Minimal Module Extraction from DL-Lite Ontologies using QBF Solvers

R. Kontchakov,[1] L. Pulina,[2] U. Sattler,[3] T. Schneider,[3] P. Selmer,[1] F. Wolter[4] and M. Zakharyaschev[1]
[1]School of Computer Science and Information Systems, Birkbeck College London
[2]Dipartimento di Informatica Sistemistica e Telematica, University of Genoa
[3] School of Computer Science, University of Manchester
[4]Department of Computer Science, University of Liverpool

Abstract

We present a formal framework for (minimal) module extraction based on an abstract notion of inseparability w.r.t. a signature between ontologies. Two instances of this framework are discussed in detail for *DL-Lite* ontologies: concept inseparability, when ontologies imply the same complex concept inclusions over the signature, and query inseparability, when they give the same answers to existential queries for any instance data over the signature. We demonstrate that different types of corresponding minimal modules for these inseparability relations can be automatically extracted from large-scale *DL-Lite* ontologies by composing the tractable syntactic locality-based module extraction algorithm with intractable extraction algorithms using the multi-engine QBF solver AQME. The extracted minimal modules are compared with those obtained using non-logic-based approaches.

1 Introduction

In computer science, ontologies are used to provide a common vocabulary (used here synonymously with 'signature') for a domain of interest, together with a description of the relationships between terms built from the vocabulary. Module extraction—the problem of finding a (minimal) subset of a given ontology that provides the same description of the relationships between terms over a given sub-vocabulary as the whole ontology—has recently become an active research topic in various ontology-related areas such as the semantic web and description logic; see, e.g., the recent volume on ontology modularisation [Parent *et al.*, 2009] and the WoMO workshop series devoted to this problem [Haase *et al.*, 2006; Cuenca-Grau *et al.*, 2007]. The reasons for this are manifold, with one of the most important being ontology re-use. It is often impossible and not even desirable to develop an entirely new ontology for every new application; a better methodology is to re-use appropriate existing ontologies. However, typically only a relatively small part of the vocabulary of a possibly large ontology is required, that is, one only needs a subset, or module, of the ontology that gives the same description of this sub-vocabulary. Extracting such a module is the problem we are concerned with in this paper.

The phrase 'gives the same description of the vocabulary' (formalised below as a family of equivalence relations, one for each signature, and called an *inseparability relation*) is rather vague and has been interpreted in a variety of different ways, ranging from structural approaches [Noy and Musen, 2004; Seidenberg and Rector, 2006] to logic-based approaches [Cuenca-Grau *et al.*, 2006; 2008; Konev *et al.*, 2008]. While structural approaches use, and depend on, the syntax of the axioms of ontologies and mostly only take into account the induced is-a hierarchy, logic-based approaches consider the *consequences* of ontologies and require these to be the same for the relevant vocabulary. Although theoretically attractive and elegant, the logic-based approaches suffer from the high computational complexity of the problems to be solved: even checking whether two ontologies imply the same concept inclusions over a given signature is typically one exponential harder than standard reasoning problems (e.g., for $\mathcal{ALC}$ ontologies this problem is 2EXPTIME-complete [Ghilardi *et al.*, 2006]). In [Cuenca-Grau *et al.*, 2008], this difficulty has been addressed by developing a tractable (even for $\mathcal{SHIQ}$) syntactic locality-based module extraction algorithm, the only disadvantage of which is that the extracted modules are typically *not minimal*. The only existing practical (and tractable) logic-based algorithm capable of extracting minimal modules was developed for *acyclic* $\mathcal{EL}$-ontologies [Konev *et al.*, 2008].

The main aim of this paper is to introduce a framework for module extraction based on the notion of inseparability between ontologies and to demonstrate that a purely logic-based approach to minimal module extraction is feasible in practice for large-scale *DL-Lite*$_{bool}$ ontologies.

The *DL-Lite* family of description logics [Calvanese *et al.*, 2005; 2006] has been originally designed with the aim of providing query access to large amounts of data via a high level conceptual (ontological) interface. Thus, the *DL-Lite* logics resulted from various compromises between the necessity of keeping the complexity of query answering low and the desire of having the expressive means for representing various constraints of data modelling formalisms such as the ER model and UML class diagrams [Artale *et al.*, 2007b]. For example, the logic *DL-Lite*$_{bool}$ [Artale *et al.*, 2007a] (containing many other *DL-Lite* logics) can express is-a hierarchies of concepts; disjointness and covering constraints for concepts; domain, range and cardinality constraints for roles;

and multiplicity constraints for attributes. Therefore, standard reasoning in *DL-Lite$_{bool}$* (say, testing concept satisfiability) is NP-complete and, similarly to $\mathcal{ALC}$, the main reasoning tasks required for module extraction are even harder: deciding whether two *DL-Lite$_{bool}$* ontologies imply the same concept inclusions over a given signature or whether they give the same answers to conjunctive queries for arbitrary ABoxes over this signature is Π_2^p-complete [Kontchakov *et al.*, 2008].

The contribution of this paper is as follows. We present generic algorithms extracting minimal logic-based modules from *DL-Lite$_{bool}$* ontologies which call (quadratically often, in the worst case) an oracle deciding whether an appropriate inseparability relation holds between *DL-Lite$_{bool}$* ontologies. In our experiments with two large-scale *DL-Lite$_{bool}$* ontologies, the oracle is realised using encodings into quantified Boolean formulas (QBFs) and solving these with the self-adaptive multi-engine QBF solver AQME [Pulina and Tacchella, 2009]. A significant speed-up is achieved by first extracting the (typically non-minimal) locality-based module and then applying to it the above mentioned algorithms. Finally, we provide a comparison of the sizes of the modules extracted for various logic-based notions of modules as well as existing structure-based notions of modules.

2 Inseparability modules

We begin by introducing an abstract notion of inseparability between ontologies w.r.t. a given signature and by investigating corresponding minimal module extraction algorithms. This notion, as well as the algorithms, do not depend on the underlying ontology language. However, to be precise, we introduce it for ontologies given as *DL-Lite$_{bool}$* TBoxes only. The language of *DL-Lite$_{bool}$* TBoxes is based on *concept names* $A_1, A_2, \dots$ and *role names* $P_1, P_2, \dots$, with complex *roles* R and *concepts* C defined as follows:

$$\begin{array}{lcl} R & ::= & P_i \quad | \quad P_i^-, \\ B & ::= & \bot \quad | \quad \top \quad | \quad A_i \quad | \quad \geq q\, R, \\ C & ::= & B \quad | \quad \neg C \quad | \quad C_1 \sqcap C_2, \end{array}$$

where $q \geq 1$. (Other concept constructs like $\exists R$, $\leq q\, R$ and $C_1 \sqcup C_2$ will be used as standard abbreviations.) As usual, a *concept inclusion* is of the form $C_1 \sqsubseteq C_2$, where C_1 and C_2 are concepts, and a *DL-Lite$_{bool}$ TBox* is a finite set of concept inclusions. A *signature* is a finite set of concept and role names. Given a concept, role, concept inclusion or TBox E, we denote by $\mathsf{sig}(E)$ the *signature* of E, that is, the set of concept and role names that occur in E.

An *inseparability relation* $S = \{\equiv_\Sigma^S \mid \Sigma$ a signature$\}$ is a family of equivalence relations $\equiv_\Sigma^S$ on the set of *DL-Lite$_{bool}$* TBoxes. Intuitively, $\mathcal{T}_1 \equiv_\Sigma^S \mathcal{T}_2$ means that $\mathcal{T}_1$ and $\mathcal{T}_2$ are indistinguishable w.r.t. (or give the same description of) the signature Σ. We call an inseparability relation S *monotone* if it satisfies the following conditions, for all TBoxes $\mathcal{T}_1$, $\mathcal{T}_2$, signatures Σ and $\equiv_\Sigma^S$ in S:

(M_{sig}) if $\mathcal{T}_1 \equiv_\Sigma^S \mathcal{T}_2$ then $\mathcal{T}_1 \equiv_{\Sigma'}^S \mathcal{T}_2$, for every $\Sigma' \subseteq \Sigma$;
(M_{T}) if $\mathcal{T}_1 \subseteq \mathcal{T}_2 \subseteq \mathcal{T}_3$ and $\mathcal{T}_1 \equiv_\Sigma^S \mathcal{T}_3$, then $\mathcal{T}_1 \equiv_\Sigma^S \mathcal{T}_2$.

Condition (M_{sig}) formalises the intuition that if two TBoxes are indistinguishable w.r.t. a certain signature, then they are indistinguishable w.r.t. any smaller signature; (M_{T}) demands that any TBox sandwiched between two indistinguishable TBoxes should be indistinguishable from either of them.

We now introduce three distinct notions of modules induced by an inseparability relation.

Definition 1 Let S be an inseparability relation, $\mathcal{T}$ a TBox, $\mathcal{M} \subseteq \mathcal{T}$, and Σ a signature. We say that $\mathcal{M}$ is

- an *S_Σ-module of $\mathcal{T}$* if $\mathcal{M} \equiv_\Sigma^S \mathcal{T}$;
- a *self-contained S_Σ-module of $\mathcal{T}$* if $\mathcal{M} \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})}^S \mathcal{T}$;
- a *depleting S_Σ-module of $\mathcal{T}$* if $\emptyset \equiv_{\Sigma \cup \mathsf{sig}(\mathcal{M})}^S \mathcal{T} \setminus \mathcal{M}$.

$\mathcal{M}$ is a *minimal* (*self-contained*, *depleting*) *S_Σ-module of $\mathcal{T}$* if $\mathcal{M}$ is a (self-contained, depleting) S_Σ-module of $\mathcal{T}$, but no proper subset of $\mathcal{M}$ is such an S_Σ-module of $\mathcal{T}$.

Clearly, every self-contained S_Σ-module is an S_Σ-module; see below for concrete examples showing that no other inclusion between these different types of modules holds in general. We start our investigation by considering minimal S_Σ-modules. The following theorem presents a straightforward algorithm extracting *one* minimal S_Σ-module from a given TBox, using an oracle deciding S_Σ-inseparability.

Theorem 2 *Let S be an inseparability relation satisfying (M_{T}), $\mathcal{T}$ a TBox, and Σ a signature. Then the following algorithm computes a minimal S_Σ-module of $\mathcal{T}$:*

```
input $\mathcal{T}, \Sigma$
$\mathcal{M} := \mathcal{T}$
repeat
  $\mathcal{M}' := \mathcal{M}$
  for each $\alpha \in \mathcal{M}'$ do
    if $\mathcal{M} \setminus \{\alpha\} \equiv_\Sigma^S \mathcal{M}$ then $\mathcal{M} := \mathcal{M} \setminus \{\alpha\}$
  end for
until $\mathcal{M}' = \mathcal{M}$
output $\mathcal{M}$
```

Proof The algorithm computes an S_Σ-module $\mathcal{M}$ of $\mathcal{T}$ such that $\mathcal{M} \setminus \{\alpha\}$ is not an S_Σ-module of $\mathcal{T}$, for any $\alpha \in \mathcal{M}$. By (M_{T}), no proper subset of $\mathcal{M}$ is an S_Σ-module. ❑

Note that the minimal S_Σ-module extracted by this algorithm depends on the order of picking the axioms α. There exist natural inseparability relations (see below) for which there are exponentially many distinct minimal S_Σ-modules. Consider, e.g., an ontology $\{\alpha_1, \beta_1, \dots, \alpha_n, \beta_n\}$, where α_i and β_i are syntactically different but S_Σ-inseparable axioms.

Denote by $|\mathcal{T}|$ the size of $\mathcal{T}$, that is, the number of occurrences of symbols in it. Then the algorithm runs in quadratic time in $|\mathcal{T}|$ calling an oracle deciding $\equiv_\Sigma^S$ at most $|\mathcal{T}|^2$ times.

Self-contained S_Σ-modules are indistinguishable from the original TBox not only w.r.t. Σ but also w.r.t. their own signature. To discuss depleting S_Σ-modules, we require two additional conditions on inseparability relations.

Definition 3 We say that an inseparability relation S is

- *robust under replacement* if, for all TBoxes $\mathcal{T}$, $\mathcal{T}_1$, $\mathcal{T}_2$ and signatures Σ, we have $\mathcal{T}_1 \cup \mathcal{T} \equiv_\Sigma^S \mathcal{T}_2 \cup \mathcal{T}$ whenever $\mathcal{T}_1 \equiv_\Sigma^S \mathcal{T}_2$ and $\mathsf{sig}(\mathcal{T}) \cap \mathsf{sig}(\mathcal{T}_1 \cup \mathcal{T}_2) \subseteq \Sigma$;
- *robust under vocabulary extensions* if, for all TBoxes $\mathcal{T}_1$, $\mathcal{T}_2$ and signatures $\Sigma \subseteq \Sigma'$ such that $\mathsf{sig}(\mathcal{T}_1 \cup \mathcal{T}_2) \cap \Sigma' \subseteq \Sigma$ we have $\mathcal{T}_1 \equiv_{\Sigma'}^S \mathcal{T}_2$ whenever $\mathcal{T}_1 \equiv_\Sigma^S \mathcal{T}_2$.

Robustness is fundamental for ontology re-use. Suppose an ontology developer imports an S_Σ-module $\mathcal{M}$ of a TBox $\mathcal{T}$ into her own ontology $\mathcal{O}$ because she is interested in the relations between terms over Σ defined by $\mathcal{T}$. Then, if S is robust under replacement and vocabulary extensions, we have $\mathcal{O} \cup \mathcal{T} \equiv^S_{\Sigma'} \mathcal{O} \cup \mathcal{M}$ for every signature Σ' such that $\Sigma' \cap \mathsf{sig}(\mathcal{T}) \subseteq \Sigma$ and $\mathsf{sig}(\mathcal{T}) \cap \mathsf{sig}(\mathcal{O}) \subseteq \Sigma'$. Thus, these properties ensure that it does not make any difference whether she imports $\mathcal{T}$ or some S_Σ-module $\mathcal{M}$ of $\mathcal{T}$ into $\mathcal{O}$, and this does not depend on $\mathcal{O}$.

Proposition 4 *If S is an inseparability relation that is robust under replacement, then every depleting S_Σ-module is a self-contained S_Σ-module.*

Proof If $\mathcal{T} \setminus \mathcal{M} \equiv^S_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset$, robustness under replacement implies $\mathcal{T} = (\mathcal{T} \setminus \mathcal{M}) \cup \mathcal{M} \equiv^S_{\Sigma \cup \mathsf{sig}(\mathcal{M})} \emptyset \cup \mathcal{M} = \mathcal{M}$. ❑

We will consider depleting S_Σ-modules only if S is robust under replacement and, therefore, only if they are self-contained modules as well.

Theorem 5 *Let S be a monotone inseparability relation that is robust under replacement, $\mathcal{T}$ a TBox, and Σ a signature. Then there is a unique minimal depleting S_Σ-module of $\mathcal{T}$, which is computed by the following algorithm:*

```
input $\mathcal{T},\Sigma$
$\mathcal{T}' := \mathcal{T}$; $\Gamma := \Sigma$; $\mathcal{W} := \emptyset$
while $\mathcal{T}' \setminus \mathcal{W} \neq \emptyset$ do
  choose $\alpha \in \mathcal{T}' \setminus \mathcal{W}$
  $\mathcal{W} := \mathcal{W} \cup \{\alpha\}$
  if $\mathcal{W} \not\equiv^S_\Gamma \emptyset$ then
    $\mathcal{T}' := \mathcal{T}' \setminus \{\alpha\}$; $\mathcal{W} := \emptyset$; $\Gamma := \Gamma \cup \mathsf{sig}(\alpha)$
  endif
end while
output $\mathcal{T} \setminus \mathcal{T}'$
```

Proof Let $\mathcal{M}$ be a minimal depleting S_Σ-module of $\mathcal{T}$. The crucial observation for proving correctness of the algorithm and uniqueness of $\mathcal{M}$ is that, if $\mathcal{T}_0 \subseteq \mathcal{T}$ is minimal with $\mathcal{T}_0 \not\equiv^S_\Sigma \emptyset$, then $\mathcal{T}_0 \subseteq \mathcal{M}$. Suppose this claim does not hold, i.e., $\mathcal{X} \equiv^S_\Sigma \emptyset$, where $\mathcal{X} = \mathcal{M} \cap \mathcal{T}_0$. By robustness under replacement, $(\mathcal{T} \setminus \mathcal{M}) \cup \mathcal{X} \equiv^S_{\mathsf{sig}(\mathcal{M}) \cup \Sigma} \mathcal{X}$. By (M_{T}), $\emptyset \subseteq \mathcal{T}_0 \subseteq (\mathcal{T} \setminus \mathcal{M}) \cup \mathcal{X}$ implies $\mathcal{T}_0 \equiv^S_{\mathsf{sig}(\mathcal{M}) \cup \Sigma} \mathcal{X}$, and so, by (M_{sig}), $\mathcal{T}_0 \equiv^S_\Sigma \emptyset$, which is a contradiction. ❑

The algorithm above computes *the* minimal depleting S_Σ-module in quadratic time by calling the oracle deciding $\equiv^S_\Sigma$-inseparability at most $|\mathcal{T}|^2$ times.

By the result above, minimal depleting modules have the advantage of being uniquely determined (under mild conditions), which sharply contrasts with the behaviour of the other types of modules. Another advantage is that depleting modules support modular ontology development in the following sense. Suppose $\mathcal{M}$ is a depleting S_Σ-module of $\mathcal{T}$ and S is robust under replacement and vocabulary extensions. Then one can import into the ontology $\mathcal{T} \setminus \mathcal{M}$ any module $\mathcal{M}'$ such that $\mathsf{sig}(\mathcal{M}') \cap \mathsf{sig}(\mathcal{T}) \subseteq \Sigma \cup \mathsf{sig}(\mathcal{M})$ and be sure that $\mathcal{T} \setminus \mathcal{M}$ does not interfere with $\mathcal{M}'$; i.e., $(\mathcal{T} \setminus \mathcal{M}) \cup \mathcal{M}' \equiv^S_{\Sigma'} \mathcal{M}'$ whenever $\Sigma' \cap \mathsf{sig}(\mathcal{T} \setminus \mathcal{M}) \subseteq \Sigma \cup \mathsf{sig}(\mathcal{M})$. The importance of this property was first pointed out in [Cuenca-Grau *et al.*, 2008] in the context of conservative extensions and modularity.

3 Two inseparability relations in *DL-Lite$_{bool}$*

The framework and algorithms presented above can be instantiated with ontologies in any standard DL and with numerous different choices for inseparability relations. Here we consider two inseparability relations between *DL-Lite$_{bool}$* ontologies which have both been introduced and investigated in [Kontchakov *et al.*, 2008]. To give precise definitions of these relations, we require some notation. A *DL-Lite$_{bool}$ ABox $\mathcal{A}$* is a set of assertions of the form $C(a_i)$, $R(a_i, a_j)$, where C is a concept, R a role, and a_i, a_j are object names from an infinite list of *object names* $a_1, a_2, \ldots$. A *DL-Lite$_{bool}$ knowledge base* (*KB*, for short) is a pair $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ with a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$. The semantic notions of an interpretation, $\mathcal{I}$, a concept inclusion $C_1 \sqsubseteq C_2$ being satisfied in $\mathcal{I}$, and of $\mathcal{I}$ being a model of a TBox, ABox and KB are standard and can be found in [Kontchakov *et al.*, 2008]. A concept inclusion $C_1 \sqsubseteq C_2$ *follows from* $\mathcal{T}$, $\mathcal{T} \models C_1 \sqsubseteq C_2$ in symbols, if every model of $\mathcal{T}$ satisfies $C_1 \sqsubseteq C_2$. An *essentially positive existential query* (simply a *query*) is a first-order formula

$$q(x_1, \ldots, x_n) = \exists y_1 \ldots \exists y_m \varphi(x_1, \ldots, x_n, y_1, \ldots, y_m),$$

where φ is constructed from atoms of the form $C(t)$ and $R(t_1, t_2)$, with C being an *DL-Lite$_{bool}$*-concept, R a role, and t_i being either an object name or a variable from the list $x_1, \ldots, x_n, y_1, \ldots, y_m$, using only $\wedge$ and $\vee$. Given a KB $\mathcal{K}$ and a query $q(\boldsymbol{x})$, $\boldsymbol{x} = x_1, \ldots, x_n$, we say that an n-tuple $\boldsymbol{a}$ of object names is an *answer* to $q(\boldsymbol{x})$ w.r.t. $\mathcal{K}$ and write $\mathcal{K} \models q(\boldsymbol{a})$ if, for every model $\mathcal{I}$ of $\mathcal{K}$, we have $\mathcal{I} \models q(\boldsymbol{a})$.

Given an ABox or query E, we denote by $\mathsf{sig}(E)$ the *signature* of E, that is, the set of concept and role names that occur in E. A concept (role, concept inclusion, TBox, ABox, query) E is called a Σ-*concept* (*role*, *concept inclusion*, *TBox*, *ABox*, *query*, respectively) if $\mathsf{sig}(E) \subseteq \Sigma$.

Definition 6 Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be *DL-Lite$_{bool}$* TBoxes and Σ a signature. We say that $\mathcal{T}_1$ and $\mathcal{T}_2$ are Σ-*concept inseparable* and write $\mathcal{T}_1 \equiv^c_\Sigma \mathcal{T}_2$ if, for all Σ-concept inclusions $C \sqsubseteq D$ in *DL-Lite$_{bool}$*, we have $\mathcal{T}_2 \models C \sqsubseteq D$ iff $\mathcal{T}_1 \models C \sqsubseteq D$. The corresponding inseparability relation will be denoted by S^c.

$\mathcal{T}_1$ and $\mathcal{T}_2$ are said to be Σ-*query inseparable* ($\mathcal{T}_1 \equiv^q_\Sigma \mathcal{T}_2$, in symbols) if, for all Σ-ABoxes $\mathcal{A}$, Σ-queries $q(\boldsymbol{x})$ and tuples $\boldsymbol{a}$ of object names from $\mathcal{A}$, we have $(\mathcal{T}_1, \mathcal{A}) \models q(\boldsymbol{a})$ iff $(\mathcal{T}_2, \mathcal{A}) \models q(\boldsymbol{a})$. The corresponding inseparability relation will be denoted by S^q.

We believe that S^q, the inseparability relation which regards ontologies as indistinguishable w.r.t. Σ if they give the same answers to Σ-queries for any Σ-ABox $\mathcal{A}$, is the most appropriate inseparability relation for typical applications of *DL-Lite$_{bool}$* ontologies: recall that the design goal of *DL-Lite* was to provide a conceptual interface for querying instance data. As the data is usually not known in advance and may change, it is unrealistic to assume that the ABox is fixed when extracting a module: that is why in our approach we regard ABoxes as 'black boxes.' One could argue that instead of existential queries one should consider the smaller class of conjunctive queries when defining inseparability (and thus obtain smaller minimal modules). However, as shown in [Kontchakov *et al.*, 2008], when considering conjunctive queries instead of existential ones, one loses the robustness

(1) $\mathsf{Publisher} \sqsubseteq \exists \mathsf{pubHasDistrib}$

(2) $\exists \mathsf{pubHasDistrib}^- \sqsubseteq \mathsf{Distributor}$

(3) $\mathsf{Publisher} \sqsubseteq \neg \mathsf{Distributor}$

(4) $\exists \mathsf{pubHasDistrib} \sqsubseteq \mathsf{Publisher}$

(5) $\mathsf{Publisher} \sqsubseteq \leq 1\, \mathsf{pubHasDistrib}$

(6) $\mathsf{Role} \sqsubseteq \neg \mathsf{Distributor}$

(7) $\mathsf{User} \sqsubseteq \neg \mathsf{Distributor}$

(8) $\mathsf{Publisher} \sqsubseteq \exists \mathsf{pubAdmedBy}$

(9) $\exists \mathsf{pubAdmedBy}^- \sqsubseteq \mathsf{AdmUser} \sqcup \mathsf{BookUser}$

(10) $\mathsf{AdmUser} \sqsubseteq \mathsf{User}$

(11) $\mathsf{BookUser} \sqsubseteq \mathsf{User}$

(12) $\mathsf{User} \sqsubseteq \exists \mathsf{hasRole}$

(13) $\exists \mathsf{hasRole}^- \sqsubseteq \mathsf{Role}$

(14) $\mathsf{Role} \sqsubseteq \neg \mathsf{Publisher}$

(15) $\mathsf{User} \sqsubseteq \neg \mathsf{Publisher}$

(16) $\mathsf{Role} \sqsubseteq \neg \mathsf{User}$

(17) $\mathsf{User} \sqsubseteq \exists \mathsf{userAdmedBy}$

(18) $\exists \mathsf{userAdmedBy}^- \sqsubseteq \mathsf{AdmUser}$

(19) $\exists \mathsf{userAdmedBy} \sqsubseteq \mathsf{User}$

(20) $\exists \mathsf{pubAdmedBy} \sqsubseteq \mathsf{Publisher}$

Figure 1: 'Publisher' ontology $\mathcal{T}$.

properties of the corresponding inseparability relation. The next theorem follows from [Kontchakov *et al.*, 2008].

Theorem 7 *S^c and S^q are monotone and robust under vocabulary extensions; S^q is robust under replacement, S^c is not.*

It follows from Proposition 4 that depleting S^q-modules are self-contained S^q-modules. This implication fails for S^c:

Example 8 Let $\mathcal{T} = \{A \sqsubseteq \exists R,\ \exists R^- \sqsubseteq B,\ B \sqsubseteq \bot\}$, $\Sigma = \{A, B\}$ and $\mathcal{M} = \{B \sqsubseteq \bot\}$. Then $\mathcal{M}$ is a depleting S^c_Σ-module of $\mathcal{T}$ (because $\{A \sqsubseteq \exists R,\ \exists R^- \sqsubseteq B\} \equiv^c_\Sigma \emptyset$), but it is not a self-contained S^c_Σ-module of $\mathcal{T}$ (because $\mathcal{T} \models A \sqsubseteq \bot$).

For this reason we will not consider depleting S^c-modules in what follows. Note also that minimal depleting S^q_Σ-modules are in general larger than self-contained S^q_Σ-modules. Take, e.g., $\mathcal{T} = \{A \sqsubseteq B, A \sqsubseteq B \sqcap B\}$ and $\Sigma = \{A, B\}$. Then $\mathcal{M}_1 = \{A \sqsubseteq B\}$ and $\mathcal{M}_2 = \{A \sqsubseteq B \sqcap B\}$ are self-contained S^q_Σ-modules, but $\mathcal{T}$ itself is the only depleting S^q_Σ-module.

Example 9 Consider the *DL-Lite$_{bool}$* TBox $\mathcal{T}$ shown in Fig. 1 (it is part of the larger Core ontology to be discussed in the next section), and let $\Sigma = \{\mathsf{Publisher}\}$. First observe that *the* minimal S^c_Σ-module of $\mathcal{T}$ is empty, which is typical for singleton signatures and S^c, as no interesting concept inclusions over a singleton signature exist. In contrast, there are three different minimal S^q_Σ-modules of $\mathcal{T}$:

- $\mathcal{M}_D$ containing axioms (1)–(3),
- $\mathcal{M}_R$ containing axioms (8)–(14), and
- $\mathcal{M}_U$ with axioms (8)–(11) and (15).

First, they are indeed S^q_Σ-modules of $\mathcal{T}$, i.e., Σ-query inseparable from $\mathcal{T}$. This can be verified via the semantic criterion [Kontchakov *et al.*, 2008, Lemma A.4]. Second, they are minimal: consider the ABox $\mathcal{A} = \{\mathsf{Publisher}(a)\}$ and the query $q = \exists x \neg \mathsf{Publisher}(x)$. Clearly, we have $(\mathcal{T}, \mathcal{A}) \models q$, while $(\mathcal{T}', \mathcal{A}) \not\models q$, for any proper subset $\mathcal{T}'$ of $\mathcal{M}_D$, $\mathcal{M}_R$ or $\mathcal{M}_U$. In contrast to this finding, *the* minimal depleting S^q_Σ-module of $\mathcal{T}$ is $\mathcal{T}$ itself. Consider now $\Sigma' = \{\mathsf{Publisher}, \mathsf{pubHasDistrib}\}$. Then the only minimal $S^c_{\Sigma'}$-module of $\mathcal{T}$ consists of axioms (1)–(5) and there are two minimal $S^q_{\Sigma'}$-modules: $\mathcal{M}^+_R = \mathcal{M}_D \cup \mathcal{M}_R \cup \{(4), (5), (6)\}$ and $\mathcal{M}^+_U = \mathcal{M}_D \cup \mathcal{M}_U \cup \{(4), (5), (7)\}$.

4 Practical Minimal Module Extraction

We have conducted experiments with three types of minimal module extraction: for a *DL-Lite$_{bool}$* TBox $\mathcal{T}$ and a signature Σ, extract *some* minimal S^c_Σ-module (MCM) of $\mathcal{T}$, *some* minimal S^q_Σ-module (MQM), and *the* minimal depleting S^q_Σ-module (MDQM) of $\mathcal{T}$. In principle, these extraction problems can be solved using the algorithms presented in Theorems 2 and 5 together with the 'oracles' from [Kontchakov *et al.*, 2008] capable of deciding the (Π^p_2-complete) problems '$\mathcal{T}_1 \equiv^c_\Sigma \mathcal{T}_2$' and '$\mathcal{T}_1 \equiv^q_\Sigma \mathcal{T}_2$.' The oracles were realised by encoding these two problems as satisfiability problems for certain ($\forall\exists$) quantified Boolean formulas, and first experimental results indicated that standard off-the-shelf QBF solvers such as sKizzo [Benedetti, 2005], 2clsQ [Samulowitz and Bacchus, 2006] and QuBE [Giunchiglia *et al.*, 2006] can be successfully used to check satisfiability of the resulting QBFs.

Unfortunately, a naïve implementation of this approach turns out to be hopelessly inefficient. In a nutshell, the main reasons are as follows. First, to extract a minimal module from an ontology with 1K axioms, even for 'typical' real-world examples the algorithm would call the oracle about 500K times. Second, as was discovered in [Kontchakov *et al.*, 2008], *no* existing QBF solver could cope alone with all the inseparability tests and, even when the solver is successful, the runtime is quite unpredictable and can range from a few seconds to a few hours. The good news, however, is that all the three solvers we used did solve about 99% of tests. (Note that the QBF encodings of our tests below contain up to 232,600 clauses, 710,000 literals and 23,300 variables.) To deal with these problems we have implemented three ideas:

(1) To reduce the number of oracle calls, we optimised the algorithms by checking a group of axioms rather than a single axiom at a time (in practice, this reduced the number of calls to 1–3K for a 1K ontology).

(2) To select 'the best' QBF solver for a given instance, we used the self-adaptive multi-engine QBF solver aqme [Pulina and Tacchella, 2009], a tool capable of choosing a QBF engine with 'more chances' to solve a given input and learning its engine-selection strategies.

(3) To reduce the size of the original ontology, we 'preprocessed' it by means of a tractable syntactic locality-based algorithm from [Cuenca-Grau *et al.*, 2008] extracting the $\top\bot$-module ($\top\bot$M), which contains all the minimal modules we are interested in. In fact, we have the following inclusions

$$\text{MCM} \subseteq \text{MQM} \subseteq \text{MDQM} \subseteq \top\bot\text{M},$$

where the first one should be read as '*every* MQM contains *some* MCM,' the second as '*every* MQM is contained in *the* MDQM,' and the third as '*the* MDQM is contained in *the* $\top\bot$M.' Thus we can use these inclusions by computing modules from right to left.

Ontologies. Our test ontologies are *DL-Lite$_{bool}$* encodings of two real-world commercial software applications called 'Core' and 'Umbrella.' The Core ontology is based on a supply-chain management system used by the bookstore chain Ottakar's, now rebranded as Waterstone's. It contains 1283 axioms, 83 concept names and 77 role names, and features numerous functionality constraints, covering and disjointness constraints, and quite a few concepts of the form $\leq q\,R$ with $q > 2$. The Umbrella ontology is based on a specialised research data validation and processing system used by the Intensive Care National Audit and Research Centre (`http://www.icnarc.org`). It contains 1247 axioms, 79 concept names and 60 role names. Both ontologies are representations of the relevant data structures and were constructed by analysing the data model, database schema and application-level business logic. The Publisher ontology in Fig. 1 is part of Core; full Core and Umbrella are available at `http://ijcai09.tripod.com/`.

Using AQME. An important property of AQME is that it can update its learned policies when the usage scenario changes substantially, by using an adaptation schema called *retraining*. Prior to module extraction, AQME computed a selection of syntactic features (characterising this particular problem) from a pool of suitable QBF instances. In view of the findings of [Kontchakov *et al.*, 2008], we used only 3 engines out of the usual 8: 2clsQ, QuBE and sKizzo. A typical run of AQME is as follows. First, it leverages its inductive model (built using 1-nearest-neighbour) to predict the best engine for the given input QBF. If the engine solves the QBF, AQME terminates and returns the answer. Otherwise, AQME starts its self-adaptive mechanism. It calls a different engine to solve the input formula. If it is successful, the retraining procedure is called and the inductive model is updated. Which engine is called for retraining and how much CPU time is granted to each engine are critical points for AQME's performance. Our solver selection strategy, ALG, relies on the engine type, which can be search-based, like in QuBE and 2clsQ, or Skolemisation-based, like in sKizzo: the failed solver is replaced by a solver of a different type. Our CPU time strategy is as follows: a fixed amount of CPU time is granted to the predicted solver; if the solver fails, another engine is called, using the ALG strategy, with a granted amount of CPU time that increases in each iteration, until the solver solves the input formula. In fact, this approach combines the TPE and ITR techniques from [Pulina and Tacchella, 2009]. An important difference from the original version of AQME is the new data management at the retraining phase. As AQME had to deal with a huge number of instances, which could all be used for training and therefore cause a substantial slowdown, we bounded the number of entries in the training set: when the bound is reached, the older entry is replaced by the newer one.

Modules for $|\Sigma| = 1$. Our first experiment was to extract the modules of all three types, for all *singleton* signatures, from the full Core and Umbrella ontologies and from the corresponding pre-computed $\top\bot$Ms. For instance, the extracted MCM, MQM, and MDQM of Core for $\Sigma = \{\mathsf{Publisher}\}$ are given in Example 9 and contain 0, 3, and 20 axioms, respectively, whereas the corresponding $\top\bot$M has 228 axioms.

The following table summarises the results of the experiments (on average per module) in terms of module sizes and other relevant parameters, where the italicised numbers refer to extraction from the full ontologies. It also contains the average sizes of the segments extracted using the approaches described in [Seidenberg and Rector, 2006] (SR), [Noy and Musen, 2004] (Prompt), and [Cuenca-Grau *et al.*, 2006] (E-conn). Since these approaches do not support role names in the initial signature, we have only extracted modules for concept names in these cases. Furthermore, SR and Prompt are not logic-based and do not, in general, preserve entailments. (The Publisher-segments for SR, Prompt, and E-conn contain 19, 189, and 349 axioms, respectively.)

	Core (1283)		Umbrella (1247)	
$\top\bot$M	226		69	
MDQM	80		57	
extraction time [s]	126	*2233*	60	*2488*
AQME calls	385	*565*	254	*463*
sKizzo calls	14%	*76%*	4%	*74%*
2clsQ calls	2%	*17%*	1%	*14%*
QuBE calls	84%	*7%*	95%	*12%*
MQM	5		5	
MCM	2		2	
SR-segment	37		14	
Prompt segment	162		102	
E-conn module	243		47	

The distribution of calls to the QBF engines changes notably if MDQMs are extracted from the whole ontology rather than from the $\top\bot$Ms: in the former case, the majority of calls is issued to sKizzo, while QuBE handles most of the calls in the latter case. This complies with the observation that, in general, QuBE tends to solve easy instances more quickly, and sKizzo performs more successfully on harder instances.

Modules for $|\Sigma| = 10$. For each of our ontologies, we randomly generated 30 signatures of 10 concept names each and extracted all possible modules; their average sizes are shown in Fig. 2. MCMs and MQMs were extracted from MDQMs, which in turn were extracted from $\top\bot$Ms. Again, in most of the AQME calls (1302/1694 for MDQMs and 152/181 for MCMs, on average) QuBE was invoked. The average runtime for MDQMs (MCMs) was around 30 (1.5) minutes.

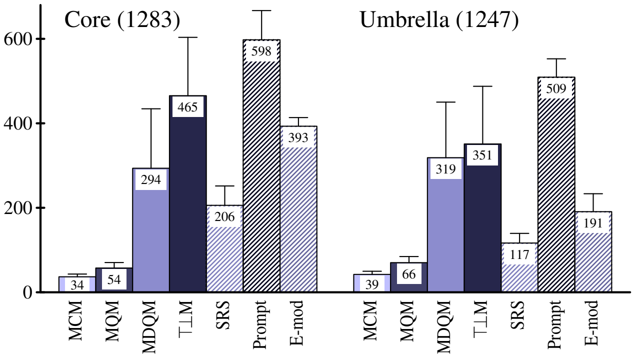

Figure 2: Module sizes for $|\Sigma| = 10$ and standard deviation.

It is to be noted that we have only been able to extract 17 MQMs for Umbrella and 8 MQMs for Core because the runtime for certain instances increases to a couple of days. One of the reasons is the growth of the QBF instances generated whenever the algorithm needs to test inseparability between module candidates and the original ontology. In the case of MDQMs, a candidate's complement needs to be compared with the empty TBox, which can be done rather efficiently. The case of MCMs and MQMs involves many comparisons of two very similar TBoxes, which, for MQMs, leads to the generation of QBF instances that are quadratic in the number of roles involved (as opposed to linear for MCMs).

Modules for $|\Sigma| = 5 + 5$. A similar experiment was conducted with 30 random signatures consisting of 5 concept names and 5 role names. The results are summarised in Fig.3. Due to performance issues, we have no MQMs available at all here. For reasons mentioned above, we did not extract module for SR, Prompt, and E-conn either.

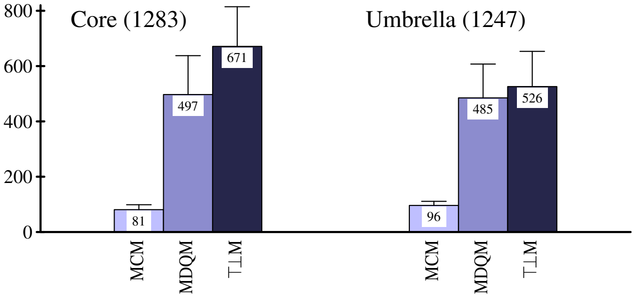

Figure 3: Module sizes for $|\Sigma| = 5 + 5$.

5 Conclusion

The main novel contributions of this paper are as follows:

(1) The definition of modules and reconstruction of their properties based on an abstract notion of inseparability formalising the intuitive notion that 'two ontologies give the same description of a vocabulary.' We believe that inseparability can be used to systematise the huge variety of distinct notions of modules in the literature.

(2) Experimental results demonstrating that, for *DL-Lite$_{bool}$* ontologies (and so many other logics in the *DL-Lite* family), minimal logic-based modules (in particular MDQMs and MCMs) can be extracted in practice using the multi-engine QBF solver AQME and locality-based module extraction. The experiments also show that QBF solvers can be used to solve complex real-world problems, though they are not as stable and scalable yet as the existing SAT solvers.

Many open problems remain. For example, for MQMs the performance of the tested extraction algorithms is still rather unsatisfactory and further optimisations are required. Second, it is straightforward to extend the framework and algorithms to the case where modules are extracted for a given signature *and* a seed set $\mathcal{M}$ of axioms to be included in the module. One can thus combine different methodologies and, say, compute the logically correct closure of non-logic based modules $\mathcal{M}$. It would be interesting to evaluate this approach in practice.

References

[Artale *et al.*, 2007a] A. Artale, D. Calvanese, R. Kontchakov, and M. Zakharyaschev. DL-Lite in the light of first-order logic. In *Proc. of AAAI*, 2007.

[Artale *et al.*, 2007b] A. Artale, D. Calvanese, R. Kontchakov, V. Ryzhikov, and M. Zakharyaschev. Reasoning over extended ER models. In *Proc. of ER*, vol. 4801 of *LNCS*, Springer, 2007.

[Benedetti, 2005] M. Benedetti. sKizzo: A suite to evaluate and certify QBFs. In *Proc. of CADE–20*, vol. 3632 of *LNCS*, Springer, 2005.

[Calvanese *et al.*, 2005] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. *DL-Lite*: Tractable description logics for ontologies. In *Proc. of AAAI*, 2005.

[Calvanese *et al.*, 2006] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Data complexity of query answering in description logics. In *Proc. of KR*, 2006.

[Cuenca-Grau *et al.*, 2007] B. Cuenca-Grau, V. Honovar, A. Schlicht, and F. Wolter, editors. *Second Workshop on Modular Ontologies*, vol. 315. CEUR 2007.

[Cuenca-Grau *et al.*, 2006] B. Cuenca-Grau, B. Parsia, E. Sirin, and A. Kalyanpur. Modularity and web ontologies. In *Proc. of KR*, 2006.

[Cuenca-Grau *et al.*, 2008] B. Cuenca-Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Modular reuse of ontologies: Theory and practice. *JAIR*, 31:2008.

[Ghilardi *et al.*, 2006] S. Ghilardi, C. Lutz, and F. Wolter. Did I damage my ontology? A case for conservative extensions in description logic. In *Proc. of KR*, 2006.

[Giunchiglia *et al.*, 2006] E. Giunchiglia, M. Narizzano, and A. Tacchella. Clause-Term Resolution and Learning in Quantified Boolean Logic Satisfiability. *JAIR*, 2006.

[Haase *et al.*, 2006] P. Haase, V. Honavar, O. Kutz, Y. Sure, and A. Tamilin, editors. *First Workshop on Modular Ontologies*, vol. 232. CEUR 2006.

[Konev *et al.*, 2008] B. Konev, C. Lutz, D. Walther, and F. Wolter. Semantic modularity and module extraction in description logic. In *Proc. of ECAI*, 2008.

[Kontchakov *et al.*, 2008] R. Kontchakov, F. Wolter, and M. Zakharyaschev. Can you tell the difference between DL-Lite ontologies? In *Proc. of KR*, 2008.

[Noy and Musen, 2004] N. Noy and M. Musen. Specifying ontology views by traversal. In *Proc. of ISWC*, 2004.

[Parent *et al.*, 2009] C. Parent, S. Spaccapietra, and H. Stuckenschmidt, editors. *Ontology Modularization*. Springer Lecture Notes, 2009. (To appear)

[Pulina and Tacchella, 2009] L. Pulina and A. Tacchella. A self-adaptive multi-engine solver for quantified Boolean formulas. *Constraints*, 14:2009.

[Samulowitz and Bacchus, 2006] H. Samulowitz and F. Bacchus. Binary clause reasoning in QBF. In *Proc. of SAT*, vol. 4121 of *LNCS*, 2006.

[Seidenberg and Rector, 2006] J. Seidenberg and A. Rector. Web ontology segmentation analysis. In *WWW*, 2006.