
Extracting Modules from Ontologies: A Logic-Based Approach

Bernardo Cuenca Grau¹, Ian Horrocks¹, Yevgeny Kazakov¹, and Ulrike Sattler²

¹ University of Oxford, UK

² University of Manchester, UK

Summary. The ability to extract meaningful fragments from an ontology is essential for ontology reuse. 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. We show that the problem of determining whether a subset of an ontology is a module for a given vocabulary is undecidable even for OWL DL. Given these negative results, we propose sufficient conditions for a fragment of an ontology to be a module. We propose an algorithm for computing modules based on those conditions and present our experimental results on a set of real-world ontologies of varying size and complexity.

6.1 Introduction

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

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

The most straightforward way to reuse these concepts is to construct the logical union $\mathcal{P} \cup \mathcal{Q}$ of the axioms in \mathcal{P} and \mathcal{Q} . This form of reuse is used frequently and can be achieved in OWL using the `owl : imports` construct. Well-established medical ontologies, such as NCI and SNOMED, 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* Q_1 within a given ontology Q for a given vocabulary S .
2. We take the above definition as a starting point, and investigate the problem of computing modules. We show that none of the reasonable variants of this problem is solvable in general already for rather restricted sub-languages of OWL DL. In fact, it is even not possible to determine whether a subset Q_1 of an ontology Q is a module in Q for S .
3. Given these negative results, we propose sufficient conditions for a fragment of an ontology to be a module—that is, if the fragment satisfies our conditions then we can guarantee that it is a module but not vice versa. These conditions are based on the notion of locality of an ontology w.r.t. a signature, as first introduced in [4].
4. We propose an algorithm for computing locality-based modules.
5. We describe our implementation and present empirical results on a set of real-world ontologies of varying size and complexity. Using our syntactic approximation, we obtain modules that are small enough for reuse applications.

6.2 Preliminaries

In this section we introduce description logics (DLs) [2], a family of knowledge representation formalisms which underlie modern ontology languages, such as OWL DL [16]. A hierarchy of commonly-used description logics is summarized in Table 6.1.

The *syntax* of a description logic \mathbb{L} is given by a signature and a set of constructors. A *signature* (or *vocabulary*) Sg of a DL is the (disjoint) union of countably infinite sets C of *atomic concepts* (A, B, \dots) representing sets of elements, R of

Table 6.1. The hierarchy of standard description logics

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

Here $r \in \mathbf{R}$, $A \in \mathbf{C}$, $a, b \in \mathbf{I}$, $R_{(i)} \in \mathbf{R}$, $C_{(i)} \in \mathbf{C}$, $n \geq 1$ and $S \in \mathbf{R}$ a simple role[10].

atomic roles (r, s, \dots) representing binary relations between elements, and \mathbf{I} of *individuals* (a, b, c, \dots) representing constants. We assume the signature to be fixed for every DL.

Each DL provides *constructors* for defining the set \mathbf{R} of (general) *roles* (R, S, \dots), the set \mathbf{C} of (general) *concepts* (C, D, \dots), and \mathbf{Ax} of *axioms* (α, β, \dots) which includes the *role axioms* (RBox), *terminological axioms* (TBox) and *assertions* (ABox).

\mathcal{EL} [1] is a simple DL which allows one to construct complex concepts using *conjunction* $C_1 \sqcap C_2$ and *existential restriction* $\exists R.C$ starting from atomic concepts A , roles R and the *top concept* \top . \mathcal{EL} provides no role constructors and no role axioms; thus, each 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* $r(a, b)$. We assume the concept definition $A \equiv C$ is an abbreviation for two GCIs $A \sqsubseteq C$ and $C \sqsubseteq A$.

The basic description logic \mathcal{ALC} [17] is obtained from \mathcal{EL} by adding the *concept negation* constructor $\neg C$. We introduce some additional constructors as abbreviations: the *bottom concept* \perp is a shortcut for $\neg \top$, the *concept disjunction* $C_1 \sqcup C_2$ stands for $\neg(\neg C_1 \sqcap \neg C_2)$, and the *value restriction* $\forall R.C$ stands for $\neg(\exists R.\neg C)$. In contrast to \mathcal{EL} , \mathcal{ALC} can express *contradiction axioms* like $\top \sqsubseteq \perp$. The logic \mathcal{S} is an extension of \mathcal{ALC} where, additionally, some atomic roles can be declared to be *transitive* using a role axiom $\text{Trans}(r)$.

Further extensions of DLs add features such as *inverse roles* r^- (indicated by appending a letter \mathcal{I} to the name of the logic), *role inclusion axioms* (RIs) $R_1 \sqsubseteq R_2$ ($+ \mathcal{H}$), *functional roles* $\text{Funct}(R)$ ($+ \mathcal{F}$), *number restrictions* $(\geq n S)$, with $n \geq 1$, ($+ \mathcal{N}$), *qualified number restrictions* $(\geq n S.C)$, with $n \geq 1$, ($+ \mathcal{Q}$)¹, and *nominals* $\{a\}$ ($+ \mathcal{O}$). Nominals allow for the construction of concepts representing a singleton

¹ The dual constructors $(\leq n S)$ and $(\leq n S.C)$ are abbreviations for $\neg(\geq n + 1 S)$ and $\neg(\geq n + 1 S.C)$, respectively.

set $\{a\}$ (a *nominal* concept) from an individual a . These extensions can be used in different combinations; for example \mathcal{ALCO} is an extension of \mathcal{ALC} with nominals; \mathcal{SHIQ} is an extension of \mathcal{S} with role hierarchies, inverse roles and qualified number restrictions; and \mathcal{SHOIQ} is the DL that uses all the constructors and axiom types we have presented.

Modern ontology languages, such as OWL, are based on description logics and, to a certain extent, are syntactic variants thereof. In particular, OWL DL corresponds to \mathcal{SHOIN} [9]. In this paper, we assume an *ontology* \mathcal{O} based on a *description logic* \mathbb{L} to be a finite set of axioms in \mathbb{L} . The *signature of an ontology* \mathcal{O} (of an axiom α) is the set $\text{Sig}(\mathcal{O})$ ($\text{Sig}(\alpha)$) of atomic concepts, atomic roles and individuals that occur in \mathcal{O} (respectively in α).

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

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

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

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

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

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

Let S_1, S be signatures such that $S_1 \subseteq S$. The *restriction of an S-interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ to S_1 is an interpretation $\mathcal{I}|_{S_1} = (\Delta^{\mathcal{I}_1}, \cdot^{\mathcal{I}_1})$ over S_1 such that $\Delta^{\mathcal{I}_1} = \Delta^{\mathcal{I}}$ and $X^{\mathcal{I}_1} = X^{\mathcal{I}}$ for every $X \in S_1$. An *expansion of an S_1 -interpretation \mathcal{I}_1 to S* is an S-interpretation \mathcal{I} such that $\mathcal{I}|_{S_1} = \mathcal{I}_1$. A *trivial expansion of an S_1 -interpretation \mathcal{I}_1 to S* is an expansion of \mathcal{I}_1 to S such that $X^{\mathcal{I}} = \emptyset$ for every atomic concept and atomic role $X \in S \setminus S_1$.

6.3 Modules for Knowledge Reuse

Suppose that an ontology engineer wants to build an ontology about research projects. He defines two concepts `Genetic_Disorder_Project` and `Cystic_Fibrosis_EUPProject` in his ontology \mathcal{P} . The first one describes projects about genetic disorders; the second one, European projects about cystic fibrosis, as given by the axioms P1 and P2 in Figure 6.1.

Ontology of medical research projects \mathcal{P}:	
P1	<code>Genetic_Disorder_Project</code> \equiv <code>Project</code> \sqcap \exists <code>has_Focus</code> . <code>Genetic_Disorder</code>
P2	<code>Cystic_Fibrosis_EUPProject</code> \equiv <code>EUPProject</code> \sqcap \exists <code>has_Focus</code> . <code>Cystic_Fibrosis</code>
P3	<code>EUPProject</code> \sqsubseteq <code>Project</code>
Ontology of medical terms \mathcal{Q}:	
M1	<code>Cystic_Fibrosis</code> \equiv <code>Fibrosis</code> \sqcap \exists <code>located_In</code> . <code>Pancreas</code> \sqcap \exists <code>has-Origin</code> . <code>Genetic-Origin</code>
M2	<code>Genetic_Fibrosis</code> \equiv <code>Fibrosis</code> \sqcap \exists <code>has-Origin</code> . <code>Genetic-Origin</code>
M3	<code>Fibrosis</code> \sqcap \exists <code>located_In</code> . <code>Pancreas</code> \sqsubseteq <code>Genetic_Fibrosis</code>
M4	<code>Genetic_Fibrosis</code> \sqsubseteq <code>Genetic_Disorder</code>
M5	<code>DEFBI_Gene</code> \sqsubseteq <code>Immuno_Protein_Gene</code> \sqcap <code>associated_With</code> . <code>Cystic_Fibrosis</code>

Fig. 6.1. Reusing medical terminology in an ontology on research projects

The ontology engineer is an expert on research projects: he knows, for example, that a `EUPProject` is a `Project` (axiom P3). He is unfamiliar, however, with most of the topics the projects cover and, in particular, with the terms `Cystic_Fibrosis` and `Genetic_Disorder` mentioned in P1 and P2. In 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} .

Suppose that the concepts `Cystic_Fibrosis` and `Genetic_Disorder` are described in an ontology \mathcal{Q} containing axioms M1-M5 in Figure 6.1. If we include in the module

\mathcal{Q}_1 just the axioms that mention either Cystic_Fibrosis or Genetic_Disorder, namely M1, M4 and M5, we lose the following dependency:

$$\text{Cystic_Fibrosis} \sqsubseteq \text{Genetic_Disorder} \quad (6.1)$$

The dependencies $\text{Cystic_Fibrosis} \sqsubseteq \text{Genetic_Fibrosis} \sqsubseteq \text{Genetic_Disorder}$ follow from axioms M1-M5, but not from M1, M4, M5, since the dependency $\text{Cystic_Fibrosis} \sqsubseteq \text{Genetic_Fibrosis}$ does not hold after removing M2 and M3. The dependency (6.1), however, is crucial for our ontology \mathcal{P} as it (together with axiom P3) implies the following axiom:

$$\text{Cystic_Fibrosis_EUProject} \sqsubseteq \text{Genetic_Disorder_Project} \quad (6.2)$$

This means, in particular, that all the projects annotated with the concept name Cystic_Fibrosis_EUProject must be included in the answer for a query on the concept name 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 (6.2), 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 L be a description logic, $\mathcal{Q}_1 \subseteq \mathcal{Q}$ be two ontologies expressed in L and let \mathbf{S} be a signature. We say that \mathcal{Q}_1 is an \mathbf{S} -module in \mathcal{Q} w.r.t. L , if for every ontology \mathcal{P} and every axiom α expressed in L with $\text{Sig}(\mathcal{P} \cup \{\alpha\}) \cap \text{Sig}(\mathcal{Q}) \subseteq \mathbf{S}$, we have $\mathcal{P} \cup \mathcal{Q} \models \alpha$ iff $\mathcal{P} \cup \mathcal{Q}_1 \models \alpha$. \diamond

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 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.

According to Definition 1, there may be a large number of modules for a given input ontology and signature. In many applications one is usually not interested in extracting arbitrary modules from a reused ontology, but in extracting modules that are easy to process afterwards. Ideally, the extracted modules should be as small as possible. Hence, it is reasonable to consider the problem of extracting *minimal modules*—that is, modules that contain no other module as a subset. In our example from Figure 6.1, there are two minimal \mathbf{S} -modules $\mathcal{Q}_1 = \{\mathcal{M}1, \mathcal{M}2, \mathcal{M}4\}$ and $\mathcal{Q}_2 = \{\mathcal{M}1, \mathcal{M}3, \mathcal{M}4\}$: if we remove any axiom from them, the dependency (6.1) will no longer hold.

As seen above, 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:

- $$\begin{array}{ll} \text{T1. compute } \textit{all} \text{ minimal } \mathbf{S}\text{-modules in } \mathcal{Q} & \\ \text{T2. compute } \textit{some} \text{ minimal } \mathbf{S}\text{-module in } \mathcal{Q} & \end{array} \quad (6.3)$$

Axioms that do not occur in a minimal module of \mathcal{Q} are not essential for \mathcal{P} in the sense that they do not need to be imported into \mathcal{P} . This is not true for the axioms that occur in minimal modules of \mathcal{Q} . These arguments motivate the following notion:

Definition 2 (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

In our example, the axioms $\mathcal{M}1 - \mathcal{M}4$ from \mathcal{Q} are essential for the signature $\mathbf{S} = \{\text{CysticFibrosis}, \text{Genetic_Disorder}\}$, and the axiom $\mathcal{M}5$ is not essential. In certain situations one might be interested in computing the set of (non)essential axioms of an ontology, which can be done by computing the union of all minimal modules. Hence, the following task may also be of interest:

- $$\begin{array}{ll} \text{T3. compute } \textit{the union} \text{ of all minimal } \mathbf{S}\text{-modules in } \mathcal{Q}, & \\ \text{which is the set of all } \mathbf{S}\text{-essential axioms in } \mathcal{Q} & \end{array} \quad (6.4)$$

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

6.4 Computational Properties of Module Extraction

In this section, we study the decidability/computability of the tasks described in Section 6.3. We start by investigating the relationships between Tasks T1, T2 and T3; our main result is that tasks T1 and T2 are inter-reducible whereas T3 is “easier” than both T1 and T2. Next, we establish the relationship between our notion of module and the notion of a *conservative extension*, whose complexity/decidability

for Description Logics has been recently established. Finally, we show that deciding whether an axiom is essential in an ontology for a given signature is an undecidable problem for the logic \mathcal{ALCO} . As a consequence, tasks T1-T3 are proved algorithmically unsolvable for \mathcal{ALCO} .

6.4.1 Reductions between Tasks

Before we formalize we establish the reductions between tasks T1-T3, we prove in the following proposition some important properties of modules that we will exploit along this section.

Proposition 1 (Properties of Modules)

Let $\mathcal{Q}_1 \subseteq \mathcal{Q}_2 \subseteq \mathcal{Q}_3$ be three \mathcal{SHOIQ} 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 $\text{Sig}(\mathcal{P} \cup \{\alpha\}) \cap \text{Sig}(\mathcal{Q}_3) \subseteq \mathbf{S}$ 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 , $\text{Sig}(\mathcal{P} \cup \{\alpha\}) \cap \text{Sig}(\mathcal{Q}_3) \subseteq \mathbf{S}$ 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 , $\text{Sig}(\mathcal{P} \cup \{\alpha\}) \cap \text{Sig}(\mathcal{Q}_2) \subseteq \text{Sig}(\mathcal{P} \cup \{\alpha\}) \cap \text{Sig}(\mathcal{Q}_3) \subseteq \mathbf{S}$, 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 $\text{Sig}(\mathcal{P} \cup \{\alpha\}) \cap \text{Sig}(\mathcal{Q}_2) \subseteq \mathbf{S}$ and $\mathcal{P} \cup \mathcal{Q}_2 \models \alpha$. We demonstrate that $\mathcal{P} \cup \mathcal{Q}_1 \models \alpha$ (\dagger):

Without loss of generality, we can assume that $\text{Sig}(\mathcal{P} \cup \{\alpha\}) \cap \text{Sig}(\mathcal{Q}_3) \subseteq \mathbf{S}$, since the symbols that are in $\text{Sig}(\mathcal{P} \cup \{\alpha\})$ but not in $\text{Sig}(\mathcal{Q}_2)$ could be renamed so that they are not contained in $\text{Sig}(\mathcal{Q}_3)$. 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 $\mathcal{P} \cup \mathcal{Q}_1 \models \alpha$ (\dagger).

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 $\text{Sig}(\mathcal{P} \cup \{\alpha\}) \cap \text{Sig}(\mathcal{Q}_3) \subseteq \mathbf{S}$ and $\mathcal{P} \cup \mathcal{Q}_3 \models \alpha$. We demonstrate that $\mathcal{P} \cup \mathcal{Q}_2 \models \alpha$ (\ddagger):

Since \mathcal{Q}_1 is an \mathbf{S} -module in \mathcal{Q}_3 , $\text{Sig}(\mathcal{P} \cup \{\alpha\}) \cap \text{Sig}(\mathcal{Q}_3) \subseteq \mathbf{S}$, 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 $\mathcal{P} \cup \mathcal{Q}_2 \models \alpha$ (\ddagger). \diamond

Intuitively, Part 2(a) of Proposition 1 claims 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.

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 1 (A Criterion 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 1 (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 1, \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 1, \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 . \diamond

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

Proposition 2. *Tasks T1 and T2 from (6.3) 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 1. Note that procedure **P2** satisfies the following property:

$$\text{Given } \mathbf{S} \text{ and } \mathcal{Q}_2, \text{ the procedure } \mathbf{P2} \text{ for T2 returns } \mathcal{Q}_2 \text{ if and only if } \mathcal{Q}_2 \text{ is the only minimal } \mathbf{S}\text{-module in } \mathcal{Q}_2. \quad (6.5)$$

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 (6.5) 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 1, **P1** returns exactly all minimal \mathbf{S} -modules in \mathcal{Q} . \diamond

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

Proposition 3. *Tasks T1 and T2 are reducible to task T3; that is, any procedure for T1 or T2 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 6.4.2, this issue is not relevant since all of the tasks formulated above are algorithmically unsolvable for OWL DL.

6.4.2 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, 12]. In the literature we can find at least two different notions of conservative extensions in the context of ontologies [12]:

Definition 3 (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 $\text{Sig}(\alpha) \subseteq \mathbf{S}$, 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 model conservative extension is strictly stronger than the deductive one [12] 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 1. Let \mathcal{Q} be the ontology consisting of axioms $\mathcal{M}1 - \mathcal{M}5$ in Figure 6.1. Let $\mathbf{S} = \{\text{Cystic_Fibrosis}, \text{Genetic_Disorder}\}$ and $\mathcal{Q}_1 = \{\mathcal{M}1, \dots, \mathcal{M}4\}$. 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. $\mathcal{I}|_{\mathbf{S}} = \mathcal{I}_1|_{\mathbf{S}}$.

Let \mathcal{I} be as \mathcal{I}_1 except for the interpretation of the atomic concepts DEFBI_Gene and Immuno_Protein_Gene, and the atomic role 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 $\mathcal{M5}$ 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 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 DEFBI_Gene since $\mathcal{M5}$ is satisfied in every interpretation where this concept is interpreted as the empty set.

Now, if we remove $\mathcal{M2}$ and $\mathcal{M3}$ from \mathcal{Q}_1 , then \mathcal{Q} is no longer a model \mathbf{S} -conservative extension of \mathcal{Q}_1 for $\mathbf{S} = \{\text{Cystic_Fibrosis}, \text{Genetic_Disorder}\}$. Indeed, it is possible to find an interpretation \mathcal{I}_1 of the remaining axioms $\mathcal{M1}$ and $\mathcal{M4}$ from \mathcal{O}_1 , in which Genetic_Disorder is interpreted as the empty set, but Cystic_Fibrosis is not. For example, consider an interpretation $\mathcal{I}_1 = (\{a\}, \mathcal{I}_1^1)$ with:

$$\begin{aligned} \text{Cystic_Fibrosis}^{\mathcal{I}_1} &= \text{Fibrosis}^{\mathcal{I}_1} = \text{Pancreas}^{\mathcal{I}_1} = \text{Genetic_Origin}^{\mathcal{I}_1} = \{a\}; \\ \text{located_In}^{\mathcal{I}_1} &= \text{has_Origin}^{\mathcal{I}_1} = \{(a, a)\}; \text{ and} \\ \text{Genetic_Fibrosis}^{\mathcal{I}_1} &= \text{Genetic_Disorder}^{\mathcal{I}_1} = \emptyset. \end{aligned}$$

We can see that \mathcal{I}_1 is a model of $\mathcal{M1}$ and $\mathcal{M4}$, 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 $\mathcal{I} \models \alpha := (\text{Cystic_Fibrosis} \sqsubseteq \text{Genetic_Disorder})$ 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 4 (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. \mathbf{L} then \mathcal{Q} is a deductive \mathbf{S} -conservative extension of \mathcal{Q}_1 w.r.t. \mathbf{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 $\text{Sig}(\alpha) \in \mathbf{S}$ 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} , $\text{Sig}(\mathcal{P} \cup \{\alpha\}) \cap \text{Sig}(\mathcal{Q}) \subseteq \mathbf{S}$, 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 $\text{Sig}(\mathcal{P} \cup \{\alpha\}) \cap \text{Sig}(\mathcal{Q}) \subseteq \mathbf{S}$, 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 $\mathcal{I}'_1 := \mathcal{I}_1|_{\text{SUSig}(\mathcal{Q})}$. Obviously, $\mathcal{I}'_1 \models \mathcal{Q}_1$. By Definition 3, 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 $\mathcal{I}'|_{\text{SUSig}(\mathcal{Q})}$ to $\text{Sig}(\mathcal{P} \cup \{\alpha\})$ by setting $X^{\mathcal{I}} := X^{\mathcal{I}'_1}$ for every $X \in \text{Sig}(\mathcal{P} \cup \{\alpha\}) \setminus \mathbf{S}$. Note that we also have $\mathcal{I}|_{\mathbf{S}} = \mathcal{I}'|_{\mathbf{S}} = \mathcal{I}'_1|_{\mathbf{S}} = \mathcal{I}_1|_{\mathbf{S}}$, hence $\mathcal{I}|_{\text{Sig}(\mathcal{P} \cup \{\alpha\})} = \mathcal{I}_1|_{\text{Sig}(\mathcal{P} \cup \{\alpha\})}$, and so $\mathcal{I} \models \mathcal{P}$ and $\mathcal{I} \not\models \alpha$. Since $\mathcal{I}|_{\text{SUSig}(\mathcal{Q})} = \mathcal{I}'|_{\text{SUSig}(\mathcal{Q})}$ and $\mathcal{I}' \models \mathcal{Q}$, we have $\mathcal{I} \models \mathcal{Q}$, which yields a contradiction. \diamond

Proposition 4 shows that our notion of module stays “in between” the two notions of conservative extensions. In particular, by applying Property 2 in Proposition 4 to Example 1, we can show that the axioms M1-M4 in Figure 6.1 constitute a module in the ontology \mathcal{Q} , consisting of M1-M5. The converse of Property 1 in Proposition 4, however, does not hold in general:

Example 2. 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 $\top \sqsubseteq \top$, $\top \sqsubseteq \perp$, $\top \sqsubseteq A$ or $A \sqsubseteq \perp$, 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 $\mathcal{P} = \{A \sqsubseteq \perp\}$, which is constructed over \mathbf{S} . We can see that $\mathcal{P} \cup \mathcal{Q} \models \top \sqsubseteq \perp$, but $\mathcal{P} \cup \mathcal{Q}_1 \not\models \top \sqsubseteq \perp$. \diamond

Note that the construction in Example 2 also shows that the notion of deductive conservative extension is strictly weaker than the notion of model conservative extension (as shown in [12]): \mathcal{Q} is a deductive conservative extension of \mathcal{Q}_1 but, according to Property 2 in Proposition 4, it is not a model conservative extension.

6.4.3 Undecidability Results

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 [12], where it is proved to be 2-EXPTIME complete for $\mathcal{ALC}\mathcal{I}\mathcal{Q}$

(roughly OWL-Lite) and undecidable for OWL DL. For model conservative extensions, the problem is highly undecidable (non recursively enumerable), even for \mathcal{ALC} [12].

The decidability result from [12] 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{ALCO} ontologies:

Theorem 1 (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. $L = \mathcal{ALCO}$.

Proof. The proof is a variation of the construction for undecidability of deciding deductive conservative extensions in \mathcal{ALCQIO} given [12], based on a reduction from a domino tiling problem.

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}_f be the subset of \mathcal{D} that admit a solution and \mathcal{D}_{\surd} be the subset of \mathcal{D} that admit a periodic solution. Note that $\mathcal{D}_{\surd} \subseteq \mathcal{D}_f$. It is well-known [3, Theorem 3.1.7] that the sets $\mathcal{D} \setminus \mathcal{D}_f$ and \mathcal{D}_{\surd} are *recursively inseparable*, that is, there is no recursive (i.e. decidable) subset $\mathcal{D}' \subseteq \mathcal{D}$ of domino systems such that $\mathcal{D}_{\surd} \subseteq \mathcal{D}' \subseteq \mathcal{D}_f$.

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 $\mathcal{D}_{\surd} \subseteq \mathcal{D}' \subseteq \mathcal{D}_f$. Since $\mathcal{D} \setminus \mathcal{D}_f$ and \mathcal{D}_{\surd} 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 6.2 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

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

Fig. 6.2. An ontology \mathcal{Q} for a domino system D

of \mathcal{Q} for which the horizontal and vertical matching relations do not commute. We can 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}^α be a minimal \mathbf{S} -module in \mathcal{Q} containing α . Note that \mathcal{Q}^α implies all axioms of form (q₁)–(q₄) in \mathcal{Q} , since the signature of these axioms is a subset of \mathbf{S} . Since \mathcal{Q}^α 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 4, \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 (q₁)–(q₄), 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 (q₁)–(q₄) and admits commutativity property for relations r_H and r_V , we can 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{ALCCO} -ontology \mathcal{P} such that $\mathcal{P} \cup \mathcal{Q} \models \perp$, but $\mathcal{P} \cup \mathcal{Q}_1 \not\models \perp$. 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}

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

The axioms (p₁)–(p₄) encode the solution $t_{i,j}$ for \mathcal{D} , and so, ensure that axioms (q₁)–(q₄) are satisfied. The axioms (p₅) and (p₆) 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} . \diamond

Corollary 1. *There exists no algorithm for performing any of the tasks T1-T3 from (6.3), and (6.4) for \mathcal{ALCO} -ontologies.*

Proof. Theorem 1 implies directly that there is no algorithm for task T3 from (6.4), 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 (6.3) are unsolvable since they are reducible to T3 by Proposition 3. \diamond

Corollary 2. *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. $\mathbf{L} = \mathcal{ALCO}$.*

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 1. 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} . \diamond

Corollary 2 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:

$$\text{T2w. compute some small enough } \mathbf{S}\text{-module in } \mathcal{Q} \quad (6.6)$$

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 what follows, 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.

6.5 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. In this section, we restrict ourselves to \mathcal{SHIQ} , although the results provided here could be extended to \mathcal{SHOIQ} using the same treatment for nominals as in [4].

6.5.1 Locality

As a consequence of Case 2 in Proposition 4, 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 [12]. 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 1, where we show that the set \mathcal{Q} of axioms M1-M5 in Figure 6.1 is a model \mathbf{S} -conservative extension of $\mathcal{Q}_1 = \{\mathcal{M}1, \dots, \mathcal{M}4\}$, for $\mathbf{S} = \{\text{Cystic_Fibrosis}, \text{Genetic_Disorder}\}$. In this example, the model conservative extension was shown by finding expansions of $\text{Sig}(\mathcal{Q}_1)$ -interpretations to models of \mathcal{Q} in which all concept and atomic roles not in $\text{Sig}(\mathcal{Q}_1)$ 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 4 (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 $\mathbf{S} \cup \text{Sig}(\alpha)$ is a model of α . We denote by $\text{local}(\mathbf{S})$ 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 $\mathcal{O} \subseteq \text{local}(\mathbf{S})$. \diamond*

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 3. Axiom M5 is local for $\mathbf{S} = \{\text{Cystic_Fibrosis}, \text{Genetic_Disorder}\}$. Indeed, as shown in Example 1, for every trivial expansion \mathcal{I} of an \mathbf{S} -interpretation to $\mathbf{S} \cup \text{Sig}(\alpha)$, the concept 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. $\mathbf{S} = \{\text{DEFBI_Gene}\}$. Indeed, take any \mathbf{S} -interpretation \mathcal{I}_1 in which 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 DEFBI_Gene. \diamond

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 2 (Anti-Monotonicity of Locality). *Let \mathbf{S}_1 and \mathbf{S}_2 be signature sets. Then $\mathbf{S}_1 \subseteq \mathbf{S}_2$ implies $\text{local}(\mathbf{S}_2) \subseteq \text{local}(\mathbf{S}_1)$.*

Proof. Let $\alpha \in \text{local}(\mathbf{S}_2)$. We demonstrate that $\alpha \in \text{local}(\mathbf{S}_1)$. 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 $\mathbf{S}_1 \cup \text{Sig}(\alpha)$ 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 $\alpha \in \text{local}(\mathbf{S}_2)$, every trivial expansion \mathcal{I}'_2 of \mathcal{I}_2 to $\mathbf{S}_2 \cup \text{Sig}(\alpha)$ is a model of α . Note that \mathcal{I}'_2 is a trivial expansion of \mathcal{I}_1 to $\mathbf{S}_2 \cup \text{Sig}(\alpha)$, hence $\mathcal{I}'_1 = \mathcal{I}'_2|_{\mathbf{S}_1 \cup \text{Sig}(\alpha)} \models \alpha$. \diamond

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

Proposition 5 (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. $\mathbf{S} \cup \text{Sig}(\mathcal{O}_1)$. 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 $\mathcal{I}_1|_{\mathbf{S} \cup \text{Sig}(\mathcal{O}_1)}$ to $\mathbf{S} \cup \text{Sig}(\mathcal{O}_1) \cup \text{Sig}(\mathcal{O}_2)$, thus, in particular, $\mathcal{I}|_{\mathbf{S} \cup \text{Sig}(\mathcal{O}_1)} = \mathcal{I}_1|_{\mathbf{S} \cup \text{Sig}(\mathcal{O}_1)}$. 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. $\mathbf{S} \cup \text{Sig}(\mathcal{O}_1)$, by Definition 4, \mathcal{I} is a model of \mathcal{O}_2 . Moreover, since $\mathcal{I}|_{\text{Sig}(\mathcal{O}_1)} = \mathcal{I}_1|_{\text{Sig}(\mathcal{O}_1)}$ 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. \diamond

Using Proposition 5 and Property 2 of Proposition 4 we obtain:

Corollary 3. *Let $\mathcal{O}_1, \mathcal{O}_2$ and \mathbf{S} be as given in Proposition 5. Then \mathcal{O}_1 is an \mathbf{S} -module in $\mathcal{O}_1 \cup \mathcal{O}_2$.*

Corollary 3 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. $\mathbf{S} \cup \text{Sig}(\mathcal{Q}_1)$. In this case, \mathcal{Q}_1 is an \mathbf{S} -module in \mathcal{Q} .

Definition 5 (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. $\mathbf{S} \cup \text{Sig}(\mathcal{Q}_1)$. \diamond

Remark 1. Note from Definition 5 that every locality-based \mathbf{S} -module \mathcal{Q}_1 in \mathcal{Q} , is also a locality-based $\mathbf{S} \cup \text{Sig}(\mathcal{Q}_1)$ -module in \mathcal{Q} . \diamond

Remark 2. 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 $\mathbf{S} \cup \text{Sig}(\mathcal{Q}_1)$ to $\mathbf{S} \cup \text{Sig}(\mathcal{Q})$, is a model for \mathcal{Q} . \diamond

Example 4 (Example 3, continued). We have seen in Example 3 that axiom M5 is local w.r.t. every \mathbf{S} that does not contain the atomic concept DEFBI_Gene. In particular, for \mathcal{Q}_1 consisting of axioms M1-M4 from Figure 6.1, M5 is local w.r.t. $\text{Sig}(\mathcal{Q}_1)$. Hence, according to Definition 5, \mathcal{Q}_1 is a locality-based \mathbf{S} -module in $\mathcal{Q} = \{\mathcal{M}1, \dots, \mathcal{M}5\}$ for every $\mathbf{S} \subseteq \text{Sig}(\mathcal{Q}_1)$. \diamond

Remark 3. Note that the analog of the Part 1 in Proposition 1 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. $\mathbf{S} \cup \text{Sig}(\mathcal{Q}_1) = \{A_1, A_2\}$. 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. $\mathbf{S} \cup \text{Sig}(\mathcal{Q}'_1) = \{A_1, A_2, B\}$. 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

6.5.2 Testing Locality

As demonstrated in Example 3, 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 6 (Testing Locality). *Let \mathcal{O} be a SHIQ 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 $(\geq n R.C)$ with \perp ; (2) remove every transitivity axiom $\text{Trans}(r)$; (3) replace every assertion $a : A$ and $r(a, b)$ with the contradiction axiom $\top \sqsubseteq \perp$.*

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 $\mathbf{S} \cup \text{Sig}(\mathcal{O})$. 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} . \diamond

Note that according to Definition 4, 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 6.

Example 5. Recall that in Example 3 we have demonstrated that axiom M5 from Figure 6.1 is local w.r.t. $\mathbf{S} = \{\text{Cystic_Fibrosis}, \text{Genetic_Disorder}\}$. Now we demonstrate this using Proposition 6. Indeed, according to this proposition we need to perform the following replacements:

$$\begin{aligned} \text{DEFBI_Gene} &\Rightarrow \perp && \text{(by (1) since DEFBI_Gene} \notin \mathbf{S}) \\ \text{Immuno_Protein_Gene} &\Rightarrow \perp && \text{(by (1) since Immuno_Protein_Gene} \notin \mathbf{S}) \\ \exists \text{associated_With.Cystic_Fibrosis} &\Rightarrow \perp && \text{(by (1) since associated_With} \notin \mathbf{S}) \end{aligned}$$

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

An important conclusion of Proposition 6 is that one can use the standard capabilities of available DL-reasoners² such as **FaCT++** [19], **RACER** [13], **Pellet** [18] or **KAON2** [14] 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 *SHIQ* 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 *SHIQ*:

Definition 6 (Syntactic Locality for *SHIQ*). Let \mathbf{S} be a signature. The following grammar recursively defines two sets of concepts $\mathcal{C}_{\mathbf{S}}^{\perp}$ and $\mathcal{C}_{\mathbf{S}}^{\top}$ for a signature \mathbf{S} :

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

where $A^{\perp} \notin \mathbf{S}$ is a atomic concept, R is a role, and C is a concept, $C^{\perp} \in \mathcal{C}_{\mathbf{S}}^{\perp}$, $C_{(i)}^{\top} \in \mathcal{C}_{\mathbf{S}}^{\top}$, $i = 1, 2$, and $R^{\perp} \notin \text{Rol}(\mathbf{S})$ 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) $\text{Trans}(R^{\perp})$, or (3) $C^{\perp} \sqsubseteq C$ or (4) $C \sqsubseteq C^{\top}$. We denote by $\text{s_local}(\mathbf{S})$ the set of all *SHIQ*-axioms that are syntactically local w.r.t. \mathbf{S} . A *SHIQ*-ontology \mathcal{O} is syntactically local w.r.t. \mathbf{S} if $\mathcal{O} \subseteq \text{s_local}(\mathbf{S})$. \diamond

Intuitively, every concept in $\mathcal{C}_{\mathbf{S}}^{\perp}$ 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 $\mathcal{C}_{\mathbf{S}}^{\top}$ 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 6.1 is local w.r.t. $\mathbf{S} = \{\text{Fibrosis, has_Origin}\}$: if we replace the remaining symbols in this axiom with \perp , we obtain a tautology $\perp \equiv \perp$:

$$\underbrace{\perp}_{\text{Genetic_Fibrosis}} \equiv \text{Fibrosis} \sqcap \underbrace{\exists \text{has_Origin. } \perp}_{\text{Genetic_Origin}}$$

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.

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

It is easy to show that the analog of Lemma 2 also holds for syntactic locality:

Lemma 3 (Anti-Monotonicity of Syntactic Locality)

Let S_1 and S_2 be signature sets. Then $S_1 \subseteq S_2$ implies $s_local(S_2) \subseteq s_local(S_1)$.

Proof. It is easy to see from Definition 6 that $C_{S_2}^\perp \subseteq C_{S_1}^\perp$, $C_{S_2}^\top \subseteq C_{S_1}^\top$, and hence, $s_local(S_2) \subseteq s_local(S_1)$. \diamond

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

Proposition 7. Let S be a signature. Then $s_local(S) \subseteq local(S)$.

Proof. Let α be an axiom that is syntactically local w.r.t. S and let $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$ be a trivial expansion of some S -interpretation to $S \cup Sig(\alpha)$. We have to demonstrate that \mathcal{I} is a model of α . By induction over the definitions of C_S^\perp and C_S^\top from Definition 6, it is easy to show that: (i) every role $R \notin Rol(S)$ and every every concept from C_S^\perp is interpreted in \mathcal{I} with the empty set, and (ii) every concept from C_S^\top is interpreted in \mathcal{I} with Δ . By checking all the possible cases for a syntactically local axiom α in Definition 5, it is easy to see that in every of these cases \mathcal{I} is a model of α . \diamond

The converse of Proposition 7 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 S . This axiom, however, is not syntactically local w.r.t. $S = \{A, B\}$ since it involves symbols in 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. $S = \{R\}$ ($\exists R. \top \sqsubseteq \exists R. \top$ is a tautology), but not syntactically

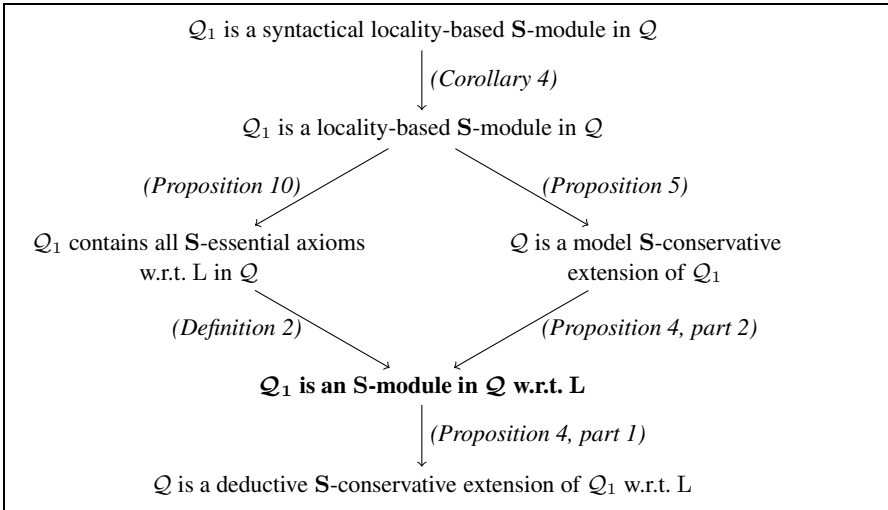


Fig. 6.3. Summary for the main theoretical results of the chapter

local. Thus, the limitation of syntactic locality is its inability to perform reasoning on 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 4. *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 6.3 the relationships between the key theoretical results in this chapter.

6.5.3 Computing Locality-Based Modules

Recall that, according to Definition 5, 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. $\mathbf{S} \cup \text{Sig}(\mathcal{Q}_1)$. Algorithm 1 outlines a simple procedure which performs this task. Given an effective locality test $\text{locality_test}(\alpha, \mathbf{S})$ (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. $\mathbf{S} \cup \text{Sig}(\mathcal{Q}_1)$ until no such axioms are left in \mathcal{Q}_2 .

In Table 6.2 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 6.1 and \mathbf{S} is a signature $\mathbf{S} = \{\text{Cystic_Fibrosis}, \text{Genetic_Disorder}\}$. 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.

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

Input:

\mathcal{Q} : ontology

\mathbf{S} : signature

Output:

\mathcal{Q}_1 : a locality-based \mathbf{S} -module in \mathcal{Q}

```

1:  $\mathcal{Q}_1 \leftarrow \emptyset$   $\mathcal{Q}_2 \leftarrow \mathcal{Q}$ 
2: while not empty( $\mathcal{Q}_2$ ) do
3:    $\alpha \leftarrow \text{select\_axiom}(\mathcal{Q}_2)$ 
4:   if locality_test( $\alpha, \mathbf{S} \cup \text{Sig}(\mathcal{Q}_1)$ ) 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  $\mathcal{Q}_1$ 

```

Table 6.2. A trace of Algorithm 1 for the input $\mathcal{Q} = \{\mathcal{M}1, \dots, \mathcal{M}5\}$ and $\mathbf{S} = \{\text{Cystic_Fibrosis}, \text{Genetic_Disorder}\}$

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

Proposition 8 (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, and (2) the output $\text{extract_module}(\mathbf{S}, \mathcal{Q})$ 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 local w.r.t. $\mathbf{S} \cup \text{Sig}(\mathcal{Q}_1)$, 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 $\mathbf{S} \cup \text{Sig}(\mathcal{Q}_1)$ does not change. \diamond

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 6.3 makes a different choice for α from \mathcal{Q}_2 than the trace in Table 6.2. In the first iteration of the while loop, we select $\alpha = \mathcal{M}2$ from \mathcal{Q}_2 instead of $\mathcal{M}1$ as in Table 6.2. This has resulted in a longer trace yet with the same result $\mathcal{Q}_1 = \{\mathcal{M}1, \dots, \mathcal{M}4\}$. Note that axioms $\mathcal{M}2$ and $\mathcal{M}3$ 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 6.2 and Table 6.3, 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 7. We say that a locality test $\text{locality_test}(\alpha, \mathbf{S})$ is anti-monotonic if for every $\mathbf{S}_1 \subseteq \mathbf{S}_2$, whenever $\text{locality_test}(\alpha, \mathbf{S}_2)$ succeeds then $\text{locality_test}(\alpha, \mathbf{S}_1)$ succeeds as well.

Table 6.3. An alternative trace of Algorithm 1 for the input $Q = \{\mathcal{M}1, \dots, \mathcal{M}5\}$ and $S = \{\text{Cystic_Fibrosis}, \text{Genetic_Disorder}\}$

	Q_1	Q_2	New elements in $S \cup \text{Sig}(Q_1)$	α	loc.
1	–	$\mathcal{M}1 - \mathcal{M}5$	Cystic_Fibrosis, Genetic_Disorder	$\mathcal{M}2$	Yes
2	–	$\mathcal{M}1, \mathcal{M}3 - \mathcal{M}5$	–	$\mathcal{M}3$	Yes
3	–	$\mathcal{M}1, \mathcal{M}4, \mathcal{M}5$	–	$\mathcal{M}1$	No
4	$\mathcal{M}1$	$\mathcal{M}2 - \mathcal{M}5$	Fibrosis, located_In, Pancreas, has_Origin, Genetic_Origin	$\mathcal{M}3$	No
5	$\mathcal{M}1, \mathcal{M}3$	$\mathcal{M}2, \mathcal{M}4, \mathcal{M}5$	Genetic_Fibrosis	$\mathcal{M}4$	No
6	$\mathcal{M}1, \mathcal{M}3, \mathcal{M}4$	$\mathcal{M}2, \mathcal{M}5$	–	$\mathcal{M}5$	Yes
7	$\mathcal{M}1, \mathcal{M}3, \mathcal{M}4$	$\mathcal{M}2$	–	$\mathcal{M}2$	No
8	$\mathcal{M}1 - \mathcal{M}4$	$\mathcal{M}5$	–	$\mathcal{M}5$	Yes
9	$\mathcal{M}1 - \mathcal{M}4$	–	–	–	

We say that locality of Q_1 w.r.t. S in Q_1 is provable using $\text{locality_test}(\alpha, S)$ if for every $\alpha \in Q \setminus Q_1$, we have that $\text{locality_test}(\alpha, S \cup \text{Sig}(S_1))$ succeeds. \diamond

Proposition 9 (Determinism of Algorithm 1)

The output of Algorithm 1 based on anti-monotonic $\text{locality_test}(\alpha, S)$ is the smallest Q_1 such that locality of Q_1 w.r.t. S is provable using $\text{locality_test}(\alpha, S)$.

Proof. It is easy to see (see the proof of Proposition 8) that the locality of every output Q_1 of Algorithm 1 is provable using $\text{locality_test}(\alpha, S)$. It remains, thus, to show that for every subset $Q'_1 \subseteq Q$ such that locality of Q'_1 w.r.t. S in Q is provable using $\text{locality_test}(\alpha, S)$, we have $Q_1 \subseteq Q'_1$.

Assume, to the contrary, that for some run of the algorithm, the output Q_1 is not a subset of Q'_1 . Since the initial $Q_1 = \emptyset$ was a subset of Q'_1 , there is a moment in the computation such that Q_1 was a subset of Q'_1 , but $Q_1 \cup \{\alpha\}$ is no longer a subset of Q'_1 . For these particular values of Q_1 and α we have: (i) $Q_1 \subseteq Q'_1$, (ii) $\alpha \in Q \setminus Q'_1$, and (iii) $\text{locality_test}(\alpha, S \cup \text{Sig}(Q_1))$ fails. From (ii) by property of Q'_1 we have $\text{locality_test}(\alpha, S \cup \text{Sig}(Q'_1))$ succeeds, which implies using (i) and anti-monotonicity of locality_test that $\text{locality_test}(\alpha, S \cup \text{Sig}(Q_1))$ succeeds which contradicts to (iii). This proves that Q_1 is indeed a subset of Q'_1 . \diamond

Corollary 5 (Uniqueness of a Minimal Locality-Based Module)

Algorithm 1 using a test based on the semantic locality produces a unique minimal locality-based S -module in Q .

Proof. By Lemma 2 the semantic locality admits anti-monotonicity. \diamond

Corollary 6 (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 S -module in Q .

Proof. By Lemma 3 the syntactic locality admits anti-monotonicity. \diamond

6.5.4 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 Q_S^{loc} be the smallest locality-based S -module in Q , which is unique by Corollary 5 and is the output of Algorithm 1 for Q and S . The first property is a direct consequence of Corollary 5:

Proposition 10. Q_S^{loc} contains all S -essential axioms in Q w.r.t. every language L with Tarski-style set-theoretic semantics.

Proof. Let Q_1 be a minimal S -module in Q . We need to show that $Q_1 \subseteq Q_S^{loc}$. Since (i) Q_1 is a subset of a locality-based S -module in Q (say, of Q itself) and (ii) there is no proper subset of Q_1 that is a locality-based S -module in Q , we have that Q_1 is a subset of a minimal locality-based S -module in Q . Since such a module is unique by Corollary 5, and it is Q_S^{loc} , we have that $Q_1 \subseteq Q_S^{loc}$. \diamond

As shown in Table 6.2 and Table 6.3, the minimal locality-based S -module extracted from Q contains all S -essential axioms $\mathcal{M}1$ – $\mathcal{M}4$. 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 (6.4).

Proposition 11. Let Q be ontology, A and B atomic concepts and $S_{(i)}$ a signature. Then:

1. $S_1 \subseteq S_2$ implies $Q_{S_1}^{loc} \subseteq Q_{S_2}^{loc}$ (monotonicity);
2. $Q \models (A \sqsubseteq B)$ iff $Q_{\{A\}}^{loc} \models (A \sqsubseteq B)$.
3. $Q \models (A \sqsubseteq B)$ implies $Q_{\{B\}}^{loc} \subseteq Q_{\{A\}}^{loc}$ or $Q_{\{A\}}^{loc} \models A \sqsubseteq \perp$.

Proof

1. Since $Q_{S_2}^{loc}$ is a locality-based S_2 -module in Q , we have $Q \setminus Q_{S_2}^{loc}$ is local w.r.t. $S_2 \cup \text{Sig}(Q_{S_2}^{loc})$. By anti-monotonicity of locality (see Lemma 2), $Q \setminus Q_{S_2}^{loc}$ is local w.r.t. $S_1 \cup \text{Sig}(Q_{S_2}^{loc})$, hence $Q_{S_2}^{loc}$ is a locality-based S_1 -module in Q . Since $Q_{S_1}^{loc}$ is contained in every locality-based S_1 -module in Q by Corollary 5, we have $Q_{S_1}^{loc} \subseteq Q_{S_2}^{loc}$.

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

(a) $B \in S$. Then by Remark 1, $Q_{\{A\}}^{loc}$ is an S -module in Q , and so, $Q_{\{A\}}^{loc} \models (A \sqsubseteq B)$ since $\text{Sig}(A \sqsubseteq B) \subseteq S$.

(b) $B \notin S$. We demonstrate that $Q_{\{A\}}^{loc} \models A \sqsubseteq \perp$ which suffices for proving $Q_{\{A\}}^{loc} \models A \sqsubseteq B$.

Assume, to the contrary, that $Q_{\{A\}}^{loc} \not\models A \sqsubseteq \perp$. Then there exists an S -interpretation \mathcal{I} such that $\mathcal{I} \models Q_{\{A\}}^{loc}$ and $A^{\mathcal{I}} \neq \emptyset$. Let \mathcal{I}' be a trivial expansion of

\mathcal{I} to $\mathbf{S} \cup \text{Sig}(\mathcal{Q})$. Since $\mathcal{Q}_{\{A\}}^{loc}$ is a locality-based \mathbf{S} -module in \mathcal{Q} (see Remark 1 and Remark 2), 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 $B \in \text{Sig}(\mathcal{Q}_{\{A\}}^{loc})$ or $\mathcal{Q}_{\{A\}}^{loc} \models A \sqsubseteq \perp$. So, it remains to show that $B \in \text{Sig}(\mathcal{Q}_{\{A\}}^{loc})$ implies that $\mathcal{Q}_{\{B\}}^{loc} \subseteq \mathcal{Q}_{\{A\}}^{loc}$. Indeed, by Remark 1, $\mathcal{Q}_{\{A\}}^{loc}$ is a locality-based $(\text{Sig}(\mathcal{Q}_{\{A\}}^{loc}) \cup \{A\})$ -module in \mathcal{Q} . Since $B \in \text{Sig}(\mathcal{Q}_{\{A\}}^{loc})$, then, in particular, $\mathcal{Q}_{\{A\}}^{loc}$ is a locality-based $\{B\}$ -module in \mathcal{Q} . Since $\mathcal{Q}_{\{B\}}^{loc}$ is contained in every locality-based $\{B\}$ -module in \mathcal{Q} , we have $\mathcal{Q}_{\{B\}}^{loc} \subseteq \mathcal{Q}_{\{A\}}^{loc}$ what was required to prove. \diamond

Proposition 11 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 $\mathcal{Q}_{\{A\}}^{loc}$ 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 $\mathcal{Q}_{\{A\}}^{loc}$ and, hence, it will be obtained in step (2).

Finally, Property 3 in Proposition 10 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 $\mathcal{Q}_{\{B\}}^{loc} \subseteq \mathcal{Q}_{\{A\}}^{loc}$ 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.

We have used these properties of locality-based modules for optimizing incremental classification—that is, to improve the efficiency of DL reasoners under changes in the ontology. We refer the interested reader to [8] for details.

6.6 Implementation and Evaluation

In this section, we show that minimal locality-based modules obtained from 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 [15];
- A2: The algorithm for extracting modules described in [6];
- A3: Our algorithm for extracting modules (Algorithm 1), based on 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:

³ <http://sourceforge.net/projects/owlapi>

Simple. In this group, we have included the National Cancer Institute (NCI) Ontology,⁴ the SUMO Upper Ontology,⁵ and the Gene Ontology (GO),⁶ These ontologies are expressed in a simple ontology language and are of a simple structure; in particular, they do not contain GCIs, but only definitions.

Complex. This group contains the well-known GALEN ontology (GALEN-Full),⁷ the DOLCE upper ontology (DOLCE-Lite),⁸ and NASA’s Semantic Web for Earth and Environmental Terminology (SWEET)⁹. 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.

Table 6.4. Comparison of Different Modularization Algorithms

Ontology	# Atomic Concepts	A1: Prompt-Factor		A2: Mod. in [6]		A3: Loc.-based mod.	
		Max.(%)	Avg.(%)	Max.(%)	Avg.(%)	Max.(%)	Avg.(%)
NCI	27772	87.6	75.84	55	30.8	0.8	0.08
GO	22357	1	0.1	1	0.1	0.4	0.05
SUMO	869	100	100	100	100	2	0.09
GALEN-Small	2749	100	100	100	100	10	1.7
GALEN-Full	24089	100	100	100	100	29.8	3.5
SWEET	1816	96.4	88.7	83.3	51.5	1.9	0.1
DOLCE-Lite	499	100	100	100	100	37.3	24.6

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, as seen before, modules for atomic concepts are especially interesting for optimized classification of ontologies.

The results we have obtained are summarized in Table 6.4. 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, and GALEN the algorithms A1 and A2 retrieve the whole ontology as the module for each atomic concept. In contrast, the modules

⁴ <http://www.mindswap.org/2003/CancerOntology/nciOncology.owl>

⁵ <http://ontology.teknowledge.com/>

⁶ <http://www.geneontology.org>

⁷ http://www.openclinical.org/prj_galen.html

⁸ <http://www.loa-cnr.it/DOLCE.html>

⁹ <http://sweet.jpl.nasa.gov/ontology/>

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.

For NCI, 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 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. 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.

6.7 Conclusion

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.

References

1. Baader, F., Brandt, S., Lutz, C.: Pushing the \mathcal{EL} envelope. In: Proc. IJCAI 2005, pp. 364–370 (2005)
2. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, Cambridge (2003)
3. Börger, E., Grädel, E., Gurevich, Y.: The Classical Decision Problem. In: Perspectives of Mathematical Logic. Springer, Heidelberg (1997); Second printing (Universitext) 2001
4. Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U.: A logical framework for modularity of ontologies. In: Proc. IJCAI 2007, pp. 298–304 (2007)
5. Cuenca Grau, B., Horrocks, I., Kutz, O., Sattler, U.: Will my Ontologies Fit Together? In: Proc. DL 2006 (2006)
6. Cuenca Grau, B., Parsia, B., Sirin, E., Kalyanpur, A.: Modularity and Web Ontologies. In: Proc. KR 2006, pp. 198–209 (2006)
7. Ghilardi, S., Lutz, C., Wolter, F.: Did I Damage my Ontology? A Case for Conservative Extensions in Description Logics. In: Proc. KR 2006, pp. 187–197 (2006)
8. Cuenca Grau, B., Halaschek-Wiener, C., Kazakov, Y.: History matters: Incremental ontology reasoning using modules. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 183–196. Springer, Heidelberg (2007)

9. Horrocks, I., Patel-Schneider, P.F., van Harmelen, F.: From *SHIQ* and RDF to OWL: The making of a web ontology language. *J. of Web Semantics* 1(1), 7–26 (2003)
10. Horrocks, I., Sattler, U.: A tableaux decision procedure for SHOIQ. In: Proc. of the IJCAI. Morgan Kaufman, San Francisco (2005)
11. Kalyanpur, A., Parsia, B., Sirin, E., Cuenca Grau, B., Hendler, J.: SWOOP: A web editing browser. *Elsevier's Journal Of Web Semantics* 4(2), 144–153 (2006)
12. Lutz, C., Walther, D., Wolter, F.: Conservative extensions in expressive description logics. In: Proc. of IJCAI 2007, pp. 453–459 (2007)
13. Möller, R., Haarslev, V.: Description logic systems. In: *The Description Logic Handbook*, ch. 8, pp. 282–305. Cambridge University Press, Cambridge (2003)
14. Motik, B.: Reasoning in Description Logics using Resolution and Deductive Databases. PhD thesis, Univesität Karlsruhe (TH), Karlsruhe, Germany (2006)
15. Noy, N.F., Musen, M.A.: The PROMPT suite: Interactive tools for ontology mapping and merging. *Int. Journal of Human-Computer Studies* 6(59) (2003)
16. Patel-Schneider, P.F., Hayes, P., Horrocks, I.: Web ontology language OWL Abstract Syntax and Semantics. W3C Recommendation (2004)
17. Schmidt-Schauß, M., Smolka, G.: Attributive concept descriptions with complements. *Artif. Intell.* 48(1), 1–26 (1991)
18. Sirin, E., Parsia, B.: Pellet system description. In: Proc. DL 2004 (2004)
19. Tsarkov, D., Horrocks, I.: FaCT++ description logic reasoner: System description. In: Furbach, U., Shankar, N. (eds.) IJCAR 2006. LNCS, vol. 4130, pp. 292–297. Springer, Heidelberg (2006)