Computing $\mathcal{ALCH}$-Subsumption Modules Using Uniform Interpolation

Patrick Koopmann[1] and Jieying Chen[2]

[1] Institute of Theoretical Comp. Science, Technische Universität Dresden, Germany
patrick.koopmann@tu-dresden.de
[2] LRI, Univ. Paris-Sud, CNRS, Université Paris-Saclay, Orsay, France
jieying.chen@lri.fr

Abstract. We investigate how minimal subsumption modules can be extracted using methods for uniform interpolation and forgetting. Given an ontology and a signature of concept and role names, a subsumption module is a subset of the ontology that preserves all logical entailments that can be expressed in the description logic of the ontology using only terms in the specified signature. As such, they are useful for ontology reuse and ontology analysis. While there exists a range of methods for computing or approximating minimal modules for a range of module types, we are not aware of a practical, implemented method for computing minimal subsumption modules in description logics beyond $\mathcal{ELH}$. In this paper, we present a method that uses uniform interpolation/forgetting to compute subsumption modules in $\mathcal{ALCH}$, and which under certain conditions guarantees minimality of the extracted modules. As a side product, our method computes a so-called LK subsumption module, which over-approximates the union of all minimal subsumption modules, and as such may already have applications of its own. We further present an initial evaluation of this method on a varied corpus of ontologies.

1 Introduction

Description Logics [1] (DLs) are a well-investigated family of logics that are commonly used to describe terminological knowledge in form of ontologies. Applications in areas such as medicine, biology and the semantic web have lead to the development of very large ontologies that, with growing size, become harder to understand and maintain. Due to the complexity of existing ontologies, there are areas where it is useful to extract a subset of the ontology, a so-called *module*, based on a set of terms of interest. For example, when developing a new ontology for a specialised application, one may want to reuse knowledge from an existing ontology. If this ontology covers a large domain of concepts, not all information in it will be relevant for the application at hand, so that it makes sense to first extract a module of the ontology that is sufficient for the application. Secondly, for maintaining an existing ontology it may be important for an ontology engineer to understand which axioms in the ontology are responsible for which of its logical entailments. Modules give the engineer an overview on the axioms of the ontology that contribute to any entailment over a selected set of terms. This allows him to browse the ontology in a more directed manner guided by the terms he is interested in. In the mentioned applications, it is usually desirable to extract a module that is optimal in some sense, for example minimal w.r.t. set inclusion.

There are a range of different notions and properties for modules that have been defined in the literature, and correspondingly a range of methods for module extraction have been developed [2]. For some of those notions, such as semantic modules, deciding whether a subset of the ontology is a module that is minimal w.r.t. set inclusion is undecidable already for ontologies formulated in the lightweight DL $\mathcal{EL}$ [13]. In this paper, however, we consider a notion for which computing a minimal module is decidable. More precisely, we are interested in computing *minimal subsumption modules*, which are modules that preserve all logical entailments in the form of concept inclusions over the specified signature of terms.While a method for computing minimal subsumption modules in acyclic $\mathcal{EL}$ ontologies has been presented in [4], in this paper we focus on the more expressive DL $\mathcal{ALCH}$. Already for $\mathcal{ALC}$ ontologies, deciding whether a subset of the ontology is a subsumption module is known to be 2ExpTime-complete [6], but we are not aware of a practical implementation for extracting minimal subsumption modules in DLs that are more expressive than $\mathcal{ELH}$ [3].

The core idea of our method is to use uniform interpolation [21] to compute a finite representation of the entailments the module has to preserve, together with techniques from axiom-pinpointing [27]. The method can compute small subsumption modules of $\mathcal{ALCH}$-ontologies for signatures that contain all role symbols, which under certain conditions guarantees minimality of the computed modules. As a side-product, the method computes a *lean kernel (LK) subsumption module*, an over-approximation of all minimal subsumption modules, which may have applications on its own.

Our method only supports signatures that contain all role names. We believe that this restriction is well motivated for practical applications. For ontology exploration, the user may also require to know the role names related to the concept names he is interested in. Moreover, modules for signatures that include all role names provide for a weak form of *robustness under replacement* [12], which is especially useful for ontology reuse.

The paper is structured as follows. We first recall related work on module extraction and uniform interpolation in Section 2, and give the preliminaries on $\mathcal{ALCH}$, subsumption modules and uniform interpolation in Section 3. We then describe our core algorithm for computing subsumption modules based on axiom pin-pointing in Section 4. A central idea for reducing the number of entailment checks is to use a technique to quickly compute uniform interpolants for different subsets of the input ontology, for which we compute an *annotated uniform interpolant* defined in Section 5. As a by-product, the annotated uniform interpolant encodes an upper approximation of the minimal subsumption module, which indeed over-approximates all minimal subsumption modules. We call this module lean kernel subsumption module, as they are similar to lean kernels in axiom pinpointing, which we discuss in more detail in Section 6. Finally, we give results from an initial evaluation in Section 7 and conclude with a discussion in Section 8.

2 Related Work

There is a range of types and properties of modules that have been investigated in the literature, surveys of which can be found in [12] and [2]. Usually, modules are computed on the basis of an ontology and a signature Σ, i.e. a set of concept and role names, and preserve certain properties of the ontology with respect to that signature Σ. Examples include *semantic modules*, which preserve all models of the ontology when restricted to Σ [13] and *subsumption modules*, which preserve all logical entailments in the form of concept inclusions over Σ that can be expressed in the description logic under consideration [4,3]. Apart from minimality under set inclusion, additional properties have been considered such as *self-containedness* (the module is also a module with respect to its own signature) and depletedness (the remaining ontology only entails tautologies in the specified signature, i.e. all relevant information is in the module). Deciding whether a subset of the ontology is a semantic module for a signature is undecidable already for $\mathcal{EL}$-ontologies [13], and consequently minimal (depleting, self-contained) semantic modules can only be approximated in practice. For $\mathcal{EL}$ and $\mathcal{ALCI}$, an exception are modules of acyclic ontologies and for signatures that contain only concept names, for which methods to extract depleting modules have been implemented in the tool MEX [13]. A well-known approximation of semantic modules are locality-based modules, of which syntactical variants, such as the prominent $\top\bot*$-modules, can be computed very cheaply [8,14]. However, locality-based modules may still contain a large portion of the original ontology [26]. A more refined technique for extracting semantic modules is presented in [5], which computes lower and upper approximations of minimal depleting modules in $\mathcal{ALCQI}$ using QBF-reasoning. Depending on the application, modules that only preserve entailments in a certain query-language may be sufficient. A method that approximates minimal modules taylored towards specific query languages uses datalog reasoning and has been presented in [25].

For computing minimal subsumption modules of $\mathcal{ELH}$-terminologies, a method has been presented in [3]. An alternative approach is presented in [4], which uses a black-box search algorithm that detects axioms that can be safely removed without causing a *logical difference*, i.e. a difference in the set of entailed concept inclusions over the selected signature. For computing logical differences, the authors use the tool CEX [11], which currently supports $\mathcal{EL}$ terminologies. However, in principle, the same algorithm could be used for ontologies formulated in any logic for which a tool for computing logical differences exists. We tried this for $\mathcal{ALCH}$ ontologies, deploying the tool Lethe [17] that allows for computing logical differences in $\mathcal{ALCH}$. However, we found that this approach is too computationally expensive in practice.

A notion strongly related to that of subsumption modules is that of uniform interpolants, which are also computed as part of our method. Both subsumption modules and uniform interpolants preserve all logical entailments in the specified signature that can be expressed in the respective description logic. However, while subsumption modules are subsets of the input ontology, uniform interpolants are themselves completely formulated in the specified signature, and may therefore contain axioms that do not occur in the original ontology. In fact, in the worst case, already for the description logics $\mathcal{EL}$ and $\mathcal{ALC}$, the uniform interpolant may have a size that is triple exponential in the size of the input ontology [22,21]. Despite this discouraging theoretical result, various practical methods for computing uniform interpolants in expressive description logics have been developed [20,16,18,30]. In fact it turns out that in practice, uniform interpolants are often of moderate size.

3 Preliminaries

We recall the description logic $\mathcal{ALCH}$ [1], as well as the notions of subsumption modules and uniform interpolants.

Let N_c and N_r be two disjoint, countably infinite, sets of respectively *concept names* and *role names*. A signature $\Sigma \subseteq N_c \cup N_r$ is a finite set of concept names and role names.

The set of *concepts* C, D, *TBox axioms* α and *RBox axioms* β the set of $\mathcal{ALCH}$*-inclusions* α are built according to the following grammar rules:

$$C ::= \top \mid \bot \mid A \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \exists r.C \mid \forall r.C$$
$$\alpha ::= C \sqsubseteq C \mid C \equiv C$$
$$\beta ::= r \sqsubseteq s \mid r \equiv s$$

where $A \in N_c$ and $r \in N_r$. A TBox is a finite set of TBox axioms, an RBox a finite set of RBox axioms, and an ontology is the union of a TBox and RBox.

The semantics of $\mathcal{ALCH}$ is defined using interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where the domain $\Delta^{\mathcal{I}}$ is a non-empty set, and $\cdot^{\mathcal{I}}$ is a function assigning each concept name A to a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$ and every role name r to a binary relation $r^{\mathcal{I}}$ over $\Delta^{\mathcal{I}}$. Then $\cdot^{\mathcal{I}}$ is inductively extended to complex concepts by: $(\top)^{\mathcal{I}} := \Delta^{\mathcal{I}}$, $(\bot)^{\mathcal{I}} := \emptyset$, $(\neg C)^{\mathcal{I}} := \Delta^{\mathcal{I}} \backslash C^{\mathcal{I}}$, $(C \sqcap D)^{\mathcal{I}} := C^{\mathcal{I}} \cap D^{\mathcal{I}}$, $(C \sqcup D)^{\mathcal{I}} := C^{\mathcal{I}} \cup D^{\mathcal{I}}$, $(\exists r.C)^{\mathcal{I}} := \{x \in \Delta^{\mathcal{I}} \mid \exists y \in C^{\mathcal{I}} : (x, y) \in r^{\mathcal{I}}\}$, and $(\forall r.C)^{\mathcal{I}} := \{x \in \Delta^{\mathcal{I}} \mid \forall (x, y) \in r^{\mathcal{I}} : y \in C^{\mathcal{I}}\}$.

An interpretation $\mathcal{I}$ satisfies a TBox axiom $C \sqsubseteq D$ ($C \equiv D$) iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ ($C^{\mathcal{I}} = D^{\mathcal{I}}$). It satisfies an RBox axiom $r \sqsubseteq s$ ($r \sqsubseteq s$) iff $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$ ($r^{\mathcal{I}} = s^{\mathcal{I}}$). We write $\mathcal{I} \models \alpha$ if $\mathcal{I}$ satisfies the axiom α. An interpretation $\mathcal{I}$ is a *model* of an ontology $\mathcal{O}$ if $\mathcal{I}$ satisfies all axioms in $\mathcal{O}$. An axiom α *is entailed by* $\mathcal{O}$, written $\mathcal{O} \models \alpha$, if for all models $\mathcal{I}$ of $\mathcal{O}$, we have that $\mathcal{I} \models \alpha$.

A *signature* is a (countable but possibly infinite) set $\Sigma \subseteq N_r \cup N_c$ of concept and role names. Given a concept/axiom/ontology α, we denote by $\mathsf{sig}(\alpha)$ the set of concept and role names occuring in α.

Definition 1 (Σ-Inseparability, Subsumption Module, Uniform Interpolant.). *Let $\mathcal{O}_1$ and $\mathcal{O}_2$ be two $\mathcal{ALCH}$-ontologies, and let Σ be a signature. Then $\mathcal{O}_1$ and $\mathcal{O}_2$ are* Σ-inseparable*, denoted as $\mathcal{O}_1 \equiv_\Sigma \mathcal{O}_2$, iff for every axiom α s.t. $\mathsf{sig}(\alpha) \subseteq \Sigma$, we have $\mathcal{O}_1 \models \alpha$ iff $\mathcal{O}_2 \models \alpha$.*

A Σ-subsumption module of $\mathcal{O}$ *is an ontology $\mathcal{M}$ s.t. $\mathcal{O} \equiv_\Sigma \mathcal{M}$ and $\mathcal{M} \subseteq \mathcal{O}$. $\mathcal{M}$ is* minimal *iff there exists no Σ-subsumption module $\mathcal{M}'$ of $\mathcal{O}$ s.t. $\mathcal{M}' \subsetneq \mathcal{M}$.*

A uniform interpolant of $\mathcal{O}$ for Σ *is an ontology $\mathcal{O}_\Sigma$ s.t. $\mathcal{O} \equiv_\Sigma \mathcal{O}_\Sigma$ and $\mathsf{sig}(\mathcal{O}_\Sigma) \subseteq \Sigma$.*

4 Minimal Subsumption Modules as Justifications

A related problem to minimal subsumption module extraction is that of computing justifications [27]. Given an ontology $\mathcal{O}$ and an axiom α that is entailed by $\mathcal{O}$, a *justification for α in $\mathcal{O}$* is a subset $\mathcal{J}$ of $\mathcal{O}$ s.t. $\mathcal{J}$ is minimal w.r.t. $\subsetneq$ and $\mathcal{J} \models \alpha$. We can generalise this notion to *justifications of ontologies* $\mathcal{O}'$ by asking for minimal subsets $\mathcal{J} \subseteq \mathcal{O}$ s.t. $\mathcal{J} \models \mathcal{O}'$. One easily sees that every minimal subsumption module is a justification of a uniform interpolant: for a given ontology $\mathcal{O}$ and signature Σ, a uniform interpolant $\mathcal{O}^\Sigma$ captures all entailments of $\mathcal{O}$ that are in Σ, and therefore, any subset of $\mathcal{O}$ that entails $\mathcal{O}^\Sigma$ entails all axioms that are in Σ. Therefore, one possible approach for computing minimal subsumption modules is to first compute a uniform interpolant, and then compute a justification for it using any DL reasoner that supports this. As there are implemented systems for both for uniform interpolation (e.g. [20,17,30]), and for computing justifications (reasoners such as HermiT [7], JFact [29] and Pellet [28] support this directly via the OWL API [9]), it seems that such a method could be implemented without much effort.

However, there are two short-comings of this approach. First, it is well-known that uniform interpolants do not always exists for any pair of ontology and signature. For this reason, existing methods for uniform interpolation either only compute approximations of the uniform interpolant, or they compute a uniform interpolant in an extended language that uses greatest fixpoint operators. Unfortunately, we are not aware of any reasoner that supports fixpoint operators, so that the method can only be applied for ontology-signature pairs for which there exist a uniform interpolant without fixpoint operators. Secondly, in first experiments of this idea we quickly found out that reasoners such as HermiT, Pellet and JFact struggle with the computation of justifications for large entailments such as uniform interpolants. While in this paper, we offer no general solution for the case in which there is no uniform interpolant without fixpoints, to overcome the more practical problem of computing justifications for uniform interpolants without fixpoint operators, we developed a more refined approach based on ideas for computing justifications.

Given an ontology $\mathcal{O}_1$ and a set of entailed axioms $\mathcal{O}_2$, such as a uniform interpolant, we can compute a justification for $\mathcal{O}_2$ in $\mathcal{O}_1$ using the following algorithm **A1**.

1. Input: ontology $\mathcal{O}_1$, entailed set of axioms $\mathcal{O}_2$

2. For each $\alpha \in \mathcal{O}_1$:
   (a) Set $\mathcal{O}_1' = \mathcal{O}_1 \setminus \{\alpha\}$
   (b) If $\mathcal{O}_1' \models \mathcal{O}_2$, set $\mathcal{O}_1 = \mathcal{O}_1'$
3. Return $\mathcal{O}_1$

If in Step 2b), we test entailment of $\mathcal{O}_2$ by checking one axiom after the other, this algorithm has to perform a quadratic number of entailment tests, namely one for each pair (α, β) of axioms $\alpha \in \mathcal{O}_1$ and $\beta \in \mathcal{O}_2$. However, if $\mathcal{O}_1$ and $\mathcal{O}_2$ overlap syntactically, this number of tests can be reduced, as we do not need to call a reasoner for axioms that are already in $\mathcal{O}_1$. Unfortunately, if $\mathcal{O}_2$ is a uniform interpolant of $\mathcal{O}_1$ for Σ, it only contains axioms in Σ, and is therefore unlikely to syntactically overlap with $\mathcal{O}_1$, especially if Σ is significantly smaller than the signature of $\mathcal{O}_1$. To overcome this problem, we propose the following algorithm **A2**, which checks for entailment of the uniform interpolant by another uniform interpolant.

1. Input: Ontology $\mathcal{O}$, signature Σ
2. Initialise $\mathcal{O}_m$ to the $\top\bot*$-module of $\mathcal{O}$ for Σ
3. Compute the uniform interpolant $\mathcal{O}^\Sigma$ of $\mathcal{O}_m$ for Σ
4. For each $\beta \in \mathcal{O}_m$:
   (a) Compute the uniform interpolant $\mathcal{O}_2^\Sigma$ of $\mathcal{O}_m \setminus \{\beta\}$ for Σ
   (b) Set $\mathcal{O}_d = \mathcal{O}^\Sigma \setminus \mathcal{O}_2^\Sigma$
   (c) If $\mathcal{O}_2^\Sigma \models \mathcal{O}_d$, set $\mathcal{O}_m = \mathcal{O}_m \setminus \{\beta\}$
5. Return $\mathcal{O}_m$

Of course, the improvement of this method relies on the shape of the uniform interpolants we compute: in general, $\mathcal{O}^\Sigma$ and $\mathcal{O}_2^\Sigma$ may not overlap at all, so that $\mathcal{O}_d$ may have the same size as $\mathcal{O}^\Sigma$. However, as our experiments confirmed, if the uniform interpolants are computed wisely, in the algorithm above $\mathcal{O}_d$ is usually significantly smaller than $\mathcal{O}_2^\Sigma$, and in fact often contains only a single axiom. Therefore, the algorithm is expected to require a much smaller number of entailment checks than **A1**. However, we now need to compute a uniform interpolant in every iteration, which usually is a much more expensive operation than testing for entailment of axioms. Luckily, as it turns out, for signatures Σ s.t. $N_r \subseteq \Sigma$, we can compute the required uniform interpolants very efficiently if we compute an *annotated uniform interpolant* first, from which all required uniform interpolants can then be obtained by simple replacement operations.

5 Annotated Uniform Interpolants

In the following, for simplicity, let $\mathcal{O}$ be the input ontology of our method. Let $N_a \subseteq N_c$ be a special set of concept names called *annotation concepts*, which we assume to be disjoint from the signature of $\mathcal{O}$, and let $\mathsf{A} : \mathcal{O} \to N_c$ be a bijective function that maps each axiom in $\mathcal{O}$ to an annotation concept. To improve readability, we write $\mathsf{A}(\alpha)$ as A_α.

Note that in $\mathcal{ALCH}$, every TBox axiom is equivalent to a set of TBox axioms of the form $C \sqsubseteq D$. Given a TBox axiom α, we denote by $\mathsf{gci}(\alpha)$ the set of GCIs that is equivalent to α.

Definition 2. *Given an axiom α, the* annotation α_a of α *is defined as*

$$\{C \sqsubseteq D \sqcup \mathsf{A}_\alpha \mid C \sqsubseteq D \in \mathsf{gci}(\alpha)\}.$$

Given an ontology $\mathcal{O}$, the annotation $\mathcal{O}_a$ of $\mathcal{O}$ is the union of all annotations of axioms in $\mathcal{O}$. Given a signature Σ, a annotated uniform interpolant of $\mathcal{O}$ for Σ *is a uniform interpolant $\mathcal{O}_a^\Sigma$ of the annotation of $\mathcal{O}$ for the signature $\Sigma \cup \{\mathsf{A}_\alpha \mid \alpha \in \mathcal{O}\}$.*

Note that the annotation $\mathcal{O}_a$ of an ontology $\mathcal{O}$ is usually not a conservative extension. Specifically, $\mathcal{O}_a \not\models C \sqsubseteq D$ may not hold even for $C \sqsubseteq D \in \mathcal{O}$, due to the added disjuncts. Instead, all non-tautological entailments now involve annotation concepts that refer to the axioms that have been used to infer the axiom.

The idea of the annotation concepts is to track which axioms contributed to computing a uniform interpolant. This way, we can easily obtain a uniform interpolant of any subset of the original ontology. Specifically, given an annotated uniform interpolant of $\mathcal{O}$ for Σ, where $N_r \subseteq \Sigma$, we can obtain uniform interpolants for Σ of any subset $\mathcal{O}'$ of $\mathcal{O}$ as follows: we replace every annotation concept A_α s.t. $\alpha \in \mathcal{O}'$ by $\bot$, and every remaining annotation concept by $\top$.

Example 1. Consider the following ontology $\mathcal{O}$.

$$\exists r.\top \sqsubseteq A \sqcup B \qquad A \equiv \exists r.B$$

The following ontology is a uniform interpolant of $\mathcal{O}$ for $\Sigma = \{A, r\}$.

$$\begin{aligned} A &\sqsubseteq \exists r.\top \\ \exists r.(\exists r.\top \sqcap \neg A) &\sqsubseteq A \end{aligned}$$

To track which axioms contributed to the uniform interpolant, we compute the annotated uniform interpolant. For this, we first compute the annotation of $\mathcal{O}$, which is the following.

$$\begin{aligned} \exists r.\top &\sqsubseteq A \sqcup B \sqcup \mathsf{A}_{\exists r.\top \sqsubseteq A \sqcup B} \\ A &\sqsubseteq \exists r.B \sqcup \mathsf{A}_{A \equiv \exists r.B} \\ \exists r.B &\sqsubseteq A \sqcup \mathsf{A}_{A \equiv \exists r.B} \end{aligned}$$

By interpolating the annotation of $\mathcal{O}$, we obtain the following annotated uniform interpolant of $\mathcal{O}$ for Σ.

$$\begin{aligned} A &\sqsubseteq \exists r.\top \sqcup \mathsf{A}_{A \equiv \exists r.B} \\ \exists r.(\exists r.\top \sqcap \neg A \sqcap \neg \mathsf{A}_{\exists r.\top \sqsubseteq A \sqcup B}) &\sqsubseteq A \sqcup \mathsf{A}_{A \equiv \exists r.B} \end{aligned}$$

The annotation concepts mark which parts of the uniform interpolant where influenced by which axiom. If we replace every annotation concept by $\bot$, we obtain a uniform interpolant of $\mathcal{O}$ again. If instead, we replace $\mathsf{A}_{A\equiv\exists r.B}$ by $\bot$ and $\mathsf{A}_{\exists r.\top\sqsubseteq A\sqcup B}$ by $\top$, we obtain the following uniform interpolant of $A \equiv \exists r.B$:

$$A \sqsubseteq \exists r.\top \sqcup \bot$$
$$\exists r.(\exists r.\top \sqcap \neg A \sqcap \neg\top) \sqsubseteq A \sqcup \bot,$$

which can be simplified to $\{A \sqsubseteq \exists r.\top\}$, as the second axiom is tautological. In a same way, we obtain that a uniform interpolant of $\exists r.\top \sqsubseteq A \sqcup B$ is $\{\bot \sqsubseteq \top\}$.

This technique however only works for signatures that contain all role symbols of the original ontology. The following lemma, which can be shown by inspection of the uniform interpolation method presented in [15], provides the central property of uniform interpolants which make our technique possible.

Lemma 1. *Let $\mathcal{O}$ be an ontology and Σ a signature s.t. $N_r \subseteq \Sigma$. Let $\mathcal{O}_1 \subseteq \mathcal{O}$ be such that $\mathsf{sig}(\mathcal{O}_1) = \Sigma$ and $\mathcal{O}_1$ contains no RBox axioms, and let $\mathcal{O}_2 = \mathcal{O} \setminus \mathcal{O}_1$. Further, let $\mathcal{O}_2^\Sigma$ be a uniform interpolant of $\mathcal{O}_2$ for Σ. Then, $\mathcal{O}_1 \cup \mathcal{O}_2^\Sigma$ is a uniform interpolant of $\mathcal{O}$ for Σ.*

As a corollary of this lemma, we obtain the robustness property of subsumption modules for signatures Σ s.t. $N_r \subseteq \Sigma$ which was claimed in the introduction, and which can equivalently be proved based on a corresponding result for $\mathcal{ALC}$ from [12].

Corollary 1 (Weak robustness under replacement). *Let $\mathcal{O}$ be an ontology, Σ a signature s.t. $N_r \subseteq \Sigma$, and $\mathcal{M}$ be a Σ-subsumption module for $\mathcal{O}$. Let $\mathcal{O}'$ be an ontology s.t. $\mathsf{sig}(\mathcal{O}')\cap\mathsf{sig}(\mathcal{O}) \subseteq \Sigma$ and $\mathcal{O}'$ contains no RBox axioms containing role names from $\mathcal{O}$. Then, for any axiom α s.t. $\mathsf{sig}(\alpha) \subseteq \mathsf{sig}(\mathcal{O}') \cup \Sigma$, we have $\mathcal{O} \cup \mathcal{O}' \models \alpha$ iff $\mathcal{M} \cup \mathcal{O}' \models \alpha$.*

We can now show that uniform interpolants of subsets of the original ontology can indeed be computed using the replacement operations described earlier. (Recall that annotated uniform interpolants are defined as uniform interpolants of the annotated ontology, for the given signature extended by annotation concepts.)

Theorem 1. *Let $\mathcal{O}$ be an ontology, Σ a signature s.t. $N_r \subseteq \Sigma$, and $\mathcal{O}_a^\Sigma$ an annotated uniform interpolant of $\mathcal{O}$ for Σ. Let $\mathcal{O}_1 \subseteq \mathcal{O}$. Then, the ontology*

$$\mathcal{O}_1^\Sigma = \mathcal{O}_a^\Sigma[\mathsf{A}_\alpha \mapsto \bot \mid \alpha \in \mathcal{O}_1][\mathsf{A}_\alpha \mapsto \top \mid \alpha \in \mathcal{O} \setminus \mathcal{O}_1]$$

is a uniform interpolant of $\mathcal{O}_1$ for Σ.

Proof. Let $\mathcal{O}$, Σ, $\mathcal{O}_a^\Sigma$ and $\mathcal{O}_1$ be as in the lemma. Let $\mathcal{O}_a$ be the annotation of $\mathcal{O}$, and extend $\mathcal{O}_a$ to following ontology $\mathcal{O}_2$.

$$\mathcal{O}_a \cup \{\mathsf{A}_\alpha \equiv \bot \mid \alpha \in \mathcal{O}_1\} \cup \{\mathsf{A}_\alpha \equiv \top \mid \mathsf{A}_\alpha \in \mathcal{O} \setminus \mathcal{O}_1\}$$

By looking at the way axioms are annotated, one easily establishes that the uniform interpolant of $\mathcal{O}_2$ for $\mathsf{sig}(\mathcal{O})$ is equivalent to $\mathcal{O}_1$. Also, one easily sees that this uniform interpolant is simply obtained by replacing every annotation concept A_α s.t. $\alpha \in \mathcal{O}$ by $\bot$, and every annotation concept A_α s.t. $\mathsf{A}_\alpha \in \mathcal{O} \setminus \mathcal{O}_1$ by $\top$. Since uniform interpolation is commutative, we can obtain a uniform interpolant of $\mathcal{O}_1$ for Σ, starting from $\mathcal{O}_2$, in two ways. Either we first compute $\mathcal{O}_1$ as uniform interpolant of $\mathcal{O}_2$, and compute then the uniform interpolant of $\mathcal{O}_1$ for Σ. Or we first compute the uniform interpolant of $\mathcal{O}_2$ for $\Sigma \cup N_a$, of which we then compute the uniform interpolant for Σ. As observed earlier, this last step is simply performed by replacing annotation concepts by $\bot$ respectively $\top$. By Lemma 1, the uniform interpolant of $\mathcal{O}_2$ is equivalent to the uniform interpolant of $\mathcal{O}_a$—the annotated uniform interpolant—together with the additional axioms in $\mathcal{O}_2$, which means, these axioms are only involved in computing the second uniform interpolant. Therefore, we obtain the same ontology if we first compute the annotated uniform interpolant of $\mathcal{O}$, and then perform the substitution on the annotated uniform interpolant. □

6 LK Subsumption Modules

Theorem 1 allows us to apply Algorithm **A2** without having to use an expensive uniform interpolation method in each step. Another consequence of Theorem 1 is that, if we take the axioms associated with the set of annotation concepts occurring in the annotated uniform interpolant, we obtain a set of axioms that contains all minimal subsumption modules. This module is not necessarily equal to the union of all minimal subsumption modules, since the annotated uniform interpolant may contain tautological axioms, so that it is an over-approximation. We call this module *lean kernel subsumption module* (LK subsumption module for short), since it contains all axioms that were involved in computing the uniform interpolant, similar to lean kernels in SAT-solving [19]. LK subsumption modules can be computed more cheaply than minimal subsumption modules, as they do not require any further subsumption tests after the annotated uniform interpolant is computed. We believe that they also have a special use in ontology engineering: given a set of concept names, they allow the ontology engineer to quickly examine all the axioms that are involved in inferring any entailments involving no other concept names.

Definition 3. *Let $\mathcal{O}$ be an ontology and Σ a signature. An ontology $\mathcal{M}^{\Sigma}_{lk}$ is a Σ LK subsumption module iff there exists an annotated uniform interpolant $\mathcal{O}^{\Sigma}_{a}$ of $\mathcal{O}$ for Σ that is obtained by collecting all axioms that belong to some annotation concepts occurring in $\mathcal{O}^{\Sigma}_{a}$:*

$$\mathcal{M}^{\Sigma}_{lk} = \{\alpha \mid \mathsf{A}_\alpha \in \mathsf{sig}(\mathcal{O}^{\Sigma}_{a})\}.$$

Corollary 2. *Given an ontology $\mathcal{O}$, a signature Σ s.t. $N_r \subseteq \Sigma$ and a Σ LK subsumption module $\mathcal{M}^{\Sigma}_{lk}$ for $\mathcal{O}$, we have $\mathcal{M} \subseteq \mathcal{M}^{\Sigma}_{lk}$ for every minimal Σ-subsumption module $\mathcal{M}$ of $\mathcal{O}$.*

Since LK subsumption modules can be obtained from the annotated uniform interpolant without additional subsumption tests, they can always be computed even in the case where the annotated uniform interpolant contains fixpoint operators. However, different to minimal subsumption modules they do not offer any minimality guarantees: which axioms are contained in the LK subsumption module depends on the uniform interpolation procedure used, and as uniform interpolants may contain redundant and tautological information, it is in general possible that an LK subsumption module contains axioms that are not included in any minimal subsumption module.

The requirement that the signature contains all role names is indeed crucial for the correctness of our method, as is exemplified by the following example.

Example 2. Take following ontology $\mathcal{O}$:

$$A \sqsubseteq \exists r.B \qquad A \sqsubseteq C \qquad B \sqsubseteq \bot,$$

and consider the signature $\Sigma_1 = \{A, r\}$. The annotation $\mathcal{O}_a$ of $\mathcal{O}$ looks as follows.

$$\begin{aligned} &A \sqsubseteq \exists r.B \sqcup \mathsf{A}_{A\sqsubseteq\exists r.B} \\ &A \sqsubseteq C \sqcup \mathsf{A}_{A\sqsubseteq C} \\ &B \sqsubseteq \bot \sqcup \mathsf{A}_{B\sqsubseteq\bot} \end{aligned}$$

A uniform interpolant $\mathcal{O}_a^{\Sigma}$ for $\mathcal{O}_a$ for $\{A, r, \mathsf{A}_{A\sqsubseteq\exists r.B}, \mathsf{A}_{A\sqsubseteq C}, \mathsf{A}_{B\sqsubseteq\bot}\}$ is

$$A \sqsubseteq \exists r.\mathsf{A}_{B\sqsubseteq\bot} \sqcup \mathsf{A}_{A\sqsubseteq\exists r.B}.$$

Consequently, the Σ_1 LK subsumption module contains the first and the last axiom of $\mathcal{O}$. One easily verifies that these two axioms also form the only minimal Σ_1-subsumption module of $\mathcal{O}$.

Now consider the signature $\Sigma_2 = \{A, C\}$. A close look at the original ontology shows that in fact, the concept A is unsatisfiable, which can be inferred just from the first and the last axiom. This makes the second axiom redundant, and therefore the minimal subsumption module for Σ_2 is the same as for Σ_1. However, the uniform interpolant $\mathcal{O}_a^{\Sigma_2}$ for $\mathcal{O}_a$ for $\{A, C, \mathsf{A}_{A\sqsubseteq\exists r.B}, \mathsf{A}_{A\sqsubseteq C}, \mathsf{A}_{B\sqsubseteq\bot}\}$ just contains the following axiom.

$$A \sqsubseteq C \sqcup \mathsf{A}_{A\sqsubseteq C}$$

That is, the LK subsumption module contains just one axiom, which is exactly the only axiom that does not occur in the minimal subsumption module. The reason that the annotated uniform interpolant does not contain any more axioms is that the axiom $A \sqsubseteq \exists r.\mathsf{A}_{B\sqsubseteq\bot} \sqcup \mathsf{A}_{A\sqsubseteq\exists r.B}$ has no non-tautological $\mathcal{ALCH}$-entailment that does not also make use of the role name r.

Finally, for the complete signature $\Sigma_3 = \{A, r, C\}$, the Σ_3 LK subsumption module contains all axioms. However, as we already observed, the second axiom is redundant, and thus, this subsumption module is not minimal.

7 Evaluation

To evaluate how our method performs in practice, we implemented a Java prototype of our method and did an initial evaluation on a set of 96 ontologies that were taken from the classification track for OWL DL ontologies at the ORE competition 2014 [23]. The prototype was implemented in Java 1.7, using the OWL-API [9] for ontology access, LETHE [17] for computing the annotated uniform interpolants, and MORe [24] as a reasoner in the minimisation step. MORe is a hybrid reasoner that utilises the OWL DL reasoner HermiT [7] together with the $\mathcal{EL}$ reasoner ELK [10]. As it was to be expected that the module as well as the uniform interpolants could often be expressed purely in $\mathcal{EL}$, this reasoner was selected to improve reasoner performance for these cases. In fact, we noticed that the minimalisation step was usually significantly faster when using MORe than when using HermiT.

We were particularly interested in the performance for computing small subsumption modules. In particular, we were interested in the following questions: 1) how well does the method perform in practice for computing LK subsumption modules and minimal subsumption modules, and 2) can minimal subsumption modules in $\mathcal{ALCH}$ be expected to be smaller than modules extracted by alternative methods, such as locality-based modules. Since we expected both the computation of the annotated uniform interpolant, as well as the minimisation step afterwards, to be costly in practice, we computed subsumption modules in a relaxed setting. For this, we used a timeout for the uniform interpolation step. LETHE computes uniform interpolants by eliminating names outside of the specified signature one after the other. If it did not succeed in computing the uniform interpolant within the specified timeout, we continued the computation with the uniform interpolant it computed so far, thus obtaining LK subsumption modules and minimal subsumption modules for an extended signature. This way, we were able to get an upper bound on the size of the minimal subsumption module even if the annotated uniform interpolant was too difficult to compute. Furthermore, note that the annotated uniform interpolant computed in the first step may contain fixpoint expressions, so that the LK subsumption module cannot be minimised in the second step, as we cannot test for entailment of axioms with fixpoint expressions using MORe. To still get an idea about the minimisation potential of our approach, we simply treated entailment of axioms with fixpoint expressions as failure, unless the axiom was syntactically contained in the current module. Note that this may result in a module that is not minimal, similar to when the uniform interpolation procedure created a timeout.

The ontologies for our experiments were selected as follows. We selected our ontologies from the track *"Classification of OWL DL ontologies"* of the OWL Reasoner Evaluation competition 2015, because they provide a small, yet well balanced mix of ontologies with very different properties [23]. As our method only supports $\mathcal{ALCH}$ ontologies, we further removed from each ontology the axioms that were not in $\mathcal{ALCH}$, where we kept n-ary equivalence and disjointness axioms, as well as domain and range restrictions, that can equivalently be expressed as $\mathcal{ALCH}$ concept inclusions. From this set of 315 ontologies, we selected

$\lvert \Sigma \cap N_c \rvert$	10	25	50	100	150
Success Rate	90.3%	92.2%	90.6%	87.7%	85.7%
Minimal	70.3%	71.2%	68.7%	66.5%	66.0%
Fixpoints	19.8%	20.8%	23.6%	21.7%	20.6%
UI Timeout	2.2%	2.0%	2.0%	2.6%	1.7%
Duration (s)	0.4 / 591.9 / 4.1 / 27.1	0.4 / 594.0 / 3.7 / 20.6	0.5 / 592.0 / 6.1 / 32.5	0.9 / 584.5 / 9.1 / 39.1	1.0 / 594.9 / 11.6 / 39.7
Size $\top\bot*$-Mod	0 / 1374 / 164.0 / 256.7	0 / 1021 / 131.0 / 204.4	0 / 1047 / 204.0 / 260.3	1 / 1374 / 305.0 / 352.2	6 / 1242 / 385.0 / 435.8
Size LK Mod	0 / 1054 / 144.0 / 216.7	0 / 723 / 103.5 / 170.1	0 / 835 / 170.0 / 218.5	1 / 1054 / 267.0 / 301.2	6 / 1242 / 342.0 / 382.1
Size Minimised Mod	0 / 1051 / 142.0 / 216.1	0 / 723 / 103.0 / 169.8	0 / 814 / 169.0 / 218.0	1 / 1051 / 264.0 / 300.0	6 / 1234 / 336.0 / 379.9
Size Original Ontology	186 / 8926 / 1792.0 / 2547.5				

Table 1. Results of our evaluation (minimal/maximal/median/average).

those that contained less than 10,000 axioms and more than 150 concept names, resulting in a set of 96 ontologies in total.

The experiment was performed on a server running Ubuntu 15.10 with Intel Xeon 2.50GHz cluster Core 4 Duo CPU with 64GiB RAM. We used our prototype to compute subsumption modules for randomly created signatures containing 10, 25, 50, 100 and 150 concept symbols, were we used 30 samples per signature size. The timeout for the interpolation procedure where set to 5 minutes, while the overall timeout was set to 10 minutes. The results of our experiment are shown in Table 1. The success rate shows the number of runs in which the method succeeded within the timeout. The next rows show the percentage of runs that produced modules which were guaranteed to be minimal (Row 3), that did not guarantee minimality due to fixpoints in the uniform interpolant (Row 4), and that did not guarantee minimality due to a timeout in the interpolation procedure (Row 5). Note that the latter two cases may overlap. We then list the minimal, maximal, median and average duration of computing the module in the successful runs. We obtained median durations between 4.1 and 11.6 seconds, which might be reasonable for daily applications. However, as to be expected, in general our method is more computationally expensive than other methods for module extraction. The following rows compare the sizes of the $\top\bot*$-modules, the LK subsumption modules and the minimised subsumption modules. The LK subsumption modules were on average 14.5%–16.8% smaller than the $\top\bot*$-modules, while the minimised variants differed only little in size to the LK subsumption modules. This shows that in practice, LK subsumption modules provide for good approximations of minimal subsumption modules. The last row shows for comparison the minimal, maximal, median and average size of the input ontologies.

8 Conclusion and Future Work

We presented a method for extracting subsumption modules in $\mathcal{ALCH}$ for signatures Σ s.t. $N_r \subseteq \Sigma$. The method ensures minimality of the extracted modules provided that a uniform interpolant without fixpoint operators can be computed for the given ontology and signature. In the first step, the method computes an annotated uniform interpolant, from which an approximation of the minimal subsumption module, the LK subsumption module can be obtained. This module is then minimised in a subsequent step by comparing entailments of corresponding uniform interpolants. Our evaluation indicates that in most cases, the LK subsumption module is already minimal or close-to minimal, so that the second step can be omitted. As for LK subsumption modules, we do not have a restriction for cyclic ontologies, this means that our method provides for good approximations of minimal subsumption modules for $\mathcal{ALCH}$ ontologies in general. Our evaluation indicates that our method is often able to compute subsumption modules that are smaller than locality-based modules. However, in some cases, the computation of these modules was quite time-consuming. Our implementation uses a timeout for the uniform interpolation step, and computes a subsumption module for the signature for which a uniform interpolant could be computed within that timeout. It would be interesting to see the exact effect of this timeout on the computed subsumption modules. For small timeouts, our implementation would compute an LK subsumption module for an extended signature that in addition contains those concept names that are especially hard for Lethe to eliminate, which usually is only a small fraction of complete signature. It is possible that those LK subsumption modules provide for close approximations of the LK subsumption modules for the given signature, though computable in much shorter time.

There are obvious short-comings of our evaluation that should be addressed in future work. First, the sample size used for the signatures was very small, which is why we are currently running the experiments with a higher number of signatures per ontology. Second, we only compared our method with the syntactical method for computing $\top\bot*$-modules. An alternative obvious choice for comparison would be AMEX [5], which can approximate depleting semantic modules of $\mathcal{ALCQI}$ ontologies. Such a comparison could give insights on whether subsumption modules are in practice smaller than semantic modules, or whether they are mostly similar. Third, it would be interesting to evaluate our method on larger ontologies and not only ontologies with at most 10,000 axioms. Furthermore, it would be interesting to test our method with other implementations for uniform interpolation than Lethe, for example the Ackermann-based method presented in [30].

Open problems with our approach are 1) how to obtain an optimal, practical method in the case uniform interpolants contain fixpoints, and 2) how to compute minimal subsumption modules for signatures that do not contain all role names. In order to tackle these, it might be necessary to use the uniform interpolation method not as a black box, but to modify its implementation to track inferences directly. Again, it is possible that techniques from the area of justification and axiom-pinpointing could be used here.

References

1. 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, New York, NY, USA (2007)

2. Botoeva, E., Konev, B., Lutz, C., Ryzhikov, V., Wolter, F., Zakharyaschev, M.: Inseparability and conservative extensions of description logic ontologies: A survey. In: Pan, J.Z., Calvanese, D., Eiter, T., Horrocks, I., Kifer, M., Lin, F., Zhao, Y. (eds.) Reasoning Web: Logical Foundation of Knowledge Graph Construction and Query Answering: 12th International Summer School 2016, Aberdeen, UK, September 5-9, 2016, Tutorial Lectures. pp. 27–89. Springer International Publishing, Cham (2017), https://doi.org/10.1007/978-3-319-49493-7_2
3. Chen, J., Ludwig, M., Ma, Y., Walther, D.: Zooming in on ontologies: Minimal modules and best excerpts. In: The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part I. pp. 173–189 (2017)
4. Chen, J., Ludwig, M., Walther, D.: On computing minimal $\mathcal{EL}$-subsumption modules. In: Proceedings of the Joint Ontology Workshops 2016 Episode 2: The French Summer of Ontology co-located with the 9th International Conference on Formal Ontology in Information Systems (FOIS 2016), Annecy, France, July 6-9, 2016. (2016), http://ceur-ws.org/Vol-1660/womocoe-paper6.pdf
5. Gatens, W., Konev, B., Wolter, F.: Lower and upper approximations for depleting modules of description logic ontologies. In: Proceedings of the Twenty-first European Conference on Artificial Intelligence. pp. 345–350. IOS Press (2014)
6. Ghilardi, S., Lutz, C., Wolter, F.: Did I damage my ontology? A case for conservative extensions in description logics. In: Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2-5, 2006. pp. 187–197 (2006), http://www.aaai.org/Library/KR/2006/kr06-021.php
7. Glimm, B., Horrocks, I., Motik, B., Stoilos, G., Wang, Z.: HermiT: an OWL 2 reasoner. Journal of Automated Reasoning 53(3), 245–269 (2014)
8. Grau, B.C., Horrocks, I., Kazakov, Y., Sattler, U.: Modular reuse of ontologies: Theory and practice. Journal of Artificial Intelligence Research (JAIR) 31(1), 273–318 (2008)
9. Horridge, M., Bechhofer, S.: The OWL API: A Java API for OWL ontologies. Semantic Web 2(1), 11–21 (2011)
10. Kazakov, Y., Krötzsch, M., Simančík, F.: The incredible ELK. Journal of Automated Reasoning 53(1), 1–61 (2014)
11. Konev, B., Ludwig, M., Walther, D., Wolter, F.: The logical difference for the lightweight description logic $\mathcal{EL}$. Journal of Artificial Intelligence Research (JAIR) 44, 633–708 (2012)
12. Konev, B., Lutz, C., Walther, D., Wolter, F.: Formal properties of modularisation. In: Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization, pp. 25–66. Springer (2009), https://doi.org/10.1007/978-3-642-01907-4_3
13. Konev, B., Lutz, C., Walther, D., Wolter, F.: Model-theoretic inseparability and modularity of description logic ontologies. Artificial Intelligence 203, 66–103 (2013)
14. Kontchakov, R., Wolter, F., Zakharyaschev, M.: Logic-based ontology comparison and module extraction, with an application to DL-Lite. Artificial Intelligence 174(15), 1093–1141 (2010)
15. Koopmann, P., Schmidt, R.A.: Forgetting concept and role symbols in $\mathcal{ALCH}$-ontologies. In: Logic for Programming, Artificial Intelligence, and Reasoning - 19th International Conference, LPAR-19, Stellenbosch, South Africa, December 14-19, 2013. Proceedings. pp. 552–567 (2013), https://doi.org/10.1007/978-3-642-45221-5_37

16. Koopmann, P., Schmidt, R.A.: Count and forget: Uniform interpolation of $\mathcal{SHQ}$-ontologies. In: Demri, S., Kapur, D., Weidenbach, C. (eds.) Automated Reasoning: 7th International Joint Conference, IJCAR 2014, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 19-22, 2014. Proceedings. pp. 434–448. Springer International Publishing, Cham (2014), https://doi.org/10.1007/978-3-319-08587-6_34
17. Koopmann, P., Schmidt, R.A.: Lethe: Saturation-based reasoning for non-standard reasoning tasks. In: Proc. ORE'15. pp. 23–30 (2015)
18. Koopmann, P., Schmidt, R.A.: Uniform interpolation and forgetting for $\mathcal{ALC}$ ontologies with ABoxes. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA. pp. 175–181 (2015), http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9981
19. Kullmann, O., Lynce, I., Marques-Silva, J.: Categorisation of clauses in conjunctive normal forms: Minimally unsatisfiable sub-clause-sets and the lean kernel. In: Theory and Applications of Satisfiability Testing - SAT 2006, 9th International Conference, Seattle, WA, USA, August 12-15, 2006, Proceedings. pp. 22–35 (2006), https://doi.org/10.1007/11814948_4
20. Ludwig, M., Konev, B.: Practical uniform interpolation and forgetting for $\mathcal{ALC}$ TBoxes with applications to logical difference. In: Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria, July 20-24, 2014 (2014), http://www.aaai.org/ocs/index.php/KR/KR14/paper/view/7985
21. Lutz, C., Wolter, F.: Foundations for uniform interpolation and forgetting in expressive description logics. In: IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011. pp. 989–995 (2011), https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-170
22. Nikitina, N., Rudolph, S.: (Non-)succinctness of uniform interpolants of general terminologies in the description logic $\mathcal{EL}$. Artif. Intell. 215, 120–140 (2014), https://doi.org/10.1016/j.artint.2014.06.005
23. Parsia, B., Matentzoglu, N., Gonçalves, R.S., Glimm, B., Steigmiller, A.: The OWL reasoner evaluation (ORE) 2015 competition report. Journal of Automated Reasoning pp. 1–28 (2015)
24. Romero, A.A., Grau, B.C., Horrocks, I.: MORe: Modular combination of owl reasoners for ontology classification. In: International Semantic Web Conference. pp. 1–16. Springer (2012)
25. Romero, A.A., Kaminski, M., Grau, B.C., Horrocks, I.: Module extraction in expressive ontology languages via datalog reasoning. J. Artif. Intell. Res. 55, 499–564 (2016), https://doi.org/10.1613/jair.4898
26. Sattler, U., Schneider, T., Zakharyaschev, M.: Which kind of module should I extract? In: Proceedings of DL'09. CEUR Workshop Proceedings, vol. 477. CEUR-WS.org (2009)
27. Schlobach, S., Cornet, R.: Non-standard reasoning services for the debugging of description logic terminologies. In: IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003. pp. 355–362 (2003), http://ijcai.org/Proceedings/03/Papers/053.pdf
28. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. Web Semantics: science, services and agents on the World Wide Web 5(2), 51–53 (2007)
29. Tsarkov, D., Horrocks, I.: FaCT++ description logic reasoner: System description. Automated reasoning pp. 292–297 (2006)

30. Zhao, Y., Schmidt, R.A.: Forgetting concept and role symbols in $\mathcal{ALCOIH}\mu + (\nabla, \cap)$-ontologies. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. pp. 1345–1352. AAAI Press (2016)