Extracting Modules from Ontologies: Theory and Practice (Technical Report)

(revised version from February 21, 2007)

Bernardo Cuenca Grau, Ian Horrocks,
Yevgeny Kazakov and Ulrike Sattler

The University of Manchester
School of Computer Science
Manchester, M13 9PL, UK
{bcg, horrocks, ykazakov, sattler}@cs.man.ac.uk

Abstract

The ability to extract meaningful fragments from an ontology is essential for ontology re-use. We propose a definition of a module that guarantees to completely capture the meaning of a given set of terms, i.e., to include all axioms relevant to the meaning of these terms, and study the problem of extracting minimally sized modules. We show that the problem of determining whether a subset of an ontology is a module for a given vocabulary is undecidable even for rather restricted sub-languages of OWL DL. Hence we propose two "approximations", i.e., alternative definitions of modules for a vocabulary that still provide the above guarantee, but that are possibly too strict, and that may thus result in larger modules: the first approximation is semantic and can be checked using existing DL reasoners; the second is syntactic, and can be computed in polynomial time. Finally, we report on an empirical evaluation of our syntactic approximation that demonstrates that the modules we extract are surprisingly small.

1 Introduction

The design, maintenance, reuse, and integration of ontologies are highly complex tasks—especially for ontologies formulated in a logic-based language such as OWL. Like software engineers, "ontology engineers" need to be supported by tools and methodologies that help them to minimise the introduction of errors, i.e., to ensure that ontologies have appropriate 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 so as to take into account the fact that an OWL ontology is, in essence, a logical theory; due to the expressive power of OWL, this turns out to be difficult.

In earlier work [4], we have studied modularity in the context of *collaborative ontology development* and *controlled integration*, and defined what it means for an ontology we are developing to be safely integrated with a "foreign" ontology; roughly speaking, such an integration is safe if it does not change the meaning of the terms in the foreign ontology.

In this paper, we focus on the use of modularity to support the *partial reuse* of ontologies: continuing with the above integration scenario, as a next step, we would like to *extract*, from the foreign ontology, a small fragment that captures the meaning of the terms we use in our ontology. For example, when building an ontology describing research projects, we may use terms such as Cystic_Fibrosis and Genetic_Disorder in our descriptions of medical research projects. In order to improve the precision of our ontology, we may want to add more detail about the meaning of these terms; for reasons of cost and accuracy, we would prefer to do this by reusing information from a medical ontology. Such ontologies are, however, typically very large, and importing the whole ontology would make the consequences of the additional information costly to compute and difficult for our ontology engineers (who are not medical experts) to understand. Thus, in practice, we need to extract a module that includes just the relevant information. Ideally, this module should be *as small as possible* while still *guaranteeing* to capture the meaning of the terms used; that is, when answering arbitrary queries against our projects ontology, importing the module would give us *exactly the same answers* as if we had imported the whole medical ontology. In this case, importing the module instead of the whole ontology will have no observable effect on our ontology—apart from allowing for more efficient reasoning.

Concerning the efficiency of reasoning, the time needed to process an ontology is often too high for ontology engineering, where fast response under changes in the ontology is required, or for deployment in applications, where fast response to queries is required. The ability to extract modules in the sense described above would address both these problems: it would allow us to identify a (hopefully small) part of the ontology that is affected by a given change or that is sufficient to answer a given query—and then to reason over this part only without losing any consequences.

The contributions of this paper are as follows:

1. We propose a definition of a *module* $\mathcal{Q}_1$ within a given ontology $\mathcal{Q}$ for a given vocabulary $\mathbf{S}$.

2. We take the above definition as a starting point, and investigate the problem of computing minimal modules. We show that none of the reasonable variants of this problem is solvable in general already for rather restricted sublanguages of OWL DL. In fact, it is even not possible to determine whether a subset $\mathcal{Q}_1$ of an ontology $\mathcal{Q}$ is a module in $\mathcal{Q}$ for $\mathbf{S}$.

3. Given these negative results, we propose two "approximations", i.e., alternative definitions of a module that still guarantee to completely capture the meaning of the terms in $\mathbf{S}$, but that are possibly too strict, and that may thus result in larger modules; these approximations are based on the notion of *locality* of an ontology with respect to a vocabulary, as first introduced in [4]. The first approximation is semantic, and can be computed using existing OWL reasoners; the second one is a restriction of the first one which can be computed in polynomial time. We propose an algorithm for computing the smallest module for each of these approximations.

4. Finally, we describe our implementation and present our experimental results on a set of real-world ontologies of varying size and complexity. We show that, using our syntactic approximation, we obtain modules that are much smaller than the ones computed using existing techniques, but still sufficient to capture the meaning of the specified vocabulary.

2 Preliminaries

In this section we introduce description logics (DLs) [2] which underly modern ontology languages, such as OWL DL. A hierarchy of commonly used description logics is summarized in Table 1. The *syntax* of a description logic L is given by a signature and a set of constructors. A *signature* (or *vocabulary*) $\mathbf{S}$ of a DL is the (disjoint) union of a set $\mathbf{C}$ of *atomic concepts* $(A, B, \dots)$ representing sets of elements, a set $\mathbf{R}$ of *atomic roles* $(r, s, \dots)$ representing binary relations between elements, and a set $\mathbf{I}$ of *individuals* $(a, b, c, \dots)$ representing elements. Every DL provides *constructors* for defining the set $\mathsf{Rol}(\mathbf{S})$ of (general) *roles* $(R, S, \dots)$, the set $\mathsf{Con}(\mathbf{S})$ of (general) *concepts* $(C, D, \dots)$, and the set $\mathsf{Ax}(\mathbf{S})$ of *axioms*

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

Here $r \in \mathbf{R}$, $A \in \mathbf{C}$, $a, b \in \mathbf{I}$, $R_{(i)} \in \mathsf{Rol}(\mathbf{S})$, $C_{(i)} \in \mathsf{Con}(\mathbf{S})$, $n \geq 1$ and $S \in \mathsf{Rol}(\mathbf{S})$ is a simple role (see [9]).

Table 1: The hierarchy of standard description logics

$(\alpha, \beta, \dots)$ for a signature $\mathbf{S}$ which is a union of *role axioms* (RBox), *terminological axioms* (TBox) and *assertions* (ABox).

$\mathcal{EL}$ [1] 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 *bottom concept* $\bot$. $\mathcal{EL}$ 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* $r(a, b)$.

The *basic description logic* $\mathcal{ALC}$ [17] is obtained from $\mathcal{EL}$ by adding *complement of concepts* $\neg C$. We introduce some additional constructors as abbreviations: the *top concept* $\top$ is a shortcut for $\neg\bot$, the *disjunction of concepts* $C_1 \sqcup C_2$ stands for $\neg(\neg C_1 \sqcap \neg C_2)$, and the *value restriction* $\forall R.C$ stands for $\neg(\exists R.\neg C)$.

$\mathcal{S}$ 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 include *inverse roles* r^- (indicated by appending a letter $\mathcal{I}$), *role inclusion axioms* (RIs) also called *role hierarchies* $R_1 \sqsubseteq R_2$ (+$\mathcal{H}$), *functional roles* $\mathsf{Funct}(R)$ (+$\mathcal{F}$), *number restrictions* $(\geqslant n\, S)$ (+$\mathcal{N}$), *qualified number restrictions* $(\geqslant n\, S.C)$[1] (+$\mathcal{Q}$), and *nominals* $\{a\}$ (+$\mathcal{O}$). Nominals make it possible to construct a concept representing a singleton set $\{a\}$

[1]we consider the dual qualified number restrictions $(\leqslant n\, S)$ and $(\leqslant n\, S.C)$ as abbreviations for $\neg(\geqslant n\, S.\neg C)$ and $\neg(\geqslant n\, S.\neg C)$, respectively

(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, 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 [15], are based on description logics and, to a certain extent, are syntactic variants thereof. In particular, OWL DL corresponds to $\mathcal{SHOIN}$ [8]. In this paper, we assume an *ontology* $\mathcal{O}$ *based on a description logic* L to be a set of axioms in 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 *query answering*: 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. Given a signature $\mathbf{S} = \mathbf{R} \cup \mathbf{C} \cup \mathbf{I}$, an $\mathbf{S}$-*interpretation* (or an *interpretation based on* $\mathbf{S}$) $\mathcal{I}$ 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 $A \in \mathbf{C}$ a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, to every $r \in \mathbf{R}$ a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and to every $a \in \mathbf{I}$ an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$.

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

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

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

$$\begin{array}{ll}
\mathcal{I} \models (A \equiv C) \text{ iff } A^{\mathcal{I}} = C^{\mathcal{I}}; & \mathcal{I} \models a : C \text{ iff } a^{\mathcal{I}} \in C^{\mathcal{I}}; \\
\mathcal{I} \models (C_1 \sqsubseteq C_2) \text{ iff } C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}; & \mathcal{I} \models r(a,b) \text{ iff } \langle a^{\mathcal{I}}, b^{\mathcal{I}}\rangle \in r^{\mathcal{I}}; \\
\multicolumn{2}{l}{\mathcal{I} \models \mathsf{Trans}(r) \text{ iff } \forall xyz \in \Delta^{\mathcal{I}}[\,\langle x,y\rangle \in r^{\mathcal{I}} \wedge \langle y,z\rangle \in r^{\mathcal{I}} \Rightarrow \langle x,z\rangle \in r^{\mathcal{I}}\,];} \\
\multicolumn{2}{l}{\mathcal{I} \models \mathsf{Funct}(R) \text{ iff } \forall xyz \in \Delta^{\mathcal{I}}[\,\langle x,y\rangle \in R^{\mathcal{I}} \wedge \langle x,z\rangle \in R^{\mathcal{I}} \Rightarrow y = z\,];} \\
\multicolumn{2}{l}{\mathcal{I} \models R_1 \sqsubseteq R_2 \text{ iff } R_1^{\mathcal{I}} \subseteq R_2^{\mathcal{I}}\ ;}
\end{array}$$

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

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

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 $\mathcal{O}$. An axiom α is a *tautology* if it is implied by the empty ontology.

Let $\mathbf{S}_1, \mathbf{S}$ be signatures such that $\mathbf{S}_1 \subseteq \mathbf{S}$. The *restriction of an* $\mathbf{S}$*-interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ *to* $\mathbf{S}_1$ is an interpretation $\mathcal{I}|_{\mathbf{S}_1} = (\Delta^{\mathcal{I}_1}, \cdot^{\mathcal{I}_1})$ over $\mathbf{S}_1$ such that $\Delta^{\mathcal{I}_1} = \Delta^{\mathcal{I}}$ and $X^{\mathcal{I}_1} = X^{\mathcal{I}}$ for every $X \in \mathbf{S}_1$. An *expansion of an* $\mathbf{S}_1$*-interpretation* $\mathcal{I}_1$ *to* $\mathbf{S}$ is an $\mathbf{S}$-interpretation $\mathcal{I}$ such that $\mathcal{I}|_{\mathbf{S}_1} = \mathcal{I}_1$. A *trivial expansion of an* $\mathbf{S}_1$*-interpretation* $\mathcal{I}_1$ *to* $\mathbf{S}$ is an expansion of $\mathcal{I}_1$ to $\mathbf{S}$ such that $X^{\mathcal{I}} = \emptyset$ for every atomic concept and atomic role $X \in \mathbf{S} \setminus \mathbf{S}_1$.

3 Modules for Knowledge Reuse

For exposition, suppose that an ontology engineer wants to build an ontology about research projects. The ontology defines different types of projects according to the research topics they focus on. Suppose that the ontology engineer defines two concepts $\mathsf{Genetic\_Disorder\_Project}$ and $\mathsf{Cystic\_Fibrosis\_EUProject}$ 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 supposed to be an expert on research projects: he knows, for example, that a $\mathsf{EUProject}$ is a $\mathsf{Project}$ (axiom P3). He is unfamiliar, however, with most of the topics the projects cover and, in particular, with the terms $\mathsf{Cystic\_Fibrosis}$ and $\mathsf{Genetic\_Disorder}$ mentioned in P1 and P2. In this case, he decides to reuse the knowledge about these subjects from a well-established and widely-used medical ontology

The most straightforward way to reuse these concepts is to import the medical ontology. This may be, however, a large ontology, which deals with other matters in which the ontology engineer is not interested, such as genes, anatomy, surgical techniques, etc. Ideally, one would like to extract a (hopefully small) fragment of the medical ontology—a *module*—that describes in detail the concepts we are reusing in our ontology. Intuitively, importing the module $\mathcal{Q}_1$ into $\mathcal{P}$ instead of the full ontology $\mathcal{Q}$ should have no impact on the modeling of the ontology $\mathcal{P}$.

Continuing with the example, suppose that the concepts $\mathsf{Cystic\_Fibrosis}$ and $\mathsf{Genetic\_Disorder}$ are described in an ontology $\mathcal{Q}$ containing axioms M1-M5 in Figure 1. If we include in the module $\mathcal{Q}_1$ just the axioms that mention either $\mathsf{Cystic\_Fibrosis}$ or $\mathsf{Genetic\_Disorder}$, namely M1, M4 and M5, we lose the following dependency:

$$\mathsf{Cystic\_Fibrosis} \sqsubseteq \mathsf{Genetic\_Disorder} \tag{1}$$

The dependencies $\mathsf{Cystic\_Fibrosis} \sqsubseteq \mathsf{Genetic\_Fibrosis} \sqsubseteq \mathsf{Genetic\_Disorder}$ follow from axioms M1-M5, but not from M1, M4, M5, since the dependency $\mathsf{Cystic\_Fibrosis} \sqsubseteq \mathsf{Genetic\_Fibrosis}$ does not hold after removing M2 and M3. The dependency (1), however, is crucial for our ontology $\mathcal{P}$ as it (together with axiom P3) implies the following axiom:

$$\mathsf{Cystic\_Fibrosis\_EUProject} \sqsubseteq \mathsf{Genetic\_Disorder\_Project} \tag{2}$$

This means, in particular, that all the projects annotated with the concept name $\mathsf{Cystic\_Fibrosis\_EUProject}$ must be included in the answer for a query on the concept name $\mathsf{Genetic\_Disorder\_Project}$. Consequently, importing a part of $\mathcal{Q}$ containing only axioms that mention the terms used in $\mathcal{P}$ instead of $\mathcal{Q}$ results in an underspecified ontology. We stress that the ontology engineer might be unaware of dependency (2), even though it concerns the concepts of his primary scope.

The example above suggests that the central requirement for a module $\mathcal{Q}_1 \subseteq \mathcal{Q}$ to be reused in our ontology $\mathcal{P}$ is that $\mathcal{P} \cup \mathcal{Q}_1$ should yield the *same* logical consequences in the vocabulary of $\mathcal{P}$ as $\mathcal{P} \cup \mathcal{Q}$ does. Note that, as seen in the example, this requirement does not force us to include in $\mathcal{Q}_1$ all the axioms in $\mathcal{Q}$ that mention the vocabulary to be reused, nor does it imply that the axioms in $\mathcal{Q}$ that do not mention this vocabulary should be omitted.

Based on the discussion above, we formalize our first notion of a *module* as follows:

Definition 1 (Module). Let $\mathcal{Q}_1 \subseteq \mathcal{Q}$ be two ontologies and $\mathbf{S}$ a signature. We say that $\mathcal{Q}_1$ is an $\mathbf{S}$-*module in* $\mathcal{Q}$ w.r.t. a language L, if for every ontology $\mathcal{P}$ and every axiom α expressed in L with $\mathsf{Sig}(\mathcal{P} \cup \{\alpha\}) \cap \mathsf{Sig}(\mathcal{Q}) \subseteq \mathbf{S}$, we have $\mathcal{P} \cup \mathcal{Q} \models \alpha$ iff $\mathcal{P} \cup \mathcal{Q}_1 \models \alpha$. $\lozenge$

In Definition 1 the signature $\mathbf{S}$ acts as the *interface* signature between $\mathcal{P}$ and $\mathcal{Q}$ in the sense that it contains the symbols that $\mathcal{P}$ and α may share with $\mathcal{Q}$. It is also important to realize that there are two free parameters in Definition 1, namely the ontology $\mathcal{P}$ and the axiom α. Both $\mathcal{P}$ and α are formulated in some ontology language L, which might not necessarily be a sub-language of OWL DL.

Fixing the language L in which $\mathcal{P}$ and α can be expressed is essential in Definition 1 since it may well be the case that $\mathcal{Q}_1$ is a module in $\mathcal{Q}$ w.r.t. a language L_1, but not w.r.t. L_2. Fixing L, however, is not always reasonable. If $\mathcal{Q}_1$ is an $\mathbf{S}$-module in $\mathcal{Q}$, it should always be possible to replace $\mathcal{Q}$ with $\mathcal{Q}_1$ regardless of the particular language in which $\mathcal{P}$ and α are expressed. In fact, we may extend our ontology $\mathcal{P}$ with a set of Horn rules, or extend our query language to support arbitrary conjunctive queries. In any case, extending the ontology language for $\mathcal{P}$ and the query language for α should not prevent $\mathcal{Q}_1$ from being a module in $\mathcal{Q}$.

It is therefore convenient to formulate a more general notion of a module which abstracts from the particular language under consideration; that is, we say that $\mathcal{Q}_1$ is an $\mathbf{S}$-module in $\mathcal{Q}$ iff it is an $\mathbf{S}$-module in $\mathcal{Q}$, according to Definition 1 for *every* language L with Tarski-style set-theoretic semantics. The modules we obtain in this paper will be modules in precisely this stronger sense.

In the last few years, numerous techniques for extracting fragments of ontologies for knowledge reuse purposes have been developed. Most of these techniques rely on syntactically traversing the axioms in the ontology and employ various heuristics for determining which axioms are relevant and which are not.

An example of such a procedure is the algorithm implemented in the PROMPT-FACTOR tool [14]. 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, 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, which makes $\mathbf{S}$ to contain 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 the axioms M1-M5.

Another example is the algorithm in [18], which was used for segmentation of the medical ontology GALEN [16]. Given a signature $\mathbf{S}$ and an ontology $\mathcal{Q}$, the algorithm adds to $\mathcal{Q}_1$ all definitions $A \equiv C$ for symbols in $\mathbf{S}$, expands $\mathbf{S}$ with symbols in $\mathsf{Sig}(\mathcal{Q}_1)$, and then repeats these steps again until a fixpoint is reached.

The main idea of this algorithm is to prune irrelevant axioms by traversing the class hierarchy only "upwards" and "across existential restrictions".

In our example, for the $\mathbf{S}$ and $\mathcal{Q}$ above, the algorithm first processes the definition M1 for Cystic_Fibrosis $\in \mathbf{S}$ and extends $\mathbf{S}$ with the symbols Fibrosis, located_In, Pancreas, has_Origin, Genetic_Origin. Next, the algorithm terminates since there are no definitions mentioning any of these symbols on the left-hand-side. Thus, the fragment of our ontology extracted by the segmentation procedure from [18] consists of the single axiom M1.

Therefore, none of these algorithms is appropriate for extracting modules according to Definition 1. On the one hand, the PROMPT-FACTOR algorithm extracts many unnecessary axioms (such as M5 in our case) whereas, on the other hand, the segmentation algorithm from [18] misses essential axioms (like M2, M3 and M4).

In our example, the PROMPT-FACTOR algorithm would extract a module (though not a minimal one). In general, however, this is also not the case. 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 for single element models, and α is satisfied in every such a model; that is, $\mathcal{Q} \models \alpha$. The PROMPT-FACTOR algorithm extracts in this case $\mathcal{Q}_1 = \{A \sqsubseteq B\}$, which does not imply α.

The main problem with these algorithms is that they ignore the semantics of the ontologies. As a consequence, they may, on the one hand, extract irrelevant axioms and, on the other hand, miss essential axioms. These algorithms, however, were not intended to extract modules in accordance to a formal collection of requirements; instead, they were intended to extract "relevant parts" of ontologies which are "likely to be related" to the given signature, and they do not guarantee the correctness of the results. Correctness, however, is the primary requirement for the procedures we present in this paper.

3.1 Computing Minimal Modules

Before we formalize the main tasks related to the extraction of modules, let us outline some important properties of modules that we will exploit along this paper.

Proposition 2 [Properties of Modules]
Let $\mathcal{Q}_1 \subseteq \mathcal{Q}_2 \subseteq \mathcal{Q}_3$ be three ontologies and $\mathbf{S}$ be a signature. Then:

1. *If $\mathcal{Q}_1$ is an $\mathbf{S}$-module in $\mathcal{Q}_2$ and $\mathcal{Q}_2$ is an $\mathbf{S}$-module in $\mathcal{Q}_3$ then*
   $\mathcal{Q}_1$ is an $\mathbf{S}$-module in $\mathcal{Q}_3$ *(transitivity)*

2. *If $\mathcal{Q}_1$ is an $\mathbf{S}$-module in $\mathcal{Q}_3$ then*
   (a) $\mathcal{Q}_1$ is an $\mathbf{S}$-module in $\mathcal{Q}_2$ and (b) $\mathcal{Q}_2$ is an $\mathbf{S}$-module in $\mathcal{Q}_3$ *(convexity)*

Proof. 1. Suppose that $\mathcal{Q}_1$ is an $\mathbf{S}$-module in $\mathcal{Q}_2$ and $\mathcal{Q}_2$ is an $\mathbf{S}$-module in $\mathcal{Q}_3$. In order to prove that $\mathcal{Q}_1$ is an $\mathbf{S}$-module in $\mathcal{Q}_3$ according to Definition 1, take any ontology $\mathcal{P}$ and an axiom α such that $\mathsf{Sig}(\mathcal{P} \cup \{\alpha\}) \cap \mathsf{Sig}(\mathcal{Q}_3) \subseteq \mathbf{S}$ and $\mathcal{P} \cup \mathcal{Q}_3 \models \alpha$. We demonstrate that $\mathcal{P} \cup \mathcal{Q}_1 \models \alpha$ ($\star$):

Since $\mathcal{Q}_2$ is an $\mathbf{S}$-module in $\mathcal{Q}_3$, $\mathsf{Sig}(\mathcal{P} \cup \{\alpha\}) \cap \mathsf{Sig}(\mathcal{Q}_3) \subseteq \mathbf{S}$ and $\mathcal{P} \cup \mathcal{Q}_3 \models \alpha$, we have $\mathcal{P} \cup \mathcal{Q}_2 \models \alpha$. Since $\mathcal{Q}_1$ is an $\mathbf{S}$-module in $\mathcal{Q}_2$, $\mathsf{Sig}(\mathcal{P} \cup \{\alpha\}) \cap \mathsf{Sig}(\mathcal{Q}_2) \subseteq \mathsf{Sig}(\mathcal{P} \cup \{\alpha\}) \cap \mathsf{Sig}(\mathcal{Q}_3) \subseteq \mathbf{S}$, and $\mathcal{P} \cup \mathcal{Q}_2 \models \alpha$, we have $\mathcal{P} \cup \mathcal{Q}_1 \models \alpha$ ($\star$).

2.(a) Suppose that $\mathcal{Q}_1$ is an $\mathbf{S}$-module in $\mathcal{Q}_3$. In order to prove that $\mathcal{Q}_1$ is an $\mathbf{S}$-module in $\mathcal{Q}_2$, consider any ontology $\mathcal{P}$ and an axiom α such that $\mathsf{Sig}(\mathcal{P} \cup \{\alpha\}) \cap \mathsf{Sig}(\mathcal{Q}_2) \subseteq \mathbf{S}$ and $\mathcal{P} \cup \mathcal{Q}_2 \models \alpha$. We demonstrate that $\mathcal{P} \cup \mathcal{Q}_1 \models \alpha$ ($\sharp$):

Without loss of generality, we can assume that $\mathsf{Sig}(\mathcal{P} \cup \{\alpha\}) \cap \mathsf{Sig}(\mathcal{Q}_3) \subseteq \mathbf{S}$, since the symbols that are in $\mathsf{Sig}(\mathcal{P} \cup \{\alpha\})$ but not in $\mathsf{Sig}(\mathcal{Q}_2)$ could be renamed so that they are not contained in $\mathsf{Sig}(\mathcal{Q}_3)$. Since $\mathcal{Q}_1$ is an $\mathbf{S}$-module in $\mathcal{Q}_3$ and $\mathcal{P} \cup \mathcal{Q}_3 \models \mathcal{P} \cup \mathcal{Q}_2 \models \alpha$, we have $\mathcal{P} \cup \mathcal{Q}_1 \models \alpha$ ($\sharp$).

2.(b) Suppose that $\mathcal{Q}_1$ is an $\mathbf{S}$-module in $\mathcal{Q}_3$. In order to prove that $\mathcal{Q}_2$ is an $\mathbf{S}$-module in $\mathcal{Q}_3$, consider any ontology $\mathcal{P}$ and an axiom α such that $\mathsf{Sig}(\mathcal{P} \cup \{\alpha\}) \cap \mathsf{Sig}(\mathcal{Q}_3) \subseteq \mathbf{S}$ and $\mathcal{P} \cup \mathcal{Q}_3 \models \alpha$. We demonstrate that $\mathcal{P} \cup \mathcal{Q}_2 \models \alpha$ ($\dagger$):

Since $\mathcal{Q}_1$ is an $\mathbf{S}$-module in $\mathcal{Q}_3$, $\mathsf{Sig}(\mathcal{P} \cup \{\alpha\}) \cap \mathsf{Sig}(\mathcal{Q}_3) \subseteq \mathbf{S}$, and $\mathcal{P} \cup \mathcal{Q}_3 \models \alpha$, we have $\mathcal{P} \cup \mathcal{Q}_1 \models \alpha$. Since $\mathcal{Q}_1 \subseteq \mathcal{Q}_2$, we have $\mathcal{P} \cup \mathcal{Q}_2 \models \alpha$ ($\dagger$). $\square$

Part 2(a) of Proposition 2 says essentially that every superset of an $\mathbf{S}$-module of this ontology is also an $\mathbf{S}$-module of the ontology. This means, in particular, that it is sufficient to compute only the minimal modules of an ontology in order to have a complete information about all the modules.

Therefore, it makes sense to focus only on minimal modules. We say that $\mathcal{Q}_1$ is a *minimal* $\mathbf{S}$*-module in* $\mathcal{Q}$ if there is no $\mathcal{Q}_2 \subsetneq \mathcal{Q}_1$ that is also an $\mathbf{S}$-module in $\mathcal{Q}$. In our example from Figure 1, there are two minimal $\mathbf{S}$-modules $\mathcal{Q}_1 = \{\mathrm{M1}, \mathrm{M2}, \mathrm{M4}\}$ and $\mathcal{Q}_2 = \{\mathrm{M1}, \mathrm{M3}, \mathrm{M4}\}$: if we remove any axiom from them, the dependency (1) will no longer hold. Hence minimal modules are not necessarily unique. While in some cases it is reasonable to extract all minimal modules, in others it may suffice to extract just one. Thus, given $\mathcal{Q}$ and $\mathbf{S}$, the following tasks are of interest:

T1. compute *all* minimal $\mathbf{S}$-modules in $\mathcal{Q}$
T2. compute *some* minimal $\mathbf{S}$-module in $\mathcal{Q}$ (3)

Intuitively, task T2 should be simpler than T1. That is, any procedure which solves the task T1, also provides a solution for task T2. Surprisingly, the converse of this property holds as well: any procedure for T2 can be turned into a procedure for T1. The following lemma is the key property underlying this reduction:

Lemma 3 [A Criterium for Minimal Modules]
Let $\mathcal{Q}$ be an ontology and $\mathbf{S}$ be a signature. Let $\mathcal{M}$ be the set of all subsets $\mathcal{Q}_2$ of $\mathcal{Q}$ such that $\mathcal{Q}_2$ is a minimal (and hence is the only) $\mathbf{S}$-module in $\mathcal{Q}_2$.

Then $\mathcal{Q}_1$ is a minimal $\mathbf{S}$-module in $\mathcal{Q}$ iff (i) $\mathcal{Q}_1 \in \mathcal{M}$, and (ii) there is no $\mathcal{Q}_2 \in \mathcal{M}$ such that $\mathcal{Q}_1 \subsetneq \mathcal{Q}_2$.

Proof. ($\Rightarrow$) Suppose $\mathcal{Q}_1$ is a minimal $\mathbf{S}$-module in $\mathcal{Q}$. We need to show that properties (i) and (ii) above hold for $\mathcal{Q}_1$.

(i) Suppose, to the contrary, that the property (i) does not hold for $\mathcal{Q}_1$, i.e. $\mathcal{Q}_1$ is not a minimal module in $\mathcal{Q}_1$. Then there exists a $\mathcal{Q}_2 \subsetneq \mathcal{Q}_1 \subseteq \mathcal{Q}$ such that $\mathcal{Q}_2$ is an $\mathbf{S}$-module in $\mathcal{Q}_1$. Since $\mathcal{Q}_1$ is an $\mathbf{S}$-module in $\mathcal{Q}$, By the part 1 of Proposition 2 (transitivity), $\mathcal{Q}_2$ is an $\mathbf{S}$-module in $\mathcal{Q}$. Hence $\mathcal{Q}_1$ is not a minimal module in $\mathcal{Q}$ contrary to what has been assumed.

(ii) Suppose, to the contrary, that the property (ii) does not hold for $\mathcal{Q}_1$, that is, there exists $\mathcal{Q}_2 \in \mathcal{M}$ such that $\mathcal{Q}_1 \subsetneq \mathcal{Q}_2 \subseteq \mathcal{Q}$. Since $\mathcal{Q}_1$ is an $\mathbf{S}$-module in $\mathcal{Q}$, by the part 2(a) of Proposition 2, $\mathcal{Q}_1$ is an $\mathbf{S}$-module in $\mathcal{Q}_2$. Hence $\mathcal{Q}_2 \notin \mathcal{M}$ by the definition of $\mathcal{M}$ (since $\mathcal{Q}_2$ is not a minimal $\mathbf{S}$-module in $\mathcal{Q}_2$), which yields a contradiction.

($\Leftarrow$) Assume that conditions (i) and (ii) above hold for $\mathcal{Q}_1$, but $\mathcal{Q}_1$ is not a minimal $\mathbf{S}$-module in $\mathcal{Q}$. There are two cases possible: (a) $\mathcal{Q}_1$ is not an $\mathbf{S}$-module in $\mathcal{Q}$, and (b) $\mathcal{Q}_1$ is an $\mathbf{S}$-module in $\mathcal{Q}$, but not a minimal $\mathbf{S}$-module.

In the case (a), there has to be a minimal $\mathbf{S}$-module $\mathcal{Q}_2$ in $\mathcal{Q}$ such that $\mathcal{Q}_1 \subsetneq \mathcal{Q}_2 \subseteq \mathcal{Q}$. By the direction ($\Rightarrow$) of the lemma applied to $\mathcal{Q}_2$, we have $\mathcal{Q}_2 \in \mathcal{M}$. But this contradicts the condition (ii), since $\mathcal{Q}_1 \in \mathcal{M}$ and $\mathcal{Q}_1 \subsetneq \mathcal{Q}_2$.

In the case (b), there is a minimal $\mathbf{S}$-module $\mathcal{Q}_2$ in $\mathcal{Q}$ such that $\mathcal{Q}_2 \subsetneq \mathcal{Q}_1 \subseteq \mathcal{Q}$. By the property 2.(a) of Proposition 2, $\mathcal{Q}_2$ is an $\mathbf{S}$-module in $\mathcal{Q}_1$, which contradicts the condition (i) since $\mathcal{Q}_1$ is not a minimal $\mathbf{S}$-module in $\mathcal{Q}_1$. $\square$

We use this property to show that tasks T1 and T2 are indeed inter-reducible:

Proposition 4 *Tasks T1 and T2 from* (3) *are inter-reducible.*

Proof. As it has been already pointed out, using a procedure for task T1 one can obtain a procedure for task T2 by just returning any of the computed minimal $\mathbf{S}$-modules in $\mathcal{Q}$.

Now suppose we have a procedure **P2** for task T2, namely, that given a signature $\mathbf{S}$ and an ontology $\mathcal{Q}$ returns some minimal $\mathbf{S}$-module $\mathcal{Q}_1$ in $\mathcal{Q}$. We construct a procedure **P1** that returns all minimal $\mathbf{S}$-modules, which is based on the criterium for minimal $\mathbf{S}$-modules formulated in Lemma 3. Note that procedure **P2** satisfies the following property:

*Given $\mathbf{S}$ and $\mathcal{Q}_2$, the procedure **P2** for T2 returns $\mathcal{Q}_2$ if and only if $\mathcal{Q}_2$ is the only minimal $\mathbf{S}$-module in $\mathcal{Q}_2$.* (4)

Procedure **P1** should work as follows. Given $\mathbf{S}$ and $\mathcal{Q}$, **P1** first computes the set $\mathcal{M}$ of subsets $\mathcal{Q}_2$ in $\mathcal{Q}$ such that $\mathcal{Q}_2$ is the only $\mathbf{S}$-module in $\mathcal{Q}_2$ using property (4) of procedure **P2**. More precisely, in order to compute $\mathcal{M}$, we enumerate all the subsets of $\mathcal{Q}$ and select those subsets $\mathcal{Q}_2$ for which **P2** returns $\mathcal{Q}_2$. Next, **P1** returns those sets from $\mathcal{M}$ that are contained in no other set from $\mathcal{M}$. By Lemma 3, **P1** returns exactly all minimal $\mathbf{S}$-modules in $\mathcal{Q}$. □

There are other variations of the task T1 and T2 that may be of interest. For example, instead of minimal modules, one might be interested only in modules of the *smallest size*. An $\mathbf{S}$-module in $\mathcal{Q}$ has the smallest size iff no other $\mathbf{S}$-module in $\mathcal{Q}$ has a smaller number of axioms:

T1s.	compute *all* the smallest in size $\mathbf{S}$-modules in $\mathcal{Q}$	(5)
T2s.	compute *some* smallest in size $\mathbf{S}$-module in $\mathcal{Q}$	

Clearly, if an $\mathbf{S}$-module in $\mathcal{Q}$ is of the smallest size, then it is a minimal $\mathbf{S}$-module; the converse, however, does not necessarily hold. It is easy to see that any procedure for T1 could be turned into a procedure for T1s and T2s: given all minimal modules, we can simply count the number of axioms they contain and retrieve the modules with the fewest number of axioms. Conversely, any procedure for T1s or T2s can be used for solving T2 since a module of smallest size is also a minimal module. As a conclusion, we have:

Proposition 5 *All tasks from* (3) *and* (5) *are inter-reducible.*

Recall that there are two minimal $\mathbf{S}$-modules $\mathcal{Q}_1 = \{\mathrm{M1}, \mathrm{M2}, \mathrm{M4}\}$ and $\mathcal{Q}_2 = \{\mathrm{M1}, \mathrm{M3}, \mathrm{M4}\}$ in our ontology $\mathcal{Q}$ from Figure 1. That is, in a certain sense, the axioms M1–M4 are essential for the dependency (1). In certain situations, one can be interested in computing just the set $\mathcal{Q}_e$ of such essential axioms, instead of computing all minimal modules. This is the case, for example, if the ontology engineer wants to compute a module that is "safe" under removal of axioms: if we remove M2 from $\mathcal{Q}$, then $\mathcal{Q}'_1 = \mathcal{Q}_1 \setminus \{\mathrm{M2}\} = \{\mathrm{M1}, \mathrm{M4}\}$ is no longer an $\mathbf{S}$-module for the updated ontology $\mathcal{Q}' := \mathcal{Q} \setminus \{\mathrm{M2}\}$ since the dependency (1) is lost, but $\mathcal{Q}'_e := \mathcal{Q}_e \setminus \{\mathrm{M2}\}$ is still a module in $\mathcal{Q}$. This example suggests the following definition:

Definition 6 (Essential Axiom). Given a signature $\mathbf{S}$ and an ontology $\mathcal{Q}$, we say that an axiom $\alpha \in \mathcal{Q}$ is $\mathbf{S}$-*essential in* $\mathcal{Q}$ w.r.t. L if α belongs to some minimal $\mathbf{S}$-module in $\mathcal{Q}$ w.r.t. L. $\Diamond$

Hence, the following task may also be of interest:

T3. compute *the union* of all minimal $\mathbf{S}$-modules in $\mathcal{Q}$, which is the set of all $\mathbf{S}$-essential axioms in $\mathcal{Q}$ (6)

Obviously, task T3 is at least not harder then task T1:

Proposition 7 *Task T1 is reducible to task T3, that is, any procedure for T1 can be used for solving T3.*

It is not clear, however, whether the procedure for T3 can be used to obtain a procedure for T1. Nevertheless, as we will demonstrate Section 3.2, this issue is not relevant since all of the tasks formulated above are algorithmically unsolvable already for simple sub-languages of OWL DL.

3.2 Modules and Conservative Extensions

The notion of a module is closely related to the notion of a conservative extension which has been used to characterize formal requirements in ontology integration tasks [7, 5, 4, 11]. In the literature we can find at least two different notions of conservative extensions in the context of ontologies [11]:

Definition 8 (Conservative Extensions).

Let $\mathcal{Q}_1 \subseteq \mathcal{Q}$ be two ontologies, $\mathbf{S}$ a signature and L a logic.

We say that $\mathcal{Q}$ is a *deductive* $\mathbf{S}$-*conservative extension* of $\mathcal{Q}_1$ w.r.t. L, if for every axiom α over L with $\mathsf{Sig}(\alpha) \subseteq \mathbf{S}$, we have $\mathcal{Q} \models \alpha$ iff $\mathcal{Q}_1 \models \alpha$.

We say that $\mathcal{Q}$ is a *model* $\mathbf{S}$-*conservative extension* of $\mathcal{Q}_1$ if, for every model $\mathcal{I}_1$ of $\mathcal{Q}_1$, there exists a model $\mathcal{I}$ of $\mathcal{Q}$ such that $\mathcal{I}|_{\mathbf{S}} = \mathcal{I}_1|_{\mathbf{S}}$. $\Diamond$

Intuitively, an ontology $\mathcal{Q}$ is a deductive conservative extension of an ontology $\mathcal{Q}_1 \subseteq \mathcal{Q}$ for a signature $\mathbf{S}$ iff every logical consequence α of $\mathcal{Q}$ constructed using only symbols from $\mathbf{S}$ is already a consequence of $\mathcal{Q}_1$; that is, the additional axioms in $\mathcal{Q}$ do not add new logical consequences over the vocabulary $\mathbf{S}$. Analogously to modules, the notion of a deductive conservative extension depends on the ontology language L in which $\mathcal{Q}$ and α are expressed.

In contrast, model conservative extensions are not defined in terms of logical entailment, but using the models directly. Intuitively, an ontology $\mathcal{Q}$ is a model conservative extension of $\mathcal{Q}_1 \subseteq \mathcal{Q}$ if every model of $\mathcal{Q}_1$ can be expanded to a model of $\mathcal{Q}$ by interpreting new symbols and leaving the interpretations of the old symbols unchanged.

The notion of semantic conservative extension is strictly stronger than the syntactic one [11] since it does not depend on expressivity of the ontology language. That is, if $\mathcal{Q}$ is a model $\mathbf{S}$-conservative extension of $\mathcal{Q}_1$, it is also a deductive $\mathbf{S}$-conservative extension of $\mathcal{Q}_1$, but not necessarily vice versa.

Example 9 Let $\mathcal{Q}$ be the ontology consisting of axioms $\mathrm{M1} - \mathrm{M5}$ in Figure 1. Let $\mathbf{S} = \{\mathsf{Cystic\_Fibrosis},\ \mathsf{Genetic\_Disorder}\}$ and $\mathcal{Q}_1 = \{\mathrm{M1}, \ldots, \mathrm{M4}\}$. We show that $\mathcal{Q}$ is a model $\mathbf{S}$-conservative extension of $\mathcal{Q}_1$ and, hence, also a deductive conservative extension of $\mathcal{Q}_1$.

Let $\mathcal{I}_1$ be an arbitrary model of $\mathcal{Q}_1$. We demonstrate that we can always construct a model $\mathcal{I}$ of $\mathcal{Q}$ which interprets the symbols from $\mathbf{S}$ in the same way as $\mathcal{I}_1$ does, i.e. $\mathcal{I}|_{\mathbf{S}} = \mathcal{I}_1|_{\mathbf{S}}$.

Indeed, let $\mathcal{I}$ be as $\mathcal{I}_1$ except for the interpretation of the atomic concepts $\mathsf{DEFBI\_Gene}$ and $\mathsf{Immuno\_Protein\_Gene}$, and the atomic role $\mathsf{associatedWith}$, all of which we interpret in $\mathcal{I}$ as the empty set. Note that these atomic concepts and this atomic role do not occur in $\mathcal{Q}_1$. Hence, $\mathcal{I}$ interprets the concepts in $\mathcal{Q}_1$ exactly like $\mathcal{I}_1$, and so $\mathcal{I}$ is a model of $\mathcal{Q}_1$. Furthermore, $\mathcal{I}$ is a model of M5 since the concepts on the left-hand-side and the right-hand-side of this axiom are both interpreted as the empty set. Thus, $\mathcal{Q}$ is a model $\mathbf{S}$-conservative extension of $\mathcal{Q}_1$.

In fact, it was sufficient to take any expansion $\mathcal{I}$ of $\mathcal{I}_1$ in which $\mathsf{DEFBI\_Gene}$ is interpreted as the empty set. Hence $\mathcal{Q}$ is a model $\mathbf{S}$-conservative extension of $\mathcal{Q}_1$ for every $\mathbf{S}$ that does not contain $\mathsf{DEFBI\_Gene}$ since M5 is satisfied in every interpretation where this concept is interpreted as the empty set.

Now, if we remove M2 and M3 from $\mathcal{Q}_1$, then $\mathcal{Q}$ is no longer an $\mathbf{S}$-conservative extension of $\mathcal{Q}_1$ for $\mathbf{S} = \{\mathsf{Cystic\_Fibrosis},\ \mathsf{Genetic\_Disorder}\}$. Indeed, it is possible to find an interpretation $\mathcal{I}_1$ of the remaining axioms M1 and M4 from $\mathcal{O}_1$, in which $\mathsf{Genetic\_Disorder}$ is interpreted as the empty set, but $\mathsf{Cystic\_Fibrosis}$ is not. For example, consider an interpretation $\mathcal{I}_1 = (\{a\}, \cdot^{\mathcal{I}_1})$ with:

$$\begin{aligned}
&\mathsf{Cystic\_Fibrosis}^{\mathcal{I}_1} = \mathsf{Fibrosis}^{\mathcal{I}_1} = \mathsf{Pancreas}^{\mathcal{I}_1} = \mathsf{Genetic\_Origin}^{\mathcal{I}_1} = \{a\};\\
&\mathsf{located\_In}^{\mathcal{I}_1} = \mathsf{has\_Origin}^{\mathcal{I}_1} = \{(a,a)\}; \quad \text{and}\\
&\mathsf{Genetic\_Fibrosis}^{\mathcal{I}_1} = \mathsf{Genetic\_Disorder}^{\mathcal{I}_1} = \emptyset.
\end{aligned}$$

It is easy to see that $\mathcal{I}_1$ is a model of M1 and M4, but there is no model $\mathcal{I}$ of $\mathcal{Q}$ such that $\mathcal{I}|_{\mathbf{S}} = \mathcal{I}_1|_{\mathbf{S}}$. Indeed, for every model $\mathcal{I}$ of $\mathcal{Q}$, we must have $\mathcal{I} \models \alpha :=$ $(\mathsf{Cystic\_Fibrosis} \sqsubseteq \mathsf{Genetic\_Disorder})$ because $\mathcal{Q} \models \alpha$. However, this would imply also that $\mathcal{I}_1 \models \alpha$, since $\mathcal{I}|_{\mathbf{S}} = \mathcal{I}_1|_{\mathbf{S}}$, but this does not hold for $\mathcal{I}_1$ defined above. $\Diamond$

Although Definition 1 is close to the notion of deductive conservative extension, there are two important differences. First, in the definition of deductive conservative extension, the logical consequences are considered only w.r.t. the ontologies $\mathcal{Q}$ and $\mathcal{Q}_1$ of interest whereas, in our definition of module, all the possible ontologies $\mathcal{P}$ in which the module can be used are taken into account. Second, in the definition of deductive conservative extension, the signature of α is required to be a subset of $\mathbf{S}$ whereas, in our definition of module, only the common part of $\{\alpha\} \cup \mathcal{P}$ and $\mathcal{Q}$ is required to be a subset of $\mathbf{S}$. Despite these differences, the two notions of conservative extensions are related to our notion of module:

Proposition 10 [Modules vs. Conservative Extensions]
Let $\mathcal{Q}_1 \subseteq \mathcal{Q}$ be two ontologies. Then:

1. *If $\mathcal{Q}_1$ is an $\mathbf{S}$-module in $\mathcal{Q}$ w.r.t.* L *then $\mathcal{Q}$ is a deductive $\mathbf{S}$-conservative extension of $\mathcal{Q}_1$ w.r.t.* L*;*

2. *If $\mathcal{Q}$ is a model $\mathbf{S}$-conservative extension of $\mathcal{Q}_1$ then $\mathcal{Q}_1$ is an $\mathbf{S}$-module in $\mathcal{Q}$ for every ontology language* L *with Tarski-style set-theoretic semantics.*

Proof. 1. Let α be an axiom with $\mathsf{Sig}(\alpha) \in \mathbf{S}$ such that $\mathcal{Q} \models \alpha$. We have to show that $\mathcal{Q}_1 \models \alpha$ ($\star$). Take $\mathcal{P} := \emptyset$ (the empty ontology). Since $\mathcal{Q}_1$ is a module in $\mathcal{Q}$, $\mathsf{Sig}(\mathcal{P} \cup \{\alpha\}) \cap \mathsf{Sig}(\mathcal{Q}) \subseteq \mathbf{S}$, and $\mathcal{P} \cup \mathcal{Q} = \mathcal{Q} \models \alpha$, by Definition 1, we have $\mathcal{Q}_1 = \mathcal{P} \cup \mathcal{Q}_1 \models \alpha$.

2. Assume that $\mathcal{Q}$ is a model $\mathbf{S}$-conservative extension of $\mathcal{Q}_1$, but $\mathcal{Q}_1$ is not an $\mathbf{S}$-module in $\mathcal{Q}$ w.r.t. some logic L. According to Definition 1, this means that there exists an ontology $\mathcal{P}$ and an axiom α over L with $\mathsf{Sig}(\mathcal{P} \cup \{\alpha\}) \cap \mathsf{Sig}(\mathcal{Q}) \subseteq \mathbf{S}$, such that $\mathcal{P} \cup \mathcal{Q} \models \alpha$ but $\mathcal{P} \cup \mathcal{Q}_1 \not\models \alpha$. The last implies that for some interpretation $\mathcal{I}_1$, we have $\mathcal{I}_1 \models \mathcal{P} \cup \mathcal{Q}_1$, but $\mathcal{I}_1 \not\models \alpha$. Let $\mathcal{I}_1' := \mathcal{I}_1|_{\mathbf{S} \cup \mathsf{Sig}(\mathcal{Q})}$. Obviously, $\mathcal{I}_1' \models \mathcal{Q}_1$. By Definition 8, since $\mathcal{Q}$ is a model $\mathbf{S}$-conservative extension of $\mathcal{Q}_1$, there exists an interpretation $\mathcal{I}'$ such that $\mathcal{I}' \models \mathcal{Q}$ and $\mathcal{I}'|_{\mathbf{S}} = \mathcal{I}_1'|_{\mathbf{S}}$. Let $\mathcal{I}$ be the expansion of $\mathcal{I}'|_{\mathbf{S} \cup \mathsf{Sig}(\mathcal{Q})}$ to $\mathsf{Sig}(\mathcal{P} \cup \{\alpha\})$ by setting $X^{\mathcal{I}} := X^{\mathcal{I}_1}$ for every $X \in \mathsf{Sig}(\mathcal{P} \cup \{\alpha\}) \setminus \mathbf{S}$. Note that we also have $\mathcal{I}|_{\mathbf{S}} = \mathcal{I}'|_{\mathbf{S}} = \mathcal{I}_1'|_{\mathbf{S}} = \mathcal{I}_1|_{\mathbf{S}}$, hence $\mathcal{I}|_{\mathsf{Sig}(\mathcal{P} \cup \{\alpha\})} = \mathcal{I}_1|_{\mathsf{Sig}(\mathcal{P} \cup \{\alpha\})}$, and so $\mathcal{I} \models \mathcal{P}$ and $\mathcal{I} \not\models \alpha$. Since $\mathcal{I}|_{\mathbf{S} \cup \mathsf{Sig}(\mathcal{Q})} = \mathcal{I}'|_{\mathbf{S} \cup \mathsf{Sig}(\mathcal{Q})}$ and $\mathcal{I}' \models \mathcal{Q}$, we have $\mathcal{I} \models \mathcal{Q}$, which yields a contradiction. $\square$

Proposition 10 shows that our notion of module stays "in between" the two notions of conservative extensions. In particular, by applying Property 2 in Proposition 10 to Example 9, we can show that the axioms M1-M4 in Figure 1 constitute a module in the ontology $\mathcal{Q}$, consisting of M1-M5. The converse of Property 1 in Proposition 10, however, does not hold in general:

Example 11 Let $\mathcal{Q}_1 = \{\}$, $\mathcal{Q} = \{\top \sqsubseteq \exists R.A\}$ and $\mathbf{S} = \{A\}$. The ontology $\mathcal{Q}$ is a deductive $\mathbf{S}$-conservative extension of $\mathcal{Q}_1$ w.r.t. $\mathcal{ALC}$. Indeed, every $\mathcal{ALC}$-axiom $\alpha = (C_1 \sqsubseteq C_2)$ over $\mathbf{S} = \{A\}$, is equivalent in $\mathcal{ALC}$ to either $\top \sqsubseteq \top$, $\top \sqsubseteq \bot$, $\top \sqsubseteq A$ or $A \sqsubseteq \bot$, which are indistinguishable by $\mathcal{Q}_1$ and $\mathcal{Q}$—that is, the axiom is implied by $\mathcal{Q}_1$ iff it is implied by $\mathcal{Q}$. $\mathcal{Q}_1$, however, is not an $\mathbf{S}$-module in $\mathcal{Q}$. Consider an $\mathcal{ALC}$-ontology $\mathcal{P} = \{A \sqsubseteq \bot\}$, which is constructed over $\mathbf{S}$. It is easy to see that $\mathcal{P} \cup \mathcal{Q} \models \top \sqsubseteq \bot$, but $\mathcal{P} \cup \mathcal{Q}_1 \not\models \top \sqsubseteq \bot$. $\Diamond$

Note that the construction in Example 11 also shows that the notion of deductive conservative extension is strictly weaker than the notion of model conservative extension (as shown in [11]): $\mathcal{Q}$ is a deductive conservative extension of $\mathcal{Q}_1$ but, according to Property 2 in Proposition 10, it is not a model conservative extension.

Given the relationships between our definition of module and conservative extensions, it is worth examining the computational complexity of the associated problems. The problem of deciding whether $\mathcal{Q}$ is an $\mathbf{S}$-conservative extension of $\mathcal{Q}_1$ has been studied in [11], where it is proved to be 2NEXPTIME-complete for $\mathcal{ALCIQ}$ (roughly OWL-Lite) and undecidable for OWL DL. For model conservative extensions, the problem is highly undecidable (non recursively enumerable), even for $\mathcal{ALC}$ [11].

The decidability result from [11] for deductive conservative extensions, however, does not transfer to our problem since an ontology $\mathcal{Q}$ may well be an $\mathbf{S}$-deductive conservative extension of $\mathcal{Q}_1$, but still $\mathcal{Q}_1$ might not be an $\mathbf{S}$-module in $\mathcal{Q}$. In fact, we show that our problem is already undecidable for $\mathcal{ALC}$ ontologies when the language allows for nominals:

Theorem 12 [Undecidability for Essential Axioms]
Given a signature $\mathbf{S}$, an $\mathcal{ALC}$-ontology $\mathcal{Q}$ and an axiom $\alpha \in \mathcal{Q}$, it is undecidable whether α is $\mathbf{S}$-essential in $\mathcal{Q}$ w.r.t. $\mathrm{L} = \mathcal{ALCO}$.

Proof. The proof is a variation of the construction for undecidability of deciding deductive conservative extensions in $\mathcal{ALCQIO}$ given [11], based on reduction to domino tiling problems.

A domino system is a triple $D = (T, H, V)$ where T 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 $t_{(\cdot,\cdot)}$ that assigns to every pair of integers $i, j \geq 1$ an element $t_{i,j} \in T$, such that $(t_{i,j}, t_{i,j+1}) \in V$ and $(t_{i,j}, t_{i+1,j}) \in H$. 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 collection 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 $\mathcal{D}$ that admit a periodic solution. Note

$$
\begin{array}{lll}
(q_1) & \top \sqsubseteq A_{t_1} \sqcup \cdots \sqcup A_{t_n} & \text{if } T = \{t_1, \ldots, t_n\} \\
(q_2) & A_{t_i} \sqcap A_{t_j} \sqsubseteq \bot & \text{whenever } t_i \neq t_j, \\
(q_3) & A_{t_i} \sqsubseteq \exists r_H.(\bigsqcup_{(t_i,t_j)\in H} A_{t_j}) & \quad t_i, t_j \in T \\
(q_4) & A_{t_i} \sqsubseteq \exists r_V.(\bigsqcup_{(t_i,t_j)\in V} A_{t_j}) & \\
\hline
(q_5) & \top \sqsubseteq \exists \boldsymbol{s}.[\exists r_H.\exists r_V.\boldsymbol{B} \sqcap \exists r_V.\exists r_H.\neg\boldsymbol{B}] & =: \alpha
\end{array}
$$

Figure 2: An ontology $\mathcal{Q}$ for a domino system D

that $\mathcal{D}_{ps} \subseteq \mathcal{D}_s$. It is well-known [3, 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 every domino system D, we construct a signature $\mathbf{S} = \mathbf{S}(D)$, an ontology $\mathcal{Q} = \mathcal{Q}(D)$ which is an $\mathcal{ALC}$-TBox, and an axiom $\alpha \in \mathcal{Q}$ such that:

(a) if D does not have a solution then α is not $\mathbf{S}$-essential in $\mathcal{Q}$ w.r.t. L, and

(b) if D has a periodic solution then α is $\mathbf{S}$-essential in $\mathcal{Q}$.

In other words, for the set $\mathcal{D}'$ of domino systems D such that α is $\mathbf{S}$-essential in $\mathcal{Q} = \mathcal{Q}(D)$ w.r.t. L, 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 $\mathbf{S}$-essential axioms, because otherwise one can use this problem for deciding membership in $\mathcal{D}'$.

The signature $\mathbf{S}$, ontology $\mathcal{Q}$ and axiom $\alpha \in \mathcal{Q}$ 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. We define $\mathcal{Q}$ to consists of axioms (q_1)–(q_5) from Figure 2 and set α to be the axiom (q_5).

Axioms of form (q_1)–(q_4) express that every domain element in a model for $\mathcal{Q}$ is assigned with a unique tile $t \in T$ and has horizontal and vertical matching successors. Axiom (q_5) plays a special role in our reduction for excluding those models of $\mathcal{Q}$ for which the horizontal and vertical matching relations do not commute. It is easy to show that all axioms from $\mathcal{Q}$ are *independent*, i.e. none of the axioms is a logical consequence of the remaining axioms. In the remainder, we prove properties (a) and (b) formulated above.

In order to prove property (a), assume that α is $\mathbf{S}$-essential in $\mathcal{Q}$ w.r.t. L. We demonstrate that D has a solution in this case.

Let $\mathcal{Q}^\alpha$ be a minimal $\mathbf{S}$-module in $\mathcal{Q}$ containing α. Note that $\mathcal{Q}^\alpha$ implies all axioms of form (q_1)–(q_4) in $\mathcal{Q}$, since the signature of these axioms is a subset of

$\mathbf{S}$. Since $\mathcal{Q}^\alpha$ contains α and all axioms of $\mathcal{Q}$ are independent, this is only possible when $\mathcal{Q}^\alpha = \mathcal{Q}$.

Since $\mathcal{Q}^\alpha = \mathcal{Q}$ is a minimal $\mathbf{S}$-module in $\mathcal{Q}$, the set $\mathcal{Q}_1 := \mathcal{Q} \setminus \{\alpha\}$ is not an $\mathbf{S}$-module in $\mathcal{Q}$, and so, by the part 2 of Proposition 10, $\mathcal{Q}$ is not a model $\mathbf{S}$-conservative extension of $\mathcal{Q}_1$. This means that there is an $\mathbf{S}$-interpretation $\mathcal{I}_1 = (\Delta, \cdot^{\mathcal{I}_1})$ that is a model of the axioms of form (q_1)–(q_4), but which cannot be expanded to a model of α by interpreting atomic role s and atomic concept B. We claim that this is possible only if relations r_H and r_V commute in $\mathcal{I}_1$, that is, whenever $r_H(a,b)$, $r_V(b,c_1)$, $r_V(a,d)$ and $r_H(d,c_2)$ hold in $\mathcal{I}_1$, then it must be the case that $c_1 = c_2$. Indeed, otherwise one can expand $\mathcal{I}_1$ to a model $\mathcal{I}$ of α by setting $s^\mathcal{I} = \{(x,a) \mid x \in \Delta\}$ and $B^\mathcal{I} = \{c_1\}$. Since $\mathcal{I}$ satisfies all formulas of forms (q_1)–(q_4) and admits commutativity property for relations r_H and r_V, it is easy to see that D has a solution.

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 α is $\mathbf{S}$-essential in $\mathcal{Q}$ by showing that $\mathcal{Q}_1 := \mathcal{Q} \setminus \{\alpha\}$ is not an $\mathbf{S}$-module in $\mathcal{Q}$. For this purpose we construct an $\mathcal{ALCO}$-ontology $\mathcal{P}$ such that $\mathcal{P} \cup \mathcal{Q} \models \bot$, but $\mathcal{P} \cup \mathcal{Q}_1 \not\models \bot$. We define $\mathcal{P}$ such that every model of $\mathcal{P}$ is a finite encoding of the periodic solution $t_{i,j}$. 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 add the following axioms to $\mathcal{P}$

$$\begin{array}{ll}
(p_1)\ a_{i,j} : A_{t_{i,j}}, & (p_4)\ \top \sqsubseteq \bigsqcup_{1 \leq i \leq m,\, 1 \leq j \leq n} \{a_{i,j}\}, \\
(p_2)\ r_V(a_{i_1,j}, a_{i_2,j}), & (p_5)\ \{a_{i_1,j}\} \sqsubseteq \forall r_V.\{a_{i_2,j}\},\ i_2 = i_1 + 1 \mod m \\
(p_3)\ r_H(a_{i,j_1}, a_{i,j_2}), & (p_6)\ \{a_{i,j_1}\} \sqsubseteq \forall r_H.\{a_{i,j_2}\},\ j_2 = j_1 + 1 \mod n
\end{array}$$

The axioms (p_1)–(p_4) encode the solution $t_{i,j}$ for $\mathcal{D}$, and so, ensure that axioms (q_1)–(q_4) are satisfied. The axioms (p_5) and (p_6) ensure that the relations r_V and r_H are defined completely, i.e. no other relations except for those specified in the first column hold in models of $\mathcal{P}$. In particular, in every model of $\mathcal{P}$, relations r_H and r_V commute, and so, axiom α is not satisfied. Consequently, $\mathcal{P} \cup \mathcal{Q}$ is unsatisfiable, whereas $\mathcal{P} \cup \mathcal{Q}_1$ is satisfiable, and so, $\mathcal{Q}_1$ is not an $\mathbf{S}$-module in $\mathcal{Q}$. □

Corollary 13 *There exists no algorithm for performing any of the tasks T1-T3, T1s and T2s from* (3), (5) *and* (6) *for $\mathcal{ALC}$-ontologies.*

Proof. Theorem 12 implies directly that there is no algorithm for task T3 from (6), because otherwise, one can check if an axiom α is $\mathbf{S}$-essential in $\mathcal{Q}$ by simply computing the set of all essential axioms by this algorithm for T3 and then checking if α is contained in this set. The remaining tasks from (3) and (5) are unsolvable since they are reducible to T3 by Proposition 7 and by Proposition 5. □

Corollary 14 *Given a signature* $\mathbf{S}$, *an* $\mathcal{ALC}$*-ontology* $\mathcal{Q}$ *and an ontology* $\mathcal{Q}_1 \subseteq \mathcal{Q}$, *it is undecidable whether* $\mathcal{Q}_1$ *is an* $\mathbf{S}$*-module in* $\mathcal{Q}$ *w.r.t.* $\mathrm{L} = \mathcal{ALCO}$.

Proof. The procedure for deciding if $\mathcal{Q}_1$ is an $\mathbf{S}$-module in $\mathcal{Q}$ can be used for solving task T1, which is not possible by Corollary 13. Indeed, by enumerating the subsets of $\mathcal{Q}$ and checking if they are modules, one can compute all subsets $\mathcal{M}$ of $\mathcal{Q}$ that are $\mathbf{S}$-modules in $\mathcal{Q}$. The set of all minimal modules in $\mathcal{Q}$ can be then computed from $\mathcal{M}$ by filtering out those sets in $\mathcal{M}$ that are proper subsets of some other sets in $\mathcal{M}$. $\square$

Corollary 14 has a strong impact on the problem of knowledge reuse and forces us to revisit the original problem we aim at solving. As the problem of extracting minimal modules cannot be computationally solved for OWL DL in none of the forms T1-T3, T1s or T2s, we propose to relax some of the requirements in these tasks. We cannot drop the requirements that extracted fragments should be modules since, in this case, we have no guarantee for the correctness of the result. We can sacrifice, however, the minimality requirements for the computed modules and consider the following weakened version of the task T2:

$$\text{T2w.}\quad \text{compute } \textit{some} \text{ small enough } \mathbf{S}\text{-module in } \mathcal{Q} \tag{7}$$

Although it is always possible to extract an $\mathbf{S}$-module in $\mathcal{Q}$ (one can simply return $\mathcal{Q}$ which is always an $\mathbf{S}$-module in $\mathcal{Q}$), it still makes sense to develop, compare, and practically apply procedures that compute reasonably small modules. In the rest of the paper we describe two procedures of this form, based on the notions of locality, which we first introduced in [4]. The modules we obtain might be larger than the minimal modules and therefore we need to show that, in practice, they are still reasonably small.

4 Modules Based on Locality

In this section, we formulate the notion of locality, first introduced in [4] which will constitute the basis of our algorithm for extracting modules.

4.1 Locality

As a consequence of Case 2 in Proposition 10, model conservative extensions can be used as a sufficient condition for the notion of module. It is not possible, however, to design a procedure that extracts modules based on this condition since the problem of deciding model conservative extensions is highly undecidable [11]. The idea underlying this notion, however, can be used to establish sufficient conditions for the notion of module which are decidable and can be used in practice.

Consider the first part of Example 9, where we show that the set $\mathcal{Q}$ of axioms M1-M5 in Figure 1 is a model $\mathbf{S}$-conservative extension of $\mathcal{Q}_1 = \{\mathrm{M1}, \ldots, \mathrm{M4}\}$, for $\mathbf{S} = \{\mathsf{Cystic\_Fibrosis}, \mathsf{Genetic\_Disorder}\}$. In this example, the model conservative extension was shown by finding expansions of $\mathsf{Sig}(\mathcal{Q}_1)$-interpretations to models of $\mathcal{Q}$ in which all concept and atomic roles not in $\mathsf{Sig}(\mathcal{Q}_1)$ were interpreted as the empty set. One could consider the cases where conservative extensions (and hence modules) can be determined in this manner. This idea can be formalized using the notion of locality:

Definition 15 (Locality [4]). Let $\mathbf{S}$ be a signature. We say that an *axiom α is local w.r.t.* $\mathbf{S}$ if every trivial expansion of any $\mathbf{S}$-interpretation to $\mathbf{S} \cup \mathsf{Sig}(\alpha)$ is a model of α. We denote by $\mathsf{local}(\mathbf{S})$ the collection of all axioms that are local w.r.t. $\mathbf{S}$. An ontology $\mathcal{O}$ is local w.r.t. $\mathbf{S}$ if $\mathcal{O} \subseteq \mathsf{local}(\mathbf{S})$. $\diamondsuit$

Intuitively, an ontology $\mathcal{O}$ is *local* w.r.t. a signature $\mathbf{S}$ if we can take *any* interpretation for the symbols in $\mathbf{S}$ and extend it to a *model* of $\mathcal{O}$ that interprets the additional symbols as the empty set.

Example 16 Consider axiom M5 from Figure 1. This axiom is local w.r.t. $\mathbf{S} = \{\mathsf{Cystic\_Fibrosis}, \mathsf{Genetic\_Disorder}\}$. Indeed, as shown in Example 9, for every trivial expansion $\mathcal{I}$ of an $\mathbf{S}$-interpretation to $\mathbf{S} \cup \mathsf{Sig}(\alpha)$, the concept $\mathsf{DEFBI\_Gene}$ is interpreted as the empty set, and so, $\mathcal{I}$ satisfies M5.

On the other hand, M5 is not local w.r.t. $\mathbf{S} = \{\mathsf{DEFBI\_Gene}\}$. Indeed, take any $\mathbf{S}$-interpretation $\mathcal{I}_1$ in which $\mathsf{DEFBI\_Gene}$ is interpreted as a non-empty set. Then, for every trivial expansion $\mathcal{I}$ of $\mathcal{I}_1$, the concept on the left-hand-side of M5 is always interpreted as a non-empty set, whereas the concept on the right-hand-side is always interpreted as the empty set. So $\mathcal{I}$ does not satisfy α.

In fact, this shows that axiom M5 is local w.r.t. $\mathbf{S}$ if and only if $\mathbf{S}$ does not contain $\mathsf{DEFBI\_Gene}$. $\diamondsuit$

The following is a simple but useful property of locality shows that the set of local axioms can only become smaller if the signature expands:

Lemma 17 [Anti-Monotonicity of Locality] *Let $\mathbf{S}_1$ and $\mathbf{S}_2$ be signature sets. Then $\mathbf{S}_1 \subseteq \mathbf{S}_2$ implies $\mathsf{local}(\mathbf{S}_2) \subseteq \mathsf{local}(\mathbf{S}_1)$.*

Proof. Let $\alpha \in \mathsf{local}(\mathbf{S}_2)$. We demonstrate that $\alpha \in \mathsf{local}(\mathbf{S}_1)$. For this purpose, let $\mathcal{I}_1$ be an arbitrary $\mathbf{S}_1$-interpretation. We need to show that every trivial expansion $\mathcal{I}_1'$ of $\mathcal{I}_1$ to $\mathbf{S}_1 \cup \mathsf{Sig}(\alpha)$ is a model of α.

Let $\mathcal{I}_2$ be a trivial expansion of $\mathcal{I}_1$ to $\mathbf{S}_2$ (note that $\mathbf{S}_1 \subseteq \mathbf{S}_2$). Since $\alpha \in \mathsf{local}(\mathbf{S}_2)$, every trivial expansion $\mathcal{I}_2'$ of $\mathcal{I}_2$ to $\mathbf{S}_2 \cup \mathsf{Sig}(\alpha)$ is a model of α. Note that $\mathcal{I}_2'$ is a trivial expansion of $\mathcal{I}_1$ to $\mathbf{S}_2 \cup \mathsf{Sig}(\alpha)$, hence $\mathcal{I}_1' = \mathcal{I}_2'|_{\mathbf{S}_1 \cup \mathsf{Sig}(\alpha)} \models \alpha$. $\square$

Locality can be used to formulate a sufficient condition for an ontology to be a model conservative extension of another ontology:

Proposition 18 [Locality $\Rightarrow$ Model Conservativity] *Let $\mathcal{O}_1$, $\mathcal{O}_2$ be two ontologies and $\mathbf{S}$ a signature such that $\mathcal{O}_2$ is local w.r.t. $\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_1)$. Then $\mathcal{O}_1 \cup \mathcal{O}_2$ is an $\mathbf{S}$-model conservative extension of $\mathcal{O}_1$.*

Proof. Let $\mathcal{I}_1$ be a model of $\mathcal{O}_1$. We show that there exists a model $\mathcal{I}$ of $\mathcal{O}_1 \cup \mathcal{O}_2$ such that $\mathcal{I}|_{\mathbf{S}} = \mathcal{I}_1|_{\mathbf{S}}$.

Let $\mathcal{I}$ be a trivial expansion of $\mathcal{I}_1|_{\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_1)}$ to $\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_1) \cup \mathsf{Sig}(\mathcal{O}_2)$, thus, in particular, $\mathcal{I}|_{\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_1)} = \mathcal{I}_1|_{\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_1)}$. We need to show that $\mathcal{I}$ is a model of $\mathcal{O}_1 \cup \mathcal{O}_2$. Since $\mathcal{O}_2$ is local w.r.t. $\mathbf{S} \cup \mathsf{Sig}(\mathcal{O}_1)$, by Definition 15, $\mathcal{I}$ is a model of $\mathcal{O}_2$. Moreover, since $\mathcal{I}|_{\mathsf{Sig}(\mathcal{O}_1)} = \mathcal{I}_1|_{\mathsf{Sig}(\mathcal{O}_1)}$ and $\mathcal{I}_1 \models \mathcal{O}_1$, we have $\mathcal{I} \models \mathcal{O}_1$. Hence, $\mathcal{I} \models \mathcal{O}_1 \cup \mathcal{O}_2$ what was required to show. $\square$

Using Proposition 18 and Property 2 of Proposition 10 we obtain:

Corollary 19 *Let $\mathcal{O}_1$, $\mathcal{O}_2$ and $\mathbf{S}$ be as given in Proposition 18. Then $\mathcal{O}_1$ is an $\mathbf{S}$-module in $\mathcal{O}_1 \cup \mathcal{O}_2$.*

Corollary 19 suggests how one can use locality for extracting modules. Given an ontology $\mathcal{Q}$ and a signature $\mathbf{S}$, it is sufficient to partition $\mathcal{Q}$ into $\mathcal{Q}_1 \cup \mathcal{Q}_2$ such that $\mathcal{Q}_2$ is local w.r.t. $\mathbf{S} \cup \mathsf{Sig}(\mathcal{Q}_1)$. In this case, $\mathcal{Q}_1$ is an $\mathbf{S}$-module in $\mathcal{Q}$.

Definition 20 (Modules based on Locality Condition).
Given an ontology $\mathcal{Q}$ and a signature $\mathbf{S}$, we say that $\mathcal{Q}_1 \subseteq \mathcal{Q}$ *is a locality-based* $\mathbf{S}$-*module in* $\mathcal{Q}$ if $\mathcal{Q} \setminus \mathcal{Q}_1$ is local w.r.t $\mathbf{S} \cup \mathsf{Sig}(\mathcal{Q}_1)$. $\Diamond$

Remark 21 Note from Definition 20 that every locality-based $\mathbf{S}$-module $\mathcal{Q}_1$ in $\mathcal{Q}$, is also a locality-based $\mathbf{S} \cup \mathsf{Sig}(\mathcal{Q}_1)$-module in $\mathcal{Q}$. $\Diamond$

Remark 22 Note that $\mathcal{Q}_1$ is a locality-based $\mathbf{S}$-module in $\mathcal{Q}$ if every trivial expansion of every model of $\mathcal{Q}_1$ based on $\mathbf{S} \cup \mathsf{Sig}(\mathcal{Q}_1)$ to $\mathbf{S} \cup \mathsf{Sig}(\mathcal{Q})$, is a model for $\mathcal{Q}$. $\Diamond$

Example 23 [Example 16, continued] We have seen in Example 16 that axiom M5 is local w.r.t. every $\mathbf{S}$ that does not contain the atomic concept $\mathsf{DEFBI\_Gene}$. In particular, for $\mathcal{Q}_1$ consisting of axioms M1-M4 from Figure 1, M5 is local w.r.t. $\mathsf{Sig}(\mathcal{Q}_1)$. Hence, according to Definition 20, $\mathcal{Q}_1$ is a locality-based $\mathbf{S}$-module in $\mathcal{Q} = \{\mathrm{M1}, \ldots, \mathrm{M5}\}$ for every $\mathbf{S} \subseteq \mathsf{Sig}(\mathcal{Q}_1)$. $\Diamond$

Remark 24 Note that the analog of the Part 1 in Proposition 2 does not hold for locality-based modules since locality-based modules are not necessarily upward-closed. For example, consider the following ontology and a signature:

$$\mathcal{Q} = \{(1)\ A_1 \sqsubseteq A_2;\ (2)\ B \sqsubseteq A_1;\ (3)\ B \sqsubseteq A_2\} \quad \mathbf{S} = \{A_1,\ A_2\}$$

It is easy to see that the set $\mathcal{Q}_1 = \{A_1 \sqsubseteq A_2\}$ consisting of the first axiom from $\mathcal{Q}$ is a locality-based $\mathbf{S}$-module in $\mathcal{Q}$, since both axioms (2) and (3) are local w.r.t. $\mathbf{S} \cup \mathsf{Sig}(\mathcal{Q}_1) = \{A_1,\ A_2\}$. However, its superset $\mathcal{Q}_1' = \{A_1 \sqsubseteq A_2;\ B \sqsubseteq A_1\}$ is not a locality-based module w.r.t. $\mathbf{S}$, since the axiom $B \sqsubseteq A_2$ in $\mathcal{Q} \setminus \mathcal{Q}_1'$ is not local w.r.t. $\mathbf{S} \cup \mathsf{Sig}(\mathcal{Q}_1') = \{A_1,\ A_2,\ B\}$. Note that $\mathcal{Q}_1'$ is an $\mathbf{S}$-module in $\mathcal{Q}$, since it is a superset of an $\mathbf{S}$-module $\mathcal{Q}_1$. $\Diamond$

We introduce a special notion to capture the modules that are supersets of the locality-based modules:

Definition 25 (Locality-Induced Modules).
We say that a subset $\mathcal{Q}_2 \subseteq \mathcal{Q}$ is a *locality-induced* $\mathbf{S}$-*module* in $\mathcal{Q}$ if there exists a locality-based $\mathbf{S}$-module $\mathcal{Q}_1$ in $\mathcal{Q}$ such that $\mathcal{Q}_1 \subseteq \mathcal{Q}_2$. $\Diamond$

4.2 Testing Locality

As demonstrated in Example 16, for testing locality of an axiom α w.r.t. $\mathbf{S}$, it is sufficient to interpret every atomic concept and atomic role not in $\mathbf{S}$ with the empty set and then check if α is satisfied for all interpretations of the remaining symbols. This observation suggests that locality can be tested by first simplifying the ontology by eliminating atomic roles and concepts that are not in $\mathbf{S}$, and then checking if the resulting axioms are satisfied in every interpretation for the remaining symbols. This idea is formalized as follows:

Proposition 26 [Testing Locality] *Let $\mathcal{O}$ be a $\mathcal{SHOIQ}$ ontology and $\mathbf{S}$ a signature. Let $\mathcal{O}_\mathbf{S}$ be obtained from $\mathcal{O}$ by applying the transformations below, where every A is an atomic concept, every r is an atomic role with $A, r \notin \mathbf{S}$, and every R is a role r or r^- with $r \notin \mathbf{S}$: (1) replace all concepts of form A, $\exists R.C$ or $(\geqslant n\, R.C)$ with $\bot$; (2) remove every transitivity axiom $\mathsf{Trans}(r)$; (3) replace every assertion $a : A$ and $r(a, b)$ with the contradiction axiom $\top \sqsubseteq \bot$.*

Then $\mathcal{O}$ is local w.r.t. $\mathbf{S}$ iff every axiom in $\mathcal{O}_\mathbf{S}$ is a tautology.

Proof. It is easy to check that the transformation above preserves the satisfaction of axioms under every trivial expansion $\mathcal{I}$ of every $\mathbf{S}$-interpretation to $\mathbf{S} \cup \mathsf{Sig}(\mathcal{O})$. Hence, the resulting ontology $\mathcal{O}_\mathbf{S}$ is local w.r.t. $\mathbf{S}$ iff the original ontology $\mathcal{O}$ was local w.r.t. $\mathbf{S}$. Moreover, it is easy to see that there are no atomic concepts and atomic roles outside $\mathbf{S}$ left in $\mathcal{O}_{\mathbf{S}}$ after the transformation. Hence, every axiom α from $\mathcal{O}_{\mathbf{S}}$ is a tautology iff $\mathcal{Q}$ is local w.r.t. $\mathbf{S}$. $\square$

Note that according to Definition 15, assertions $a : A$ and $r(a, b)$ can never be local since they can only be satisfied by interpretations that interpret A and r as non-empty sets. Hence, assertions must be included in every locality-based module, which is reflected in the step (3) of the transformation in Proposition 26.

Example 27 Recall that in Example 16 we have demonstrated that axiom M5 from Figure 1 is local w.r.t. $\mathbf{S} = \{\mathsf{Cystic\_Fibrosis}, \mathsf{Genetic\_Disorder}\}$. Now we demonstrate this using Proposition 26. Indeed, according to this proposition we need to perform the following replacements:

$$\begin{aligned}\mathsf{DEFBI\_Gene} &\Rightarrow \bot \text{ (\textit{by (1) since} } \mathsf{DEFBI\_Gene} \notin \mathbf{S}\text{)}\\ \mathsf{Immuno\_Protein\_Gene} &\Rightarrow \bot \text{ (\textit{by (1) since} } \mathsf{Immuno\_Protein\_Gene} \notin \mathbf{S}\text{)}\\ \exists \mathsf{associated\_With}.\mathsf{Cystic\_Fibrosis} &\Rightarrow \bot \text{ (\textit{by (1) since} } \mathsf{associated\_With} \notin \mathbf{S}\text{)}\end{aligned}$$

Hence, axiom M5 will be translated to axiom $\bot \sqsubseteq \bot \sqcap \bot$ which is a tautology. $\diamondsuit$

An important conclusion of Proposition 26 is that one can use the standard capabilities of available DL-reasoners[2] such as FaCT++ [21], RACER [12], Pellet [19] or KAON2 [13] for testing locality since these reasoners can test for DL-tautologies. Checking for tautologies in description logics is, theoretically, a difficult problem (e.g. for DL $\mathcal{SHOIQ}$ is NEXPTIME-complete). There are, however, several reasons to believe that the locality test would perform well in practice. First, and most importantly, the size of the axioms in an ontology is usually small compared to the size of the ontology. Second, DL reasoners are highly optimized for standard reasoning tasks and behave well for most realistic ontologies.

In case this is too costly, it is possible to formulate a tractable approximation to the locality conditions for $\mathcal{SHOIQ}$:

Definition 28 (Syntactic Locality for $\mathcal{SHOIQ}$). Let $\mathbf{S}$ be a signature. The following grammar recursively defines two sets of concepts $\mathcal{C}_{\mathbf{S}}^{\bot}$ and $\mathcal{C}_{\mathbf{S}}^{\top}$ for a signature $\mathbf{S}$:

$$\begin{aligned}\mathcal{C}_{\mathbf{S}}^{\bot} &::= A^{\bot} \mid (\neg C^{\top}) \mid (C \sqcap C^{\bot}) \mid (\exists R^{\bot}.C)\\ &\quad\mid (\exists R.C^{\bot}) \mid (\geqslant n\, R^{\bot}.C) \mid (\geqslant n\, R.C^{\bot})\,.\\ \mathcal{C}_{\mathbf{S}}^{\top} &::= (\neg C^{\bot}) \mid (C_1^{\top} \sqcap C_2^{\top})\,.\end{aligned}$$

[2]See http://www.cs.man.ac.uk/~sattler/reasoners.html for a list of currently available reasoners.

where $A^\perp \notin \mathbf{S}$ is a atomic concept, R is a role, and C is a concept, $C^\perp \in \mathcal{C}^\perp_\mathbf{S}$, $C^\top_{(i)} \in \mathcal{C}^\top_\mathbf{S}$, $i = 1, 2$, and $R^\perp \notin \mathsf{Rol}(\mathbf{S})$ is a role.

An axiom α is *syntactically local w.r.t.* $\mathbf{S}$ if it is of one of the following forms: (1) $R^\perp \sqsubseteq R$, or (2) $\mathsf{Trans}(R^\perp)$, or (3) $C^\perp \sqsubseteq C$ or (4) $C \sqsubseteq C^\top$. We denote by $\mathsf{s\_local}(\mathbf{S})$ 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 $\mathcal{O} \subseteq \mathsf{s\_local}(\mathbf{S})$. $\Diamond$

Intuitively, every concept in $\mathcal{C}^\perp_\mathbf{S}$ becomes equivalent to $\perp$ if we replace every symbol $A^\perp$ or $R^\perp$ not in $\mathbf{S}$ with the bottom concept $\perp$ and the empty role respectively, which are both interpreted as the empty set under every interpretation. Similarly, the concepts from $\mathcal{C}^\top_\mathbf{S}$ are equivalent to $\top$ under this replacement. Syntactically local axioms become tautologies after these replacements.

For example, it is easy to show that the axiom M2 from Figure 1 is local w.r.t. $\mathbf{S} = \{\mathsf{Fibrosis}, \mathsf{has\_Origin}\}$: if we replace the remaining symbols in this axiom with $\perp$, we obtain a tautology $\perp \equiv \perp$:

$$\overbrace{\mathsf{Genetic\_Fibrosis}}^{\perp} \equiv \underline{\mathsf{Fibrosis}} \sqcap \underbrace{\exists \underline{\mathsf{has\_Origin}}.\overbrace{\mathsf{Genetic\_Origin}}^{\perp}}_{\perp}$$

To distinguish the original notion of locality from its syntactic approximation, we sometimes call the first as *semantic locality*, as it is defined in terms of the interpretations.

It is easy to show that the analog of Lemma 17 also holds for syntactic locality:

Lemma 29 [Anti-Monotonicity of Syntactic Locality]
Let $\mathbf{S}_1$ *and* $\mathbf{S}_2$ *be signature sets. Then* $\mathbf{S}_1 \subseteq \mathbf{S}_2$ *implies* $\mathsf{s\_local}(\mathbf{S}_2) \subseteq \mathsf{s\_local}(\mathbf{S}_1)$.

Proof. It is easy to see from Definition 28 that $\mathcal{C}^\perp_{\mathbf{S}_2} \sqsubseteq \mathcal{C}^\perp_{\mathbf{S}_1}$, $\mathcal{C}^\top_{\mathbf{S}_2} \sqsubseteq \mathcal{C}^\top_{\mathbf{S}_1}$, and hence, $\mathsf{s\_local}(\mathbf{S}_2) \sqsubseteq \mathsf{s\_local}(\mathbf{S}_1)$. $\Box$

As expected, syntactic locality is an approximation for semantic locality:

Proposition 30 *Let* $\mathbf{S}$ *be a signature. Then* $\mathsf{s\_local}(\mathbf{S}) \subseteq \mathsf{local}(\mathbf{S})$.

Proof. Let α be an axiom that is syntactically local w.r.t. $\mathbf{S}$ and let $\mathcal{I} = (\Delta, \cdot^\mathcal{I})$ be a trivial expansion of some $\mathbf{S}$-interpretation to $\mathbf{S} \cup \mathsf{Sig}(\alpha)$. We have to demonstrate that $\mathcal{I}$ is a model of α. By induction over the definitions of $\mathcal{C}^\perp_\mathbf{S}$ and $\mathcal{C}^\top_\mathbf{S}$ from Definition 28, it is easy to show that: (i) every role $R \notin \mathsf{Rol}(\mathbf{S})$ and every every concept from $\mathcal{C}^\perp_\mathbf{S}$ is interpreted in $\mathcal{I}$ with the empty set, and (ii) every concept from $\mathcal{C}^\top_\mathbf{S}$ is interpreted in $\mathcal{I}$ with Δ. By checking all the possible cases for a syntactically local axiom α in Definition 20, it is easy to see that in every of these cases $\mathcal{I}$ is a model of α. $\Box$

Algorithm 1 extract_module($\mathcal{Q}, \mathbf{S}$)

```
Input:
    $\mathcal{Q}$: ontology
    $\mathbf{S}$: signature
Output:
    $\mathcal{Q}_1$: a locality-based $\mathbf{S}$-module in $\mathcal{Q}$

1:  $\mathcal{Q}_1 \leftarrow \emptyset \quad \mathcal{Q}_2 \leftarrow \mathcal{Q}$
2:  while not empty($\mathcal{Q}_2$) do
3:      $\alpha \leftarrow$ select_axiom($\mathcal{Q}_2$)
4:      if locality_test( $\alpha$, $\mathbf{S} \cup \mathsf{Sig}(\mathcal{Q}_1)$ ) then
5:          $\mathcal{Q}_2 \leftarrow \mathcal{Q}_2 \setminus \{\alpha\}$                    ▷ $\alpha$ is processed
6:      else
7:          $\mathcal{Q}_1 \leftarrow \mathcal{Q}_1 \cup \{\alpha\}$                    ▷ move $\alpha$ into $\mathcal{Q}_1$
8:          $\mathcal{Q}_2 \leftarrow \mathcal{Q} \setminus \mathcal{Q}_1$        ▷ reset $\mathcal{Q}_2$ to the complement of $\mathcal{Q}_1$
9:      end if
10: end while
11: return $\mathcal{Q}_1$
```

The converse of Proposition 30 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 a tautology and thus is local w.r.t. every $\mathbf{S}$. This axiom, however, is not syntactically local w.r.t. $\mathbf{S} = \{A, B\}$ since it involves symbols in $\mathbf{S}$ only. Another example, which is not a tautology, is the GCI $\alpha = (\exists R.\neg A \sqsubseteq \exists R.\neg B)$, which is semantically local w.r.t. $\mathbf{S} = \{R\}$ ($\exists R.\top \sqsubseteq \exists R.\top$ is a tautology), but not syntactically local. Thus, the limitation of syntactic locality is its inability to perform reasoning elements from $\mathbf{S}$.

We distinguish the notion of modules based on these two locality conditions as *semantic locality-based modules* and *syntactic locality-based modules*.

Corollary 31 *If $\mathcal{Q}_1$ is a syntactic locality-based $\mathbf{S}$-module in $\mathcal{Q}$, then $\mathcal{Q}_1$ is a semantic locality-based $\mathbf{S}$-module in $\mathcal{Q}$.*

For the reference and for the convenience of the reader, we illustrate in Figure 3 the relationships between the key theoretical results of this paper.

4.3 Computing Locality-Based Modules

Recall that, according to Definition 20, in order to construct a locality-based $\mathbf{S}$-module in an ontology $\mathcal{Q}$, it suffices to partition the ontology $\mathcal{Q}$ as $\mathcal{Q} = \mathcal{Q}_1 \cup \mathcal{Q}_2$ such that $\mathcal{Q}_2$ is local w.r.t. $\mathbf{S} \cup \mathsf{Sig}(\mathcal{Q}_1)$. Algorithm 1 outlines a simple procedure

$$\begin{tikzcd}
 & \mathcal{Q}_1 \text{ is a syntactical locality-based } \mathbf{S}\text{-module in } \mathcal{Q} \arrow[d, "\textit{(Corollary 31)}"] & \\
 & \mathcal{Q}_1 \text{ is a locality-based } \mathbf{S}\text{-module in } \mathcal{Q} \arrow[dl, "\textit{(Proposition 37)}"'] \arrow[dr, "\textit{(Proposition 18)}"] & \\
\mathcal{Q}_1 \text{ contains all } \mathbf{S}\text{-essential axioms w.r.t. L in } \mathcal{Q} \arrow[dr, "\textit{(Definition 6)}"'] & & \mathcal{Q} \text{ is a model } \mathbf{S}\text{-conservative extension of } \mathcal{Q}_1 \arrow[dl, "\textit{(Proposition 10, part 2)}"] \\
 & \boldsymbol{\mathcal{Q}_1} \textbf{ is an } \mathbf{S}\textbf{-module in } \boldsymbol{\mathcal{Q}} \textbf{ w.r.t. L} \arrow[d, "\textit{(Proposition 10, part 1)}"] & \\
 & \mathcal{Q} \text{ is a deductive } \mathbf{S}\text{-conservative extension of } \mathcal{Q}_1 \text{ w.r.t. L} &
\end{tikzcd}$$

Figure 3: Summary for the main theoretical results of the paper

which performs this task. Given an effective locality test locality_test$(\alpha, \mathbf{S})$ (which uses either a reasoner or the syntactical approximation) which returns true only if the axiom α is local w.r.t. $\mathbf{S}$, the algorithm first initializes the partition to the trivial one: $\mathcal{Q}_1 = \emptyset$ and $\mathcal{Q}_2 = \mathcal{Q}$, and then repeatedly moves to $\mathcal{Q}_1$ those axioms from $\mathcal{Q}_2$ that are not local w.r.t. $\mathbf{S} \cup \mathsf{Sig}(\mathcal{Q}_1)$ until no such axioms are left in $\mathcal{Q}_2$.

In Table 2 we provide a trace of Algorithm 1 for the input $(\mathcal{Q}, \mathbf{S})$, where $\mathcal{Q}$ is an ontology consisting of the axioms M1-M5 from Figure 1 and $\mathbf{S}$ is a signature $\mathbf{S} = \{\mathsf{Cystic\_Fibrosis}, \mathsf{Genetic\_Disorder}\}$. Each row in the table corresponds to an iteration of the while loop in Algorithm 1. The last column of the table provides the results of the locality test in line 4. Note that the syntactic locality condition was sufficient in all tests: all axioms that were semantically non-local were also syntactically non-local.

Proposition 32 [Correctness of Algorithm 1]
For every input $\mathcal{Q}$ and $\mathbf{S}$, Algorithm 1 computes a locality-based $\mathbf{S}$-module in $\mathcal{Q}$.

Proof. We have to show that (1) Algorithm 1 terminates for every input $\mathcal{T}$ and $\mathbf{S}$, and (2) the output extract_module$(\mathbf{S}, \mathcal{Q})$ is a locality-based $\mathbf{S}$-module in $\mathcal{Q}$.

(1) Termination of the algorithm follows from the fact that in every iteration of the while loop either the size of $\mathcal{Q}_1$ increases, or the size of $\mathcal{Q}_1$ remains the same but the size of $\mathcal{Q}_2$ decreases. Note that this means that Algorithm 1 terminates in quadratic time in the number of axioms in $\mathcal{Q}$, assuming constant time locality test.

(2) It is easy to observe that every axiom α that is neither in $\mathcal{Q}_1$ nor in $\mathcal{Q}_2$ is

	$\mathcal{Q}_1$	$\mathcal{Q}_2$	New elements in $\mathbf{S} \cup \mathsf{Sig}(\mathcal{Q}_1)$	α	local?
1	–	$\text{M1} - \text{M5}$	Cystic_Fibrosis, Genetic_Disorder	M1	No
2	M1	$\text{M2} - \text{M5}$	Fibrosis, located_In, Pancreas, has_Origin, Genetic_Origin	M2	No
3	$\text{M1}, \text{M2}$	$\text{M3} - \text{M5}$	Genetic_Fibrosis	M3	No
4	$\text{M1} - \text{M3}$	$\text{M4}, \text{M5}$	–	M4	No
5	$\text{M1} - \text{M4}$	M5	–	M5	Yes
6	$\text{M1} - \text{M4}$	–	–	–	

Table 2: A trace of Algorithm 1 for the input $\mathcal{Q} = \{\text{M1}, \dots, \text{M5}\}$ and $\mathbf{S} = \{\mathsf{Cystic\_Fibrosis},\ \mathsf{Genetic\_Disorder}\}$

local w.r.t. $\mathbf{S} \cup \mathsf{Sig}(\mathcal{Q}_1)$, since the only way such an α can appear is at the line 3 of the algorithm, and α remains in $\mathcal{Q} \setminus (\mathcal{Q}_1 \cup \mathcal{Q}_2)$ only if $\mathbf{S} \cup \mathsf{Sig}(\mathcal{Q}_1)$ does not change. □

Note that there is an implicit non-determinism in Algorithm 1, namely, in line 3 in which an axiom from $\mathcal{Q}_2$ is selected. It might well be the case that several choices for α are possible at this moment. For example, the trace in Table 3 makes a different choice for α from $\mathcal{Q}_2$ than the trace in Table 2. In the first iteration of the while loop, we select $\alpha = \text{M2}$ from $\mathcal{Q}_2$ instead of M1 as in Table 2. This has resulted in a longer trace yet with the same result $\mathcal{Q}_1 = \{\text{M1}, \dots, \text{M4}\}$. Note that axioms M2 and M3 are selected several times and produce different results for the locality tests, since $\mathcal{Q}_1$ has been modified. This demonstrates the reason why we reset $\mathcal{Q}_2$ to $\mathcal{Q} \setminus \mathcal{Q}_2$ at the line 8 of Algorithm 1, namely, not to miss axioms that has been checked to be local w.r.t. old $\mathcal{Q}_1$, but are no longer local w.r.t. new $\mathcal{Q}_1$.

As we have seen from the traces in Table 2 and Table 3, Algorithm 1 has produced the same output despite the fact that different choices for α has been made inside the while loop. One might wonder if this is always the case. It turns out that the choices for α indeed do not have any impact on the result of Algorithm 1, provided that the locality test satisfy some rather natural requirements:

Definition 33. We say that a locality test locality_test($\alpha, \mathbf{S}$) is *anti-monotonic* if for every $\mathbf{S}_1 \subseteq \mathbf{S}_2$, whenever locality_test($\alpha, \mathbf{S}_2$) succeeds then locality_test($\alpha, \mathbf{S}_1$) succeeds as well.

We say that *locality of $\mathcal{Q}_1$ w.r.t. $\mathbf{S}$ in $\mathcal{Q}_1$ is provable using locality_test($\alpha, \mathbf{S}$)* if for every $\alpha \in \mathcal{Q} \setminus \mathcal{Q}_1$, we have that locality_test($\alpha, \mathbf{S} \cup \mathsf{Sig}(\mathbf{S}_1)$) succeeds. ◊

	$\mathcal{Q}_1$	$\mathcal{Q}_2$	New elements in $\mathbf{S} \cup \mathsf{Sig}(\mathcal{Q}_1)$	α	loc.
1	–	$\mathrm{M1} - \mathrm{M5}$	Cystic_Fibrosis, Genetic_Disorder	M2	Yes
2	–	$\mathrm{M1}, \mathrm{M3} - \mathrm{M5}$	–	M3	Yes
3	–	$\mathrm{M1}, \mathrm{M4}, \mathrm{M5}$	–	M1	No
4	M1	$\mathrm{M2} - \mathrm{M5}$	Fibrosis, located_In, Pancreas, has_Origin, Genetic_Origin	M3	No
5	$\mathrm{M1}, \mathrm{M3}$	$\mathrm{M2}, \mathrm{M4}, \mathrm{M5}$	Genetic_Fibrosis	M4	No
6	$\mathrm{M1}, \mathrm{M3}, \mathrm{M4}$	$\mathrm{M2}, \mathrm{M5}$	–	M5	Yes
7	$\mathrm{M1}, \mathrm{M3}, \mathrm{M4}$	M2	–	M2	No
8	$\mathrm{M1} - \mathrm{M4}$	M5	–	M5	Yes
9	$\mathrm{M1} - \mathrm{M4}$	–	–	–	

Table 3: An alternative trace of Algorithm 1 for the input $\mathcal{Q} = \{\mathrm{M1}, \ldots, \mathrm{M5}\}$ and $\mathbf{S} = \{\mathsf{Cystic\_Fibrosis}, \mathsf{Genetic\_Disorder}\}$

Proposition 34 [Determinism of Algorithm 1]
The output of Algorithm 1 based on anti-monotonic locality_test$(\alpha, \mathbf{S})$ is the smallest $\mathcal{Q}_1$ such that locality of $\mathcal{Q}_1$ w.r.t. $\mathbf{S}$ is provable using locality_test$(\alpha, \mathbf{S})$.

Proof. It is easy to see (see the proof of Proposition 32) that the locality of every output $\mathcal{Q}_1$ of Algorithm 1 is provable using locality_test$(\alpha, \mathbf{S})$. It remains, thus, to show that for every subset $\mathcal{Q}'_1 \subseteq \mathcal{Q}$ such that locality of $\mathcal{Q}'_1$ w.r.t. $\mathbf{S}$ in $\mathcal{Q}$ is provable using locality_test$(\alpha, \mathbf{S})$, we have $\mathcal{Q}_1 \subseteq \mathcal{Q}'_1$.

Assume, to the contrary, that for some run of the algorithm, the output $\mathcal{Q}_1$ is not a subset of $\mathcal{Q}'_1$. Since the initial $\mathcal{Q}_1 = \emptyset$ was a subset of $\mathcal{Q}'_1$, there is a moment in the computation such that $\mathcal{Q}_1$ was a subset of $\mathcal{Q}'_1$, but $\mathcal{Q}_1 \cup \{\alpha\}$ is no longer a subset of $\mathcal{Q}'_1$. For these particular values of $\mathcal{Q}_1$ and α we have: (i) $\mathcal{Q}_1 \subseteq \mathcal{Q}'_1$, (ii) $\alpha \in \mathcal{Q} \setminus \mathcal{Q}'_1$, and (iii) locality_test$(\alpha, \mathbf{S} \cup \mathsf{Sig}(\mathcal{Q}_1))$ fails. From (ii) by property of $\mathcal{Q}'_1$ we have locality_test$(\alpha, \mathbf{S} \cup \mathsf{Sig}(\mathcal{Q}'_1))$ succeeds, which implies using (i) and anti-monotonicity of locality_test that locality_test$(\alpha, \mathbf{S} \cup \mathsf{Sig}(\mathcal{Q}_1))$ succeeds which contradicts to (iii). This proves that $\mathcal{Q}_1$ is indeed a subset of $\mathcal{Q}'_1$. □

Corollary 35 [Uniqueness of a Minimal Locality-Based Module]
Algorithm 1 using a test based on the semantic locality produces a unique minimal locality-based $\mathbf{S}$-module in $\mathcal{Q}$.

Proof. By Lemma 17 the semantic locality admits anti-monotonicity. □

Corollary 36 [Uniqueness of a Minimal Syntactic Locality-Based Module] *Algorithm 1 using a test based on the syntactic locality produces a unique minimal syntactic locality-based* $\mathbf{S}$*-module in* $\mathcal{Q}$*.*

Proof. By Lemma 29 the syntactic locality admits anti-monotonicity. □

4.4 Properties of Locality-based Modules

In this section, we outline some interesting properties of locality-based modules which make it possible to use them for applications other than knowledge reuse.

Let $\mathcal{Q}^{loc}_{\mathbf{S}}$ be the smallest locality-based $\mathbf{S}$-module in $\mathcal{Q}$, which is unique by Corollary 35 and is the output of Algorithm 1 for $\mathcal{Q}$ and $\mathbf{S}$. The first property is a direct consequence of Corollary 35:

Proposition 37 $\mathcal{Q}^{loc}_{\mathbf{S}}$ *contains all* $\mathbf{S}$*-essential axioms in* $\mathcal{Q}$ *w.r.t. every language* L *with Tarski-style set-theoretic semantics.*

Proof. Let $\mathcal{Q}_1$ be a minimal $\mathbf{S}$-module in $\mathcal{Q}$. We need to show that $\mathcal{Q}_1 \subseteq \mathcal{Q}^{loc}_{\mathbf{S}}$. Since (i) $\mathcal{Q}_1$ is a subset of a locality-based $\mathbf{S}$-module in $\mathcal{Q}$ (say, of $\mathcal{Q}$ itself) and (ii) there is no proper subset of $\mathcal{Q}_1$ that is a locality-based $\mathbf{S}$-module in $\mathcal{Q}$, we have that $\mathcal{Q}_1$ is a subset of a minimal locality-based $\mathbf{S}$-module in $\mathcal{Q}$. Since such a module is unique by Corollary 35, and it is $\mathcal{Q}^{loc}_{\mathbf{S}}$, we have that $\mathcal{Q}_1 \subseteq \mathcal{Q}^{loc}_{\mathbf{S}}$. □

As shown in Table 2 and Table 3, the minimal locality-based $\mathbf{S}$-module extracted from $\mathcal{Q}$ contains all $\mathbf{S}$-essential axioms M1–M4. In our case, the module contains only essential axioms; in general, however, locality-based modules might contain non-essential axioms; otherwise, they would provide a solution for our task T3 in (6).

Proposition 38 *Let* $\mathcal{Q}$ *be ontology,* A *and* B *atomic concepts and* $\mathbf{S}_{(i)}$ *a signature. Then:*

1. $\mathbf{S}_1 \subseteq \mathbf{S}_2$ *implies* $\mathcal{Q}^{loc}_{\mathbf{S}_1} \subseteq \mathcal{Q}^{loc}_{\mathbf{S}_2}$ *(monotonicity);*
2. $\mathcal{Q} \models (A \sqsubseteq B)$ *iff* $\mathcal{Q}^{loc}_{\{A\}} \models (A \sqsubseteq B)$.
3. $\mathcal{Q} \models (A \sqsubseteq B)$ *implies* $\mathcal{Q}^{loc}_{\{B\}} \subseteq \mathcal{Q}^{loc}_{\{A\}}$ *or* $\mathcal{Q}^{loc}_{\{A\}} \models A \sqsubseteq \bot$.

Proof. 1. Since $\mathcal{Q}^{loc}_{\mathbf{S}_2}$ is a locality-based $\mathbf{S}_2$-module in $\mathcal{Q}$, we have $\mathcal{Q} \setminus \mathcal{Q}^{loc}_{\mathbf{S}_2}$ is local w.r.t. $\mathbf{S}_2 \cup \mathsf{Sig}(\mathcal{Q}^{loc}_{\mathbf{S}_2})$. By anti-monotonicity of locality (see Lemma 17), $\mathcal{Q} \setminus \mathcal{Q}^{loc}_{\mathbf{S}_2}$ is local w.r.t. $\mathbf{S}_1 \cup \mathsf{Sig}(\mathcal{Q}^{loc}_{\mathbf{S}_2})$, hence $\mathcal{Q}^{loc}_{\mathbf{S}_2}$ is a locality-based $\mathbf{S}_1$-module in $\mathcal{Q}$. Since $\mathcal{Q}^{loc}_{\mathbf{S}_1}$ is contained in every locality-based $\mathbf{S}_1$-module in $\mathcal{Q}$ by Corollary 35, we have $\mathcal{Q}^{loc}_{\mathbf{S}_1} \subseteq \mathcal{Q}^{loc}_{\mathbf{S}_2}$.

Ontology	Language	♯ Atomic	A1: Prompt-Factor [14]		A2: Modularization from [6]		A3: Locality-based mod.	
		Concepts	Max. Size (%)	Avg. Size (%)	Max. Size (%)	Avg. Size (%)	Max. Size (%)	Avg. Size (%)
NCI	$\mathcal{EL}$	27772	24342 (87.6)	21045 (75.8)	15254 (55)	8565 (30.8)	226 (0.8)	22 (0.08)
SNOMED	$\mathcal{EL}$	255318	255318 (100)	255318 (100)	255318 (100)	255318 (100)	136 (0.5)	12.8 (0.05)
GO	$\mathcal{EL}$	22357	226 (1)	22 (0.1)	226 (1)	22 (0.1)	92 (0.4)	13 (0.05)
SUMO	$\mathcal{EL}$	869	869 (100)	869 (100)	869 (100)	869 (100)	18 (2)	8 (0.09)
GALEN-Small	$\mathcal{SHF}$	2749	2748 (100)	2748 (100)	2748 (100)	2748 (100)	297 (10)	47.7 (1.7)
GALEN-Full	$\mathcal{SHIF}$	24089	24089 (100)	24089 (100)	24089 (100)	24089 (100)	7379 (29.8)	865.5 (3.5)
SWEET	$\mathcal{SHOIF}$	1816	1750 (96.4)	1610 (88.7)	1512 (83.3)	935 (51.5)	34 (1.9)	1.7 (0.1)
DOLCE-Lite	$\mathcal{SHOIN}$	499	498 (100)	497.9 (100)	498 (100)	497.9 (100)	186 (37.3)	123.4 (24.6)

Table 4: Comparison of Different Modularization Algorithms

2. The "if" part of this property is trivial since $\mathcal{Q}^{loc}_{\{A\}} \subseteq \mathcal{Q}$. In order to prove the "only if" part of the property, assume that $\mathcal{Q} \models (A \sqsubseteq B)$. Let $\mathbf{S} := \mathsf{Sig}(\mathcal{Q}^{loc}_{\{A\}}) \cup \{A\}$, and consider the following two cases:

(a) $B \in \mathbf{S}$. Then by Remark 21, $\mathcal{Q}^{loc}_{\{A\}}$ is an $\mathbf{S}$-module in $\mathcal{Q}$, and so, $\mathcal{Q}^{loc}_{\{A\}} \models (A \sqsubseteq B)$ since $\mathsf{Sig}(A \sqsubseteq B) \subseteq \mathbf{S}$.

(b) $B \notin \mathbf{S}$. We demonstrate that $\mathcal{Q}^{loc}_{\{A\}} \models A \sqsubseteq \bot$ which suffices for proving $\mathcal{Q}^{loc}_{\{A\}} \models A \sqsubseteq B$.

Assume, to the contrary, that $\mathcal{Q}^{loc}_{\{A\}} \not\models A \sqsubseteq \bot$. Then there exists an $\mathbf{S}$-interpretation $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{Q}^{loc}_{\{A\}}$ and $A^{\mathcal{I}} \neq \emptyset$. Let $\mathcal{I}'$ be a trivial expansion of $\mathcal{I}$ to $\mathbf{S} \cup \mathsf{Sig}(\mathcal{Q})$. Since $\mathcal{Q}^{loc}_{\{A\}}$ is a locality-based $\mathbf{S}$-module in $\mathcal{Q}$ (see Remark 21 and Remark 22), we have $\mathcal{I}' \models \mathcal{Q}$. However, $\mathcal{I}'$ is not a model of $(A \sqsubseteq B)$ since $A^{\mathcal{I}'} \neq \emptyset$, but $B^{\mathcal{I}'} = \emptyset$ since $B \notin \mathbf{S}$. This contradicts to the assumption $\mathcal{Q} \models A \sqsubseteq B$.

3. As has been shown in the proof of property 2 above, if $\mathcal{Q} \models (A \sqsubseteq B)$, then either $B \in \mathsf{Sig}(\mathcal{Q}^{loc}_{\{A\}})$ or $\mathcal{Q}^{loc}_{\{A\}} \models A \sqsubseteq \bot$. So, it remains to show that $B \in \mathsf{Sig}(\mathcal{Q}^{loc}_{\{A\}})$ implies that $\mathcal{Q}^{loc}_{\{B\}} \subseteq \mathcal{Q}^{loc}_{\{A\}}$. Indeed, by Remark 21, $\mathcal{Q}^{loc}_{\{A\}}$ is a locality-based $(\mathsf{Sig}(\mathcal{Q}^{loc}_{\{A\}}) \cup \{A\})$-module in $\mathcal{Q}$. Since $B \in \mathsf{Sig}(\mathcal{Q}^{loc}_{\{A\}})$, then, in particular, $\mathcal{Q}^{loc}_{\{A\}}$ is a locality-based $\{B\}$-module in $\mathcal{Q}$. Since $\mathcal{Q}^{loc}_{\{B\}}$ is contained in every locality-based $\{B\}$-module in $\mathcal{Q}$, we have $\mathcal{Q}^{loc}_{\{B\}} \subseteq \mathcal{Q}^{loc}_{\{A\}}$ what was required to prove. $\square$

Proposition 38 gives two interesting properties of locality-based modules. The first one states that such modules may only grow if the input signature extends. The second one implies that the module for a single atomic concept A provides complete information about all the super-classes of A. This property can be used for optimizing classification: in order to *classify an ontology* $\mathcal{Q}$, i.e. to compute all *subsumption relations* $A \sqsubseteq B$ between pairs A, B of atomic concepts in $\mathcal{Q}$, it is sufficient to (1) extract all modules $\mathcal{Q}^{loc}_{\{A\}}$ of $\mathcal{Q}$ for each atomic concept A (2) classify each of these modules *independently* (possibly *in parallel*), and (3) merge the results of the individual classifications. By Property 2, if the subsumption $A \sqsubseteq B$ is implied by the ontology $\mathcal{Q}$ then it is implied by the module $\mathcal{Q}^{loc}_{\{A\}}$ and, hence, it will be obtained in step (2).

Finally, Property 3 in Proposition 37 can also be used to optimize classification. The property provides a necessary condition for a subsumption $A \sqsubseteq B$ to hold in an ontology, which can be used to quickly detect *non-subsumptions*: If the inclusion $\mathcal{Q}^{loc}_{\{B\}} \subseteq \mathcal{Q}^{loc}_{\{A\}}$ between the minimal locality-based modules does not hold, and A is found to be satisfiable, then a reasoner does not need to prove the subsumption $A \sqsubseteq B$ w.r.t. $\mathcal{Q}$, since it never holds.

5 Related Work

The problem of extracting modular fragments of ontologies has recently been addressed in [20], [14] and [18].

In [20], the authors have proposed an algorithm for partitioning the concepts in an ontology. The intended application is to facilitate the visualization of and navigation through the 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.

The algorithms in [14] and [18], which we have briefly outlined in Section 3, use structural traversal to extract modules of ontologies for a given signature. None of these approaches provides a characterization of the logical properties of the extracted modules, nor do they establish a notion of correctness of the modularization.

In [6], the authors propose a definition of a module and an algorithm for extracting modules based on that definition. The notion of a module in an ontology $\mathcal{Q}$ for a signature $\mathbf{S}$ is also based on conservative extensions: if $\mathcal{Q}_1 \subseteq \mathcal{Q}$ is an $\mathbf{S}$-module in $\mathcal{Q}$ as in [6], then it can be shown that $\mathcal{Q}$ is a model $\mathbf{S}$-conservative extension of $\mathcal{Q}$. The definition in [6], however, makes use of additional requirements which lead, in many cases, to the extraction of modules which are larger than one may wish. The reason is that, for every atomic concept $A \in \mathbf{S}$, the module $\mathcal{Q}_1$ for A in $\mathcal{Q}$ must be a module for all its sub-classes and super-classes.

It is worth pointing out that, given $\mathcal{Q}$ and $\mathbf{S}$, the fragment obtained using the algorithm in [6] is an $\mathbf{S}$-module according to Definition 1. This is not the case, however, for the fragment extracted using [18], as we have illustrated in Section 3.

6 Implementation and Evaluation

Given an input ontology and an input signature, locality-based modules are not the only possible modules we can obtain. It remains to be shown that the locality-based modules obtained in realistic ontologies are *small enough* to be useful in practice.

For evaluation and comparison, we have implemented the following algorithms using Manchester's OWL API:[3]

A1: The PROMPT-FACTOR algorithm, as described in [14];

A2: The algorithm for extracting modules described in [6];

[3]http://sourceforge.net/projects/owlapi

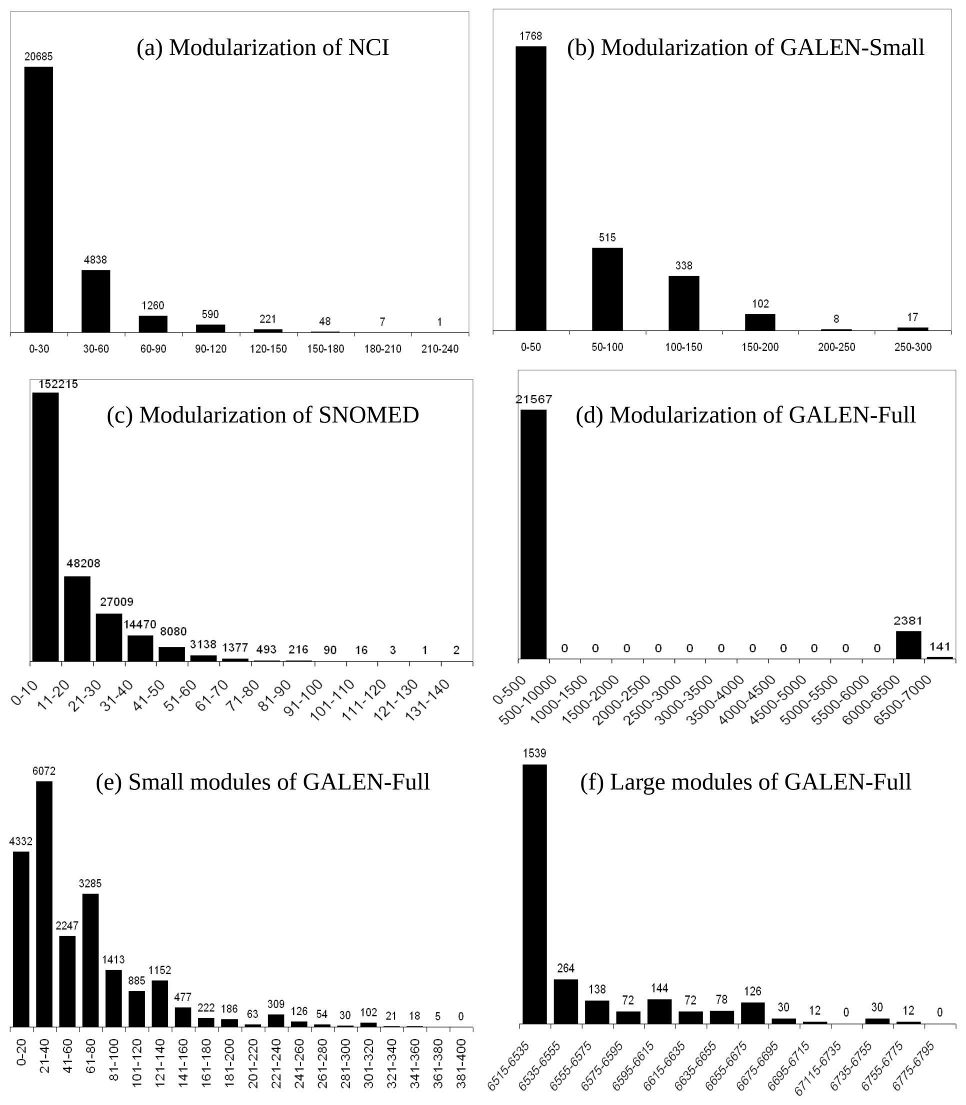


Figure 4: Distribution for the sizes of syntactic locality-based modules for atomic concepts: the X-Axis gives the number of concepts in the modules and the Y-Axis the number of modules for each size range.

A3: Our algorithm for extracting modules (Algorithm 1), based on syntactic locality.

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

Simple. In this group, we have included the National Cancer Institute (NCI) Ontology,[4] the SUMO Upper Ontology,[5] the Gene Ontology (GO),[6] and the SNOMED Ontology[7]. These ontologies use 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),[8] the DOLCE upper ontology (DOLCE-Lite),[9] and NASA's Semantic Web for Earth and Environmental Terminology (SWEET)[10]. 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.

For each of these ontologies, and for each atomic concept in their signature, we have extracted the corresponding modules using algorithms A1-A3 and measured their size. We use modules for single atomic concepts to get an idea of the typical size of locality-based modules compared to the size of the whole ontology. Also, modules for atomic concepts are especially interesting for optimized classification of ontologies, as discussed in Section 4.4.

The results we have obtained are summarized in Table 4. 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 atomic concept. In contrast, the modules we obtain using our algorithm are significantly smaller than the size of the input ontology. In fact, our modules are not only smaller, but are also strict subsets of the respective modules computed using A1 and A2.

[4]http://www.mindswap.org/2003/CancerOntology/nciOncology.owl
[5]http://ontology.teknowledge.com/
[6]http://www.geneontology.org
[7]http://www.snomed.org
[8]http://www.openclinical.org/prj_galen.html
[9]http://www.loa-cnr.it/DOLCE.html
[10]http://sweet.jpl.nasa.gov/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 1/10000 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.

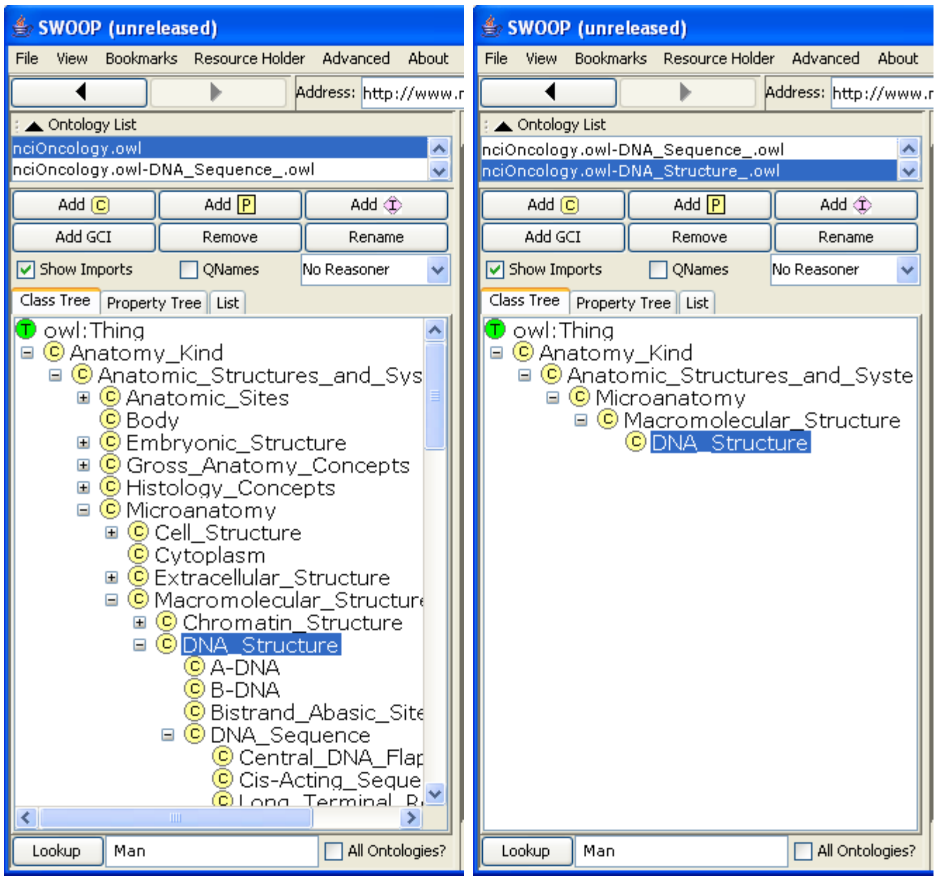
For GALEN, SWEET and DOLCE, the locality-based modules are larger. Indeed, the largest module in GALEN-Small is 1/10 of the size of the ontology, as opposed to 1/10000 in the case of SNOMED. For DOLCE, the modules are even bigger—1/3 of the size of the ontology—which indicates that the dependencies between 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 4, we have presented 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, which involves all the concepts with large modules. The presence of this cycle can be spotted more clearly in Figure 4(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 have the smooth distribution for that case. In contrast, in Figure 4(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.

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 [10]. The user interface of SWOOP allows for the selection of an input signature and the retrieval of the corresponding module.

As an illustration, consider in Figure 5 the locality-based module for the atomic concept DNA_Structure in the NCI ontology, as obtained in SWOOP. Recall that, according to Case 2 of Proposition 38, the locality-based module $\mathcal{O}^{loc}_{\{A\}}$ for every



(a) Concept DNA_Structure in NCI

(b) Syntactic locality-based module forthe concept DNA_Structure in NCI

Figure 5: The Module Extraction Functionality in Swoop

atomic concept $A \in \mathrm{Sig}(\mathcal{O})$ contains all necessary axioms for, at least, all the (entailed) super-concepts of A in $\mathcal{O}$. Thus $\mathcal{O}^{loc}_{\{A\}}$ can be seen as the "upper ontology" for A. In fact, Figure 5 shows that the locality-based module for DNA_Structure contains only the concepts in the "path" from DNA_Structure to the top level concept Anatomy_Kind. This suggests that the knowledge in NCI about the particular concept DNA_Structure is very shallow in the sense that NCI only "knows" that a DNA_Structure is a macromolecular structure, etc. which, in the end, is an anatomy kind.

7 Conclusion

In this paper, we have proposed a definition of a module for a given vocabulary within an ontology to be reused. Based on this definition, we have formulated three reasoning problems concerning the extraction of minimal modules and shown that none of them is algorithmically solvable, even for simple fragments of OWL DL. We have introduced locality-based modules as an approximation to minimal modules and have empirically demonstrated that such modules are reasonably small for many real-world ontologies.

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

8 Acknowledgments

This work is supported by the EU Project TONES (Thinking ONtologiES) ref: IST-007603 and by the EPSRC Project REOL (Reasoning in Expressive Ontology Languages) ref:EP/C537211/1. The authors would like to thank Boris Motik for his suggestions and assistance concerning this work.

References

[1] F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proc. IJCAI-2005*, pages 364–370, 2005.

[2] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

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

[4] B. Cuenca Grau, I. Horrocks, Y. Kazakov, and U. Sattler. A logical framework for modularity of ontologies. In *Proc. IJCAI-2007*, pages 298–304, 2007.

[5] B. Cuenca Grau, I. Horrocks, O. Kutz, and U. Sattler. Will my Ontologies Fit Together? In *Proc. DL-2006*, 2006.

[6] B. Cuenca Grau, B. Parsia, E. Sirin, and A. Kalyanpur. Modularity and Web Ontologies. In *Proc. KR-2006*, pages 198–209, 2006.

[7] S. Ghilardi, C. Lutz, and F. Wolter. Did I Damage my Ontology? A Case for Conservative Extensions in Description Logics. In *Proc. KR-2006*, pages 187–197, 2006.

[8] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. *J. of Web Semantics*, 1(1):7–26, 2003.

[9] I. Horrocks and U. Sattler. A tableaux decision procedure for SHOIQ. In *Proc. of the IJCAI*. Morgan Kaufman, 2005.

[10] A. Kalyanpur, B. Parsia, E.Sirin, B. Cuenca Grau, and J. Hendler. SWOOP: A web editing browser. *Elsevier's Journal Of Web Semantics*, 4(2):144–153, 2006.

[11] C. Lutz, D. Walther, and F. Wolter. Conservative extensions in expressive description logics. In *Proc. of IJCAI-2007*, pages 453–459, 2007.

[12] R. Möller and V. Haarslev. Description logic systems. In *The Description Logic Handbook*, chapter 8, pages 282–305. Cambridge University Press, 2003.

[13] B. Motik. *Reasoning in Description Logics using Resolution and Deductive Databases*. PhD thesis, Univesität Karlsruhe (TH), Karlsruhe, Germany, 2006.

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

[15] P. Patel-Schneider, P. Hayes, and I. Horrocks. Web ontology language OWL Abstract Syntax and Semantics. *W3C Recommendation*, 2004.

[16] A. Rector and J. Rogers. Ontological issues in using a description logic to represent medical concepts: Experience from GALEN. In *Proc. of IMIA WG6 Workshop*, 1999.

[17] M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artif. Intell.*, 48(1):1–26, 1991.

[18] J. Seidenberg and A. Rector. Web ontology segmentation: Analysis, classification and use. In *Proc. WWW-2006*, 2006.

[19] E. Sirin and B. Parsia. Pellet system description. In *Proc. DL-2004*, 2004.

[20] H. Stuckenschmidt and M. Klein. Structure-based partitioning of large class hierarchies. In *Proc. ISWC-2004*, pages 289–303, 2004.

[21] D. Tsarkov and I. Horrocks. FaCT++ description logic reasoner: System description. In *Proc. of IJCAR-2006*, pages 292–297, 2006.