \mathcal{O}_A) \cup \{B \sqsubseteq \bot\} \in \mathbf{O}(\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_A))$. Note that $B \sqsubseteq \bot \notin \mathcal{O}_A$, since $B \notin \mathsf{Sig}(\mathcal{O}_A)$, hence $(\mathcal{O} \setminus \mathcal{O}_A) \cup \{B \sqsubseteq \bot\} = \mathcal{O}' \setminus \mathcal{O}_A$, and, by Definition 40, we have that $\mathcal{O}_A$ is an $\mathbf{O}(\cdot)$-based $\mathbf{S}$-module in $\mathcal{O}'$. Now, since $\mathcal{O} \models (A \sqsubseteq B)$, it is the case that $\mathcal{O}' \models (A \sqsubseteq \bot)$, and hence, since $\mathcal{O}_A$ is a module in $\mathcal{O}'$ for $\mathbf{S} = \{A\}$, we have that $\mathcal{O}_A \models (A \sqsubseteq \bot)$ which implies $\mathcal{O}_A \models (A \sqsubseteq B)$.

2. The proof of this case is analogous to that for Case 1: Case (a) is applicable without changes; in Case (b) we show that $\mathcal{O}_A$ is an $\mathbf{O}(\cdot)$-based module for $\mathbf{S} = \{A\}$ in $\mathcal{O}' = \mathcal{O} \cup \{\top \sqsubseteq B\}$, and, hence, since $\mathcal{O}' \models (\top \sqsubseteq A)$, it is the case that $\mathcal{O}_B \models (\top \sqsubseteq A)$, which implies $\mathcal{O}' \models (B \sqsubseteq A)$. $\square$

Corollary 47 *Let $\mathcal{O}$ be a $\mathcal{SHOIQ}$ ontology and A, B atomic concepts. Let $\mathbf{O}^{r\leftarrow *}_{A\leftarrow\emptyset}(\cdot)$ and $\mathbf{O}^{r\leftarrow *}_{A\leftarrow\Delta}(\cdot)$ be locality classes based on local classes of interpretations of form $\mathbf{I}^{r\leftarrow *}_{A\leftarrow\emptyset}(\cdot)$ and $\mathbf{I}^{r\leftarrow *}_{A\leftarrow\Delta}(\cdot)$, respectively, from Table 3. Let $\mathcal{O}_A$ be a module for $\mathbf{S} = \{A\}$ based on $\mathbf{O}^{r\leftarrow *}_{A\leftarrow\emptyset}(\cdot)$ and $\mathcal{O}_B$ be a module for $\mathbf{S} = \{B\}$ based on $\mathbf{O}^{r\leftarrow *}_{A\leftarrow\Delta}(\cdot)$. Then $\mathcal{O} \models (A \sqsubseteq B)$ iff $\mathcal{O}_A \models (A \sqsubseteq B)$ iff $\mathcal{O}_B \models (A \sqsubseteq B)$.*

Proof. It is easy to see that $\{B \sqsubseteq \bot\} \in \mathbf{O}^{r\leftarrow *}_{A\leftarrow\emptyset}(\emptyset)$ and $\{\top \sqsubseteq A\} \in \mathbf{O}^{r\leftarrow *}_{A\leftarrow\Delta}(\Delta)$. $\square$

Proposition 46 implies that a module based on a safety class for a single atomic concept A can be used for capturing either the super-concepts (Case 1), or the sub-concepts (Case 2) B of A, provided that the safety class captures, when applied to the empty signature, axioms of the form $B \sqsubseteq \bot$ (Case 1) or $(\top \sqsubseteq B)$ (Case 2). That is, B is a super-concept or a sub-concept of A in the ontology if and only if it is such in the module. This property can be used, for example, to optimize the classification of ontologies. In order to check if the subsumption $A \sqsubseteq B$ holds in an ontology $\mathcal{O}$, it is sufficient to extract a module in $\mathcal{O}$ for $\mathbf{S} = \{A\}$ using a modularization algorithm based on a safety class in which the ontology $\{B \sqsubseteq \bot\}$ is local w.r.t. the empty signature, and check whether this subsumption holds w.r.t. the module. For this purpose, it is convenient to use a syntactically tractable approximation of the safety class in use; for example, one could use the syntactic locality conditions given in Figure 3 instead of their semantic counterparts.

It is possible to combine modularization procedures to obtain modules that are smaller than the ones obtained using these procedures individually. For example, in order to check the subsumption $\mathcal{O} \models (A \sqsubseteq B)$ one could first extract a $\mathbf{O}^{r\leftarrow *}_{A\leftarrow\emptyset}(\cdot)$-based module $\mathcal{M}_1$ for $\mathbf{S} = \{A\}$ in $\mathcal{O}$; by Corollary 47 this module is complete for all the super-concepts of A in $\mathcal{O}$, including B—that is, if an atomic concept is a super-concept of A in $\mathcal{O}$, then it is also a super-concept of A in $\mathcal{M}_1$. One could extract a $\mathbf{O}^{r\leftarrow *}_{A\leftarrow\Delta}(\cdot)$-based module $\mathcal{M}_2$ for $\mathbf{S} = \{B\}$ in $\mathcal{M}_1$ which, by Corollary 47, is complete for all the sub-concepts of B in $\mathcal{M}_1$, including A. Indeed, $\mathcal{M}_2$ is an $\mathbf{S}$-module in $\mathcal{M}_1$ for $\mathbf{S} = \{A, B\}$ and $\mathcal{M}_1$ is an $\mathbf{S}$-module in the original ontology $\mathcal{O}$. By Proposition 46, therefore, it is the case that $\mathcal{M}_2$ is also an $\mathbf{S}$-module in $\mathcal{O}$.

7. Related Work

We have seen in Section 3 that the notion of conservative extension is valuable in the formalization of ontology reuse tasks. The problem of deciding conservative extensions has been recently investigated in the context of ontologies (Ghilardi et al., 2006; Lutz et al., 2007; Lutz & Wolter, 2007). The problem of deciding whether $\mathcal{P} \cup \mathcal{Q}$ is a deductive $\mathbf{S}$-conservative extension of $\mathcal{Q}$ is EXPTIME-complete for $\mathcal{EL}$ (Lutz & Wolter, 2007), 2-EXPTIME-complete w.r.t. $\mathcal{ALCIQ}$ (Lutz et al., 2007) (roughly OWL-Lite), and undecidable w.r.t. $\mathcal{ALCIQO}$ (roughly OWL DL). Furthermore, checking model conservative extensions is already undecidable for $\mathcal{EL}$ (Lutz & Wolter, 2007), and for $\mathcal{ALC}$ it is even not semi-decidable (Lutz et al., 2007).

In the last few years, a rapidly growing body of work has been developed under the headings of Ontology Mapping and Alignment, Ontology Merging, Ontology Integration, and Ontology Segmentation (Kalfoglou & Schorlemmer, 2003; Noy, 2004a, 2004b). This field is rather diverse and has roots in several communities.

In particular, numerous techniques for extracting fragments of ontologies for the purposes of knowledge reuse have been proposed. Most of these techniques rely on syntactically traversing the axioms in the ontology and employing various heuristics to determine which axioms are relevant and which are not.

An example of such a procedure is the algorithm implemented in the Prompt-Factor tool (Noy & Musen, 2003). Given a signature $\mathbf{S}$ and an ontology $\mathcal{Q}$, the algorithm retrieves a fragment $\mathcal{Q}_1 \subseteq \mathcal{Q}$ as follows: first, the axioms in $\mathcal{Q}$ that mention any of the symbols in $\mathbf{S}$ are added to $\mathcal{Q}_1$; second, $\mathbf{S}$ is expanded with the symbols in $\mathsf{Sig}(\mathcal{Q}_1)$. These steps are repeated until a fixpoint is reached. For our example in Section 3, when $\mathbf{S} = \{\mathsf{Cystic\_Fibrosis}, \mathsf{Genetic\_Disorder}\}$, and $\mathcal{Q}$ consists of axioms M1–M5 from Figure 1, the algorithm first retrieves axioms M1, M4, and M5 containing these terms, then expands $\mathbf{S}$ with the symbols mentioned in these axioms, such that $\mathbf{S}$ contains all the symbols of $\mathcal{Q}$. After this step, all the remaining axioms of $\mathcal{Q}$ are retrieved. Hence, the fragment extracted by the Prompt-Factor algorithm consists of all the axioms M1-M5. In this case, the Prompt-Factor algorithm extracts a module (though not a minimal one). In general, however, the extracted fragment is not guaranteed to be a module. For example, consider an ontology $\mathcal{Q} = \{A \equiv \neg A,\ B \sqsubseteq C\}$ and $\alpha = (C \sqsubseteq B)$. The ontology $\mathcal{Q}$ is inconsistent due to the axiom $A \equiv \neg A$: any axiom (and α in particular) is thus a logical consequence of $\mathcal{Q}$. Given $\mathbf{S} = \{B, C\}$, the Prompt-Factor algorithm extracts $\mathcal{Q}_2 = \{B \sqsubseteq C\}$; however, $\mathcal{Q}_2 \not\models \alpha$, and so $\mathcal{Q}_2$ is not a module in $\mathcal{Q}$. In general, the Prompt-Factor algorithm may fail even if $\mathcal{Q}$ is consistent. For example, consider an ontology $\mathcal{Q} = \{\top \sqsubseteq \{a\},\ A \sqsubseteq B\}$, $\alpha = (A \sqsubseteq \forall r.A)$, and $\mathbf{S} = \{A\}$. It is easy to see that $\mathcal{Q}$ is consistent, admits only single element models, and α is satisfied in every such a model; that is, $\mathcal{Q} \models \alpha$. In this case, the Prompt-Factor algorithm extracts $\mathcal{Q}_1 = \{A \sqsubseteq B\}$, which does not imply α.

Another example is Seidenberg's segmentation algorithm (Seidenberg & Rector, 2006), which was used for segmentation of the medical ontology GALEN (Rector & Rogers, 1999). Currently, the full version of GALEN cannot be processed by reasoners, so the authors investigate the possibility of splitting GALEN into small "segments" which can be processed by reasoners separately. The authors describe a segmentation procedure which, given a set of atomic concepts $\mathbf{S}$, computes a "segment" for $\mathbf{S}$ in the ontology. The description of the procedure is very high-level. The authors discuss which concepts and roles should be included in the segment and which should not. In particular, the segment should contain all "super-" and "sub-" concepts of the input concepts, concepts that are "linked" from the input concepts (via existential restrictions) and their "super-concepts", but not their "sub-concepts"; for the included concepts, also "their restrictions", "intersection", "union", and "equivalent concepts" should be considered by including the roles and concepts they contain, together with their "super-concepts" and "super-roles" but not their "sub-concepts" and "sub-roles". From the description of the procedure it is not entirely clear whether it works with a classified ontology (which is unlikely in the case of GALEN since the full version of GALEN has not been classified by any existing reasoner), or, otherwise, how the "super-" and "sub-" concepts are computed. It is also not clear which axioms should be included in the segment in the end, since the procedure talks only about the inclusion of concepts and roles.

A different approach to module extraction proposed in the literature (Stuckenschmidt & Klein, 2004) consists of partitioning the concepts in an ontology to facilitate visualization of and navigation through an ontology. The algorithm uses a set of heuristics for measuring the degree of dependency between the concepts in the ontology and outputs a graphical representation of these dependencies. The algorithm is intended as a visualization technique, and does not establish a correspondence between the nodes of the graph and sets of axioms in the ontology.

What is common between the modularization procedures we have mentioned is the lack of a formal treatment for the notion of module. The papers describing these modularization procedures do not attempt to formally specify the intended outputs of the procedures, but rather argue what should be in the modules and what not based on intuitive notions. In particular, they do not take the semantics of the ontology languages into account. It might be possible to formalize these algorithms and identify ontologies for which the "intuition-based" modularization procedures work correctly. Such studies are beyond the scope of this paper.

Module extraction in ontologies has also been investigated from a formal point of view (Cuenca Grau et al., 2006b). Cuenca Grau et al. (2006) define a notion of a module $\mathcal{Q}_A$ in an ontology $\mathcal{Q}$ for an atomic concept A. One of the requirements for the module is that $\mathcal{Q}$ should be a conservative extension of $\mathcal{Q}_A$ (in the paper $\mathcal{Q}_A$ is called a logical module in $\mathcal{Q}$). The paper imposes an additional requirement on modules, namely that the module $\mathcal{Q}_A$ should entail all the subsumptions in the original ontology between atomic concepts involving A and other atomic concepts in $\mathcal{Q}_A$. The authors present an algorithm for partitioning an ontology into disjoint modules and proved that the algorithm is correct provided that certain safety requirements for the input ontology hold: the ontology should be consistent, should not contain unsatisfiable atomic concepts, and should have only "safe" axioms (which in our terms means that they are local for the empty signature). In contrast, the algorithm we present here works for any ontology, including those containing "non-safe" axioms.

The growing interest in the notion of "modularity" in ontologies has been recently reflected in a workshop on modular ontologies[5] held in conjunction with the International Semantic Web Conference (ISWC-2006). Concerning the problem of ontology reuse, there have been various proposals for "safely" combining modules; most of these proposals, such as $\mathcal{E}$-connections (Cuenca Grau, Parsia, & Sirin, 2006a), Distributed Description Logics (Borgida & Serafini, 2003) and Package-based Description Logics (Bao, Caragea, & Honavar, 2006) propose a specialized semantics for controlling the interaction between the importing and the imported modules to avoid side-effects. In contrast to these works, we assume here that reuse is performed by simply building the logical union of the axioms in the modules under the standard semantics, and we establish a collection of reasoning services, such as safety testing, to check for side-effects. The interested reader can find in the literature a detailed comparison between the different approaches for combining ontologies (Cuenca Grau & Kutz, 2007).

5. for information see the homepage of the workshop `http://www.cild.iastate.edu/events/womo.html`

8. Implementation and Proof of Concept

In this section, we provide empirical evidence of the appropriateness of locality for safety testing and module extraction. For this purpose, we have implemented a syntactic locality checker for the locality classes $\tilde{\mathbf{O}}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\cdot)$ and $\tilde{\mathbf{O}}^{r\leftarrow\Delta\times\Delta}_{A\leftarrow\Delta}(\cdot)$ as well as the algorithm for extracting modules given in Figure 5 from Section 6.

First, we show that the locality class $\tilde{\mathbf{O}}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\cdot)$ provides a powerful sufficiency test for safety which works for many real-world ontologies. Second, we show that $\tilde{\mathbf{O}}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\cdot)$-based modules are typically very small compared to both the size of the ontology and the modules extracted using other techniques. Third, we report on our implementation in the ontology editor Swoop (Kalyanpur, Parsia, Sirin, Cuenca Grau, & Hendler, 2006) and illustrate the combination of the modularization procedures based on the classes $\tilde{\mathbf{O}}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\cdot)$ and $\tilde{\mathbf{O}}^{r\leftarrow\Delta\times\Delta}_{A\leftarrow\Delta}(\cdot)$.

8.1 Locality for Testing Safety

We have run our syntactic locality checker for the class $\tilde{\mathbf{O}}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\cdot)$ over the ontologies from a library of 300 ontologies of various sizes and complexity some of which import each other (Gardiner, Tsarkov, & Horrocks, 2006).[6] For all ontologies $\mathcal{P}$ that import an ontology $\mathcal{Q}$, we check if $\mathcal{P}$ belongs to the locality class for $\mathbf{S} = \mathsf{Sig}(\mathcal{P}) \cap \mathsf{Sig}(\mathcal{Q})$.

It turned out that from the 96 ontologies in the library that import other ontologies, all but 11 were syntactically local for $\mathbf{S}$ (and hence also semantically local for $\mathbf{S}$). From the 11 non-local ontologies, 7 are written in the OWL-Full species of OWL (Patel-Schneider et al., 2004) to which our framework does not yet apply. The remaining 4 non-localities are due to the presence of so-called *mapping axioms* of the form $A \equiv B'$, where $A \notin \mathbf{S}$ and $B' \in \mathbf{S}$. Note that these axioms simply indicate that the atomic concepts A, B' in the two ontologies under consideration are synonyms. Indeed, we were able to easily repair these non-localities as follows: we replace every occurrence of A in $\mathcal{P}$ with B' and then remove this axiom from the ontology. After this transformation, all 4 non-local ontologies turned out to be local.

8.2 Extraction of Modules

In this section, we compare three modularization[7] algorithms that we have implemented using Manchester's OWL API:[8]

A1: The Prompt-Factor algorithm (Noy & Musen, 2003);

A2: The segmentation algorithm proposed by Cuenca Grau et al. (2006);

A3: Our modularisation algorithm (Algorithm 5), based on the locality class $\tilde{\mathbf{O}}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\cdot)$.

The aim of the experiments described in this section is not to provide a throughout comparison of the quality of existing modularization algorithms since each algorithm extracts "modules" according to its own requirements, but rather to give an idea of the typical size of the modules extracted from real ontologies by each of the algorithms.

6. The library is available at `http://www.cs.man.ac.uk/~horrocks/testing/`
7. In this section by "module" we understand the result of the considered modularization procedures which may not necessarily be a module according to Definition 10 or 12
8. `http://sourceforge.net/projects/owlapi`

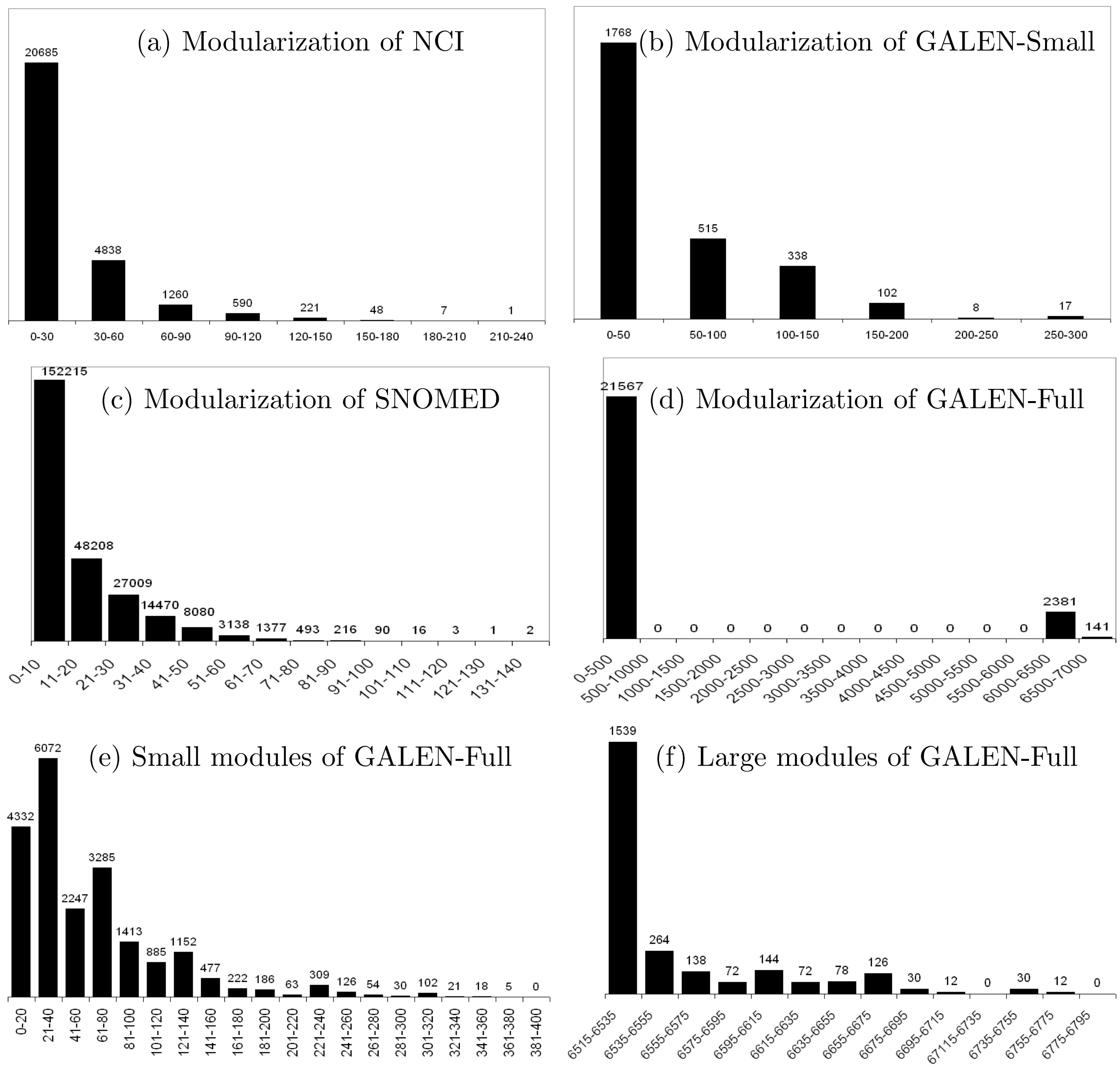


Figure 6: Distribution for the sizes of syntactic locality-based modules: the X-Axis gives the number of concepts in the modules and the Y-Axis the number of modules extracted for each size range.

Ontology	♯ Atomic	A1: Prompt-Factor		A2: Segmentation		A3: Loc.-based mod.	
	Concepts	Max.(%)	Avg.(%)	Max.(%)	Avg.(%)	Max.(%)	Avg.(%)
NCI	27772	87.6	75.84	55	30.8	0.8	0.08
SNOMED	255318	100	100	100	100	0.5	0.05
GO	22357	1	0.1	1	0.1	0.4	0.05
SUMO	869	100	100	100	100	2	0.09
GALEN-Small	2749	100	100	100	100	10	1.7
GALEN-Full	24089	100	100	100	100	29.8	3.5
SWEET	1816	96.4	88.7	83.3	51.5	1.9	0.1
DOLCE-Lite	499	100	100	100	100	37.3	24.6

Table 6: Comparison of Different Modularization Algorithms

As a test suite, we have collected a set of well-known ontologies available on the Web, which we divided into two groups:

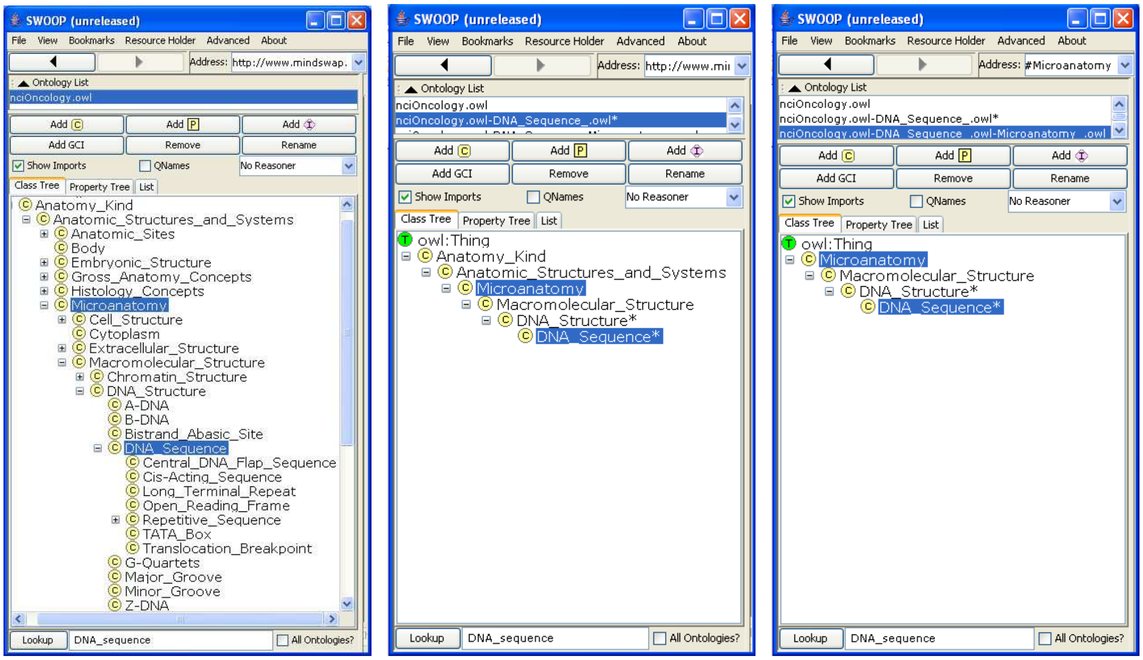
Simple. In this group, we have included the National Cancer Institute (NCI) Ontology,[9] the SUMO Upper Ontology,[10] the Gene Ontology (GO),[11] and the SNOMED Ontology[12]. These ontologies are expressed in a simple ontology language and are of a simple structure; in particular, they do not contain GCIs, but only definitions.

Complex. This group contains the well-known GALEN ontology (GALEN-Full),[13] the DOLCE upper ontology (DOLCE-Lite),[14] and NASA's Semantic Web for Earth and Environmental Terminology (SWEET)[15]. These ontologies are complex since they use many constructors from OWL DL and/or include a significant number of GCIs. In the case of GALEN, we have also considered a version GALEN-Small that has commonly been used as a benchmark for OWL reasoners. This ontology is almost 10 times smaller than the original GALEN-Full ontology, yet similar in structure.

Since there is no benchmark for ontology modularization and only a few use cases are available, there is no systematic way of evaluating modularization procedures. Therefore we have designed a simple experiment setup which, even if it may not necessarily reflect an actual ontology reuse scenario, it should give an idea of typical module sizes. For each ontology, we took the set of its atomic concepts and extracted modules for every atomic concept. We compare the maximal and average sizes of the extracted modules.

It is worth emphasizing here that our algorithm A3 does not just extract a module for the input atomic concept: the extracted fragment is also a module for its whole signature, which typically includes a fair amount of other concepts and roles.

9. `http://www.mindswap.org/2003/CancerOntology/nciOncology.owl`
10. `http://ontology.teknowledge.com/`
11. `http://www.geneontology.org`
12. `http://www.snomed.org`
13. `http://www.openclinical.org/prj_galen.html`
14. `http://www.loa-cnr.it/DOLCE.html`
15. `http://sweet.jpl.nasa.gov/ontology/`



(a) Concepts DNA_Sequence and Microanatomy in NCI

(b) $\tilde{\mathbf{O}}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\mathbf{S})$-based module for DNA_Sequence in NCI

(c) $\tilde{\mathbf{O}}^{r\leftarrow\Delta\times\Delta}_{A\leftarrow\Delta}(\mathbf{S})$-based module for Micro_Anatomy in the fragment 7b

Figure 7: The Module Extraction Functionality in Swoop

The results we have obtained are summarized in Table 6. The table provides the size of the largest module and the average size of the modules obtained using each of these algorithms. In the table, we can clearly see that locality-based modules are significantly smaller than the ones obtained using the other methods; in particular, in the case of SUMO, DOLCE, GALEN and SNOMED, the algorithms A1 and A2 retrieve the whole ontology as the module for each input signature. In contrast, the modules we obtain using our algorithm are significantly smaller than the size of the input ontology.

For NCI, SNOMED, GO and SUMO, we have obtained very small locality-based modules. This can be explained by the fact that these ontologies, even if large, are simple in structure and logical expressivity. For example, in SNOMED, the largest locality-based module obtained is approximately 0.5% of the size of the ontology, and the average size of the modules is 1/10 of the size of the largest module. In fact, most of the modules we have obtained for these ontologies contain less than 40 atomic concepts.

For GALEN, SWEET and DOLCE, the locality-based modules are larger. Indeed, the largest module in GALEN-Small is 1/10 of the size of the ontology, as opposed to 1/200 in the case of SNOMED. For DOLCE, the modules are even bigger—1/3 of the size of the ontology—which indicates that the dependencies between the different concepts in the ontology are very strong and complicated. The SWEET ontology is an exception: even though the ontology uses most of the constructors available in OWL, the ontology is heavily underspecified, which yields small modules.

In Figure 6, we have a more detailed analysis of the modules for NCI, SNOMED, GALEN-Small and GALEN-Full. Here, the X-axis represents the size ranges of the obtained modules and the Y-axis the number of modules whose size is within the given range. The plots thus give an idea of the distribution for the sizes of the different modules.

For SNOMED, NCI and GALEN-Small, we can observe that the size of the modules follows a smooth distribution. In contrast, for GALEN-Full, we have obtained a large number of small modules and a significant number of very big ones, but no medium-sized modules in-between. This abrupt distribution indicates the presence of a big cycle of dependencies in the ontology. The presence of this cycle can be spotted more clearly in Figure 6(f); the figure shows that there is a large number of modules of size in between 6515 and 6535 concepts. This cycle does not occur in the simplified version of GALEN and thus we obtain a smooth distribution for that case. In contrast, in Figure 6(e) we can see that the distribution for the "small" modules in GALEN-Full is smooth and much more similar to the one for the simplified version of GALEN.

The considerable differences in the size of the modules extracted by algorithms A1 – A3 are due to the fact that these algorithms extract modules according to different requirements. Algorithm A1 produces a fragment of the ontology that contains the input atomic concept and is syntactically separated from the rest of the axioms—that is, the fragment and the rest of the ontology have disjoint signatures. Algorithm A2 extracts a fragment of the ontology that is a module for the input atomic concept and is additionally semantically separated from the rest of the ontology: no entailment between an atomic concept in the module and an atomic concept not in the module should hold in the original ontology. Since our algorithm is based on weaker requirements, it is to be expected that it extracts smaller modules. What is surprising is that the difference in the size of modules is so significant.

In order to explore the use of our results for ontology design and analysis, we have integrated our algorithm for extracting modules in the ontology editor Swoop (Kalyanpur et al., 2006). The user interface of Swoop allows for the selection of an input signature and the retrieval of the corresponding module.[16]

Figure 7a shows the classification of the concepts DNA_Sequence and Microanatomy in the NCI ontology. Figure 7b shows the minimal $\tilde{\mathbf{O}}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\cdot)$-based module for DNA_Sequence, as obtained in Swoop. Recall that, according to Corollary 47, a $\tilde{\mathbf{O}}^{r\leftarrow\emptyset}_{A\leftarrow\emptyset}(\cdot)$-based module for an atomic concept contains all necessary axioms for, at least, all its (entailed) super-concepts in $\mathcal{O}$. Thus this module can be seen as the "upper ontology" for A in $\mathcal{O}$. In fact, Figure 7 shows that this module contains only the concepts in the "path" from DNA_Sequence to the top level concept Anatomy_Kind. This suggests that the knowledge in NCI about the particular concept DNA_Sequence is very shallow in the sense that NCI only "knows" that a DNA_Sequence is a macromolecular structure, which, in the end, is an anatomy kind. If one wants to refine the module by only including the information in the ontology necessary to entail the "path" from DNA_Sequence to Micro_Anatomy, one could extract the $\tilde{\mathbf{O}}^{r\leftarrow\Delta\times\Delta}_{A\leftarrow\Delta}(\cdot)$-based module for Micro_Anatomy in the fragment 7b. By Corollary 47, this module contains all the sub-concepts of Micro_Anatomy in the previously extracted module. The resulting module is shown in Figure 7b.

16. The tool can be downloaded at `http://code.google.com/p/swoop/`

9. Conclusion

In this paper, we have proposed a set of reasoning problems that are relevant for ontology reuse. We have established the relationships between these problems and studied their computability. Using existing results (Lutz et al., 2007) and the results obtained in Section 4, we have shown that these problems are undecidable or algorithmically unsolvable for the logic underlying OWL DL. We have dealt with these problems by defining sufficient conditions for a solution to exist, which can be computed in practice. We have introduced and studied the notion of a safety class, which characterizes any sufficiency condition for safety of an ontology w.r.t. a signature. In addition, we have used safety classes to extract modules from ontologies.

For future work, we would like to study other approximations which can produce small modules in complex ontologies like GALEN, and exploit modules to optimize ontology reasoning.

References

Baader, F., Brandt, S., & Lutz, C. (2005). Pushing the $\mathcal{EL}$ envelope. In *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30-August 5, 2005*, pp. 364–370. Professional Book Center.

Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., & Patel-Schneider, P. F. (Eds.). (2003). *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.

Bao, J., Caragea, D., & Honavar, V. (2006). On the semantics of linking and importing in modular ontologies. In *Proceedings of the 5th International Semantic Web Conference (ISWC-2006), Athens, GA, USA, November 5-9, 2006*, Vol. 4273 of *Lecture Notes in Computer Science*, pp. 72–86.

Börger, E., Grädel, E., & Gurevich, Y. (1997). *The Classical Decision Problem*. Perspectives of Mathematical Logic. Springer-Verlag. Second printing (Universitext) 2001.

Borgida, A., & Serafini, L. (2003). Distributed description logics: Assimilating information from peer sources. *J. Data Semantics*, *1*, 153–184.

Cuenca Grau, B., Horrocks, I., Kazakov, Y., & Sattler, U. (2007). A logical framework for modularity of ontologies. In *IJCAI-07, Proceedings of the Twentieth International Joint Conference on Artificial Intelligence, Hyderabad, India, January 2007*, pp. 298–304. AAAI.

Cuenca Grau, B., & Kutz, O. (2007). Modular ontology languages revisited. In *Proceedings of the Workshop on Semantic Web for Collaborative Knowledge Acquisition, Hyderabad, India, January 5, 2007*.

Cuenca Grau, B., Parsia, B., & Sirin, E. (2006a). Combining OWL ontologies using $\mathcal{E}$-connections. *J. Web Sem.*, *4*(1), 40–59.

Cuenca Grau, B., Parsia, B., Sirin, E., & Kalyanpur, A. (2006b). Modularity and web ontologies. In *Proceedings of the Tenth International Conference on Principles of Knowledge

Representation and Reasoning (KR-2006), Lake District of the United Kingdom, June 2-5, 2006, pp. 198–209. AAAI Press.

Cuenca Grau, B., Horrocks, I., Kazakov, Y., & Sattler, U. (2007). Just the right amount: extracting modules from ontologies. In *Proceedings of the 16th International Conference on World Wide Web (WWW-2007), Banff, Alberta, Canada, May 8-12, 2007*, pp. 717–726. ACM.

Cuenca Grau, B., Horrocks, I., Kutz, O., & Sattler, U. (2006). Will my ontologies fit together?. In *Proceedings of the 2006 International Workshop on Description Logics (DL-2006), Windermere, Lake District, UK, May 30 - June 1, 2006*, Vol. 189 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Gardiner, T., Tsarkov, D., & Horrocks, I. (2006). Framework for an automated comparison of description logic reasoners. In *Proceedings of the 5th International Semantic Web Conference (ISWC-2006), Athens, GA, USA, November 5-9, 2006*, Vol. 4273 of *Lecture Notes in Computer Science*, pp. 654–667. Springer.

Ghilardi, S., Lutz, C., & Wolter, F. (2006). Did I damage my ontology? a case for conservative extensions in description logics. In *Proceedings of the Tenth International Conference on Principles of Knowledge Representation and Reasoning (KR-2006), Lake District of the United Kingdom, June 2-5, 2006*, pp. 187–197. AAAI Press.

Horrocks, I., Patel-Schneider, P. F., & van Harmelen, F. (2003). From SHIQ and RDF to OWL: the making of a web ontology language. *J. Web Sem.*, *1*(1), 7–26.

Horrocks, I., & Sattler, U. (2005). A tableaux decision procedure for $\mathcal{SHOIQ}$. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05), Edinburgh, Scotland, UK, July 30-August 5, 2005*, pp. 448–453. Professional Book Center.

Kalfoglou, Y., & Schorlemmer, M. (2003). Ontology mapping: The state of the art. *The Knowledge Engineering Review*, *18*, 1–31.

Kalyanpur, A., Parsia, B., Sirin, E., Cuenca Grau, B., & Hendler, J. A. (2006). Swoop: A web ontology editing browser. *J. Web Sem.*, *4*(2), 144–153.

Lutz, C., Walther, D., & Wolter, F. (2007). Conservative extensions in expressive description logics. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07), Hyderabad, India, January 2007*, pp. 453–459. AAAI.

Lutz, C., & Wolter, F. (2007). Conservative extensions in the lightweight description logic $\mathcal{EL}$. In *Proceedings of the 21st International Conference on Automated Deduction (CADE-21), Bremen, Germany, July 17-20, 2007*, Vol. 4603 of *Lecture Notes in Computer Science*, pp. 84–99. Springer.

Möller, R., & Haarslev, V. (2003). Description logic systems. In *The Description Logic Handbook*, chap. 8, pp. 282–305. Cambridge University Press.

Motik, B. (2006). *Reasoning in Description Logics using Resolution and Deductive Databases*. Ph.D. thesis, Univesität Karlsruhe (TH), Karlsruhe, Germany.

Noy, N. F. (2004a). Semantic integration: A survey of ontology-based approaches. *SIGMOD Record*, *33*(4), 65–70.

Noy, N. F. (2004b). Tools for mapping and merging ontologies. In Staab, & Studer (Staab & Studer, 2004), pp. 365–384.

Noy, N., & Musen, M. (2003). The PROMPT suite: Interactive tools for ontology mapping and merging. *Int. Journal of Human-Computer Studies, Elsevier*, *6*(59).

Patel-Schneider, P., Hayes, P., & Horrocks, I. (2004). Web ontology language OWL Abstract Syntax and Semantics. W3C Recommendation.

Rector, A., & Rogers, J. (1999). Ontological issues in using a description logic to represent medical concepts: Experience from GALEN. In *IMIA WG6 Workshop, Proceedings*.

Schmidt-Schauß, M., & Smolka, G. (1991). Attributive concept descriptions with complements. *Artificial Intelligence, Elsevier*, *48*(1), 1–26.

Seidenberg, J., & Rector, A. L. (2006). Web ontology segmentation: analysis, classification and use. In *Proceedings of the 15th international conference on World Wide Web (WWW-2006), Edinburgh, Scotland, UK, May 23-26, 2006*, pp. 13–22. ACM.

Sirin, E., & Parsia, B. (2004). Pellet system description. In *Proceedings of the 2004 International Workshop on Description Logics (DL2004), Whistler, British Columbia, Canada, June 6-8, 2004*, Vol. 104 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Staab, S., & Studer, R. (Eds.). (2004). *Handbook on Ontologies*. International Handbooks on Information Systems. Springer.

Stuckenschmidt, H., & Klein, M. (2004). Structure-based partitioning of large class hierarchies. In *Proceedings of the Third International Semantic Web Conference (ISWC-2004), Hiroshima, Japan, November 7-11, 2004*, Vol. 3298 of *Lecture Notes in Computer Science*, pp. 289–303. Springer.

Tobies, S. (2000). The complexity of reasoning with cardinality restrictions and nominals in expressive description logics. *J. Artif. Intell. Res. (JAIR)*, *12*, 199–217.

Tsarkov, D., & Horrocks, I. (2006). FaCT++ description logic reasoner: System description. In *Proceedings of the Third International Joint Conference on Automated Reasoning (IJCAR 2006), Seattle, WA, USA, August 17-20, 2006*, Vol. 4130 of *Lecture Notes in Computer Science*, pp. 292–297. Springer.