Deductive Module Extraction for Expressive Description Logics

Patrick Koopmann[1] and **Jieying Chen**[2]
[1]Institute for Theoretical Computer Science, Technische Universität Dresden, Germany
[2]SIRIUS Centre, Department of Informatics, University of Oslo, Norway
patrick.koopmann@tu-dresden.de, jieyingc@ifi.uio.no

Abstract

In deductive module extraction, we determine a small subset of an ontology for a given vocabulary that preserves all logical entailments that can be expressed in that vocabulary. While in the literature stronger module notions have been discussed, we argue that for applications in ontology analysis and ontology reuse, deductive modules, which are decidable and potentially smaller, are often sufficient. We present methods based on uniform interpolation for extracting different variants of deductive modules, satisfying properties such as completeness, minimality and robustness under replacements, the latter being particularly relevant for ontology reuse. An evaluation of our implementation shows that the modules computed by our method are often significantly smaller than those computed by existing methods.

1 Introduction

Module extraction in description logics (DLs) is to extract some subset of a given ontology, usually for a given signature of names, that preserves certain properties w.r.t. that signature. Originally motivated by ontology reuse, ontology modularity has been widely used in different areas, such as ontology matching [Jiménez-Ruiz and Grau, 2011] and debugging [Ludwig, 2014], forgetting [Koopmann and Schmidt, 2013], or to improve reasoning [Romero *et al.*, 2012]. In this paper, we focus on applications in *ontology analysis* and *ontology reuse*, and consider a module notion called *deductive modules* [Kontchakov *et al.*, 2010], also investigated under the name of *subsumption modules* [Chen *et al.*, 2017]: given an ontology and a signature Σ of names, a deductive module is a subset of the ontology that preserves all entailments that can be expressed in the DL under consideration using only names from Σ.

In *ontology analysis*, an ontology engineer wants exhibit what an ontology states about some names of interest. For this, seeing as few axioms as possible helps getting the information needed quickly. He might furthermore want to see all axioms that are relevant for entailments in the selected signature. We cover this requirement under the notion of a *complete deductive module*, which turns out to be a stronger notion than the similarly motivated notion of Σ-*essential axioms* discussed in [Grau *et al.*, 2008].

Usually, the module will use additional names than the ones specified. The ontology engineer might thus be interested to know *why* the axioms belong to a module, and how they contribute to entailments in the signature. For standard reasoning services, the necessity of explaining inferences has long been understood and implemented under the service of *justification* [Baader and Peñaloza, 2007; Horridge *et al.*, 2008]. Our approach for computing deductive modules computes a so-called *annotated interpolant*, which shows the entailments in the selected signature, parts of which are annotated with axioms from the ontology that contribute to these entailments, and can thus be seen as an explanation of that module.

Another application of modules is in *ontology reuse* [Jiménez-Ruiz *et al.*, 2008; Grau *et al.*, 2008]. Here the ontology engineer wants to reuse a part of the ontology in another context in which only a subset of the signature is relevant, and thus a module would be sufficient. Here, being complete and explainable might not be as relevant, but *robustness* properties gain importance: specifically, it is not only important that all entailments in the signature are covered by the module, but also that entailments are preserved when further axioms are added: the module should be *replaceable* with the original ontology and still preserve the same entailments.

While most module notions investigated in the literature cover the above properties, often they concern much stronger notions of modules, which can make the problem of optimal module extraction hard. For instance, for the notion of *semantic modules*, already for the light-weight description logic $\mathcal{EL}$ it is undecidable whether a given subset is a module for a given signature, [Konev *et al.*, 2013]. As a consequence, existing algorithms for computing minimal semantic modules can only deal with acyclic $\mathcal{EL}$ [Konev *et al.*, 2012] and DL-Lite [Kontchakov *et al.*, 2010]. Practical implementations, such as the syntactical locality-based module extraction method implemented in the OWL API [Grau *et al.*, 2008], AMEX [Gatens *et al.*, 2014] and PriM [Romero *et al.*, 2016], usually compute approximations of minimal modules that may contain more axioms than necessary.

For deductive modules, the situation looks different. Deciding deductive modules is ExpTime-complete for $\mathcal{EL}$-

ontologies [Lutz and Wolter, 2007] and 2ExpTime-complete for $\mathcal{ALC}$ ontologies [Ghilardi *et al.*, 2006]. For acyclic $\mathcal{ELH}^r$-ontologies, it was shown that deductive modules are substantially smaller than other module types [Chen *et al.*, 2018; 2019]. However, we are not aware of any practical method for computing deductive modules in more expressive DLs such as $\mathcal{ALC}$.

In this paper, we present a method for computing deductive modules in $\mathcal{ALC}$ and $\mathcal{ALCH}$, which are often substantially smaller than modules computed by existing methods. This method is based on a method for *uniform interpolation* for expressive DLs presented in [Koopmann and Schmidt, 2013]. Both uniform interpolants and modules preserve entailments in a specified signature. The difference is that, while a module is always a subset of the ontology, and may use more names than specified, a uniform interpolant may only use names from the specified set and will thus usually be of a syntactical different shape. This can lead to uniform interpolants that contain substantially more complex axioms than the input ontology [Lutz and Wolter, 2011], which is why it is often preferable to use modules rather than uniform interpolants.

Uniform interpolants cover exactly the entailments in the given signature. The main idea of our technique is to track these entailments back to axioms in the original ontology. For this, we label each axiom in the input ontology with a fresh concept name, and then compute a uniform interpolant for an extended signature. The result is an *annotated interpolant*, which links the entailments shown in the uniform interpolant to the axioms in the ontology, and can be used to illustrate how the axioms in the ontology contribute to a module. Using the annotated interpolant, we present methods for computing two types of deductive modules, *complete deductive modules* and *minimised deductive modules*, which we define in the corresponding sections 4 and 5. These modules consider the two contradicting requirements a user might have, *completeness* in terms of covering every axiom relevant to an entailment on the one hand, and *minimality* on the other hand.

To obtain modules that are robust under replacement, we cannot directly re-use the uniform interpolation method that was presented in [Koopmann and Schmidt, 2013], but have to adapt it to support for *universal roles*. The adaptation has the nice side-effect that a computationally expensive rule for eliminating role names is replaced by a simpler rule, so that uniform interpolants can be computed in shorter time. The adapted method is presented in Section 3. Since robustness under replacement might not always be a requirement, in Section 6, we show how the methods can be modified to compute potentially smaller $\mathcal{ALC}$ modules that do not follow this requirement. Finally, in Section 7, we extend all methods to $\mathcal{ALCH}$, for which, in contrast to the other methods, we cannot use uniform interpolation as a black-box method.

We implemented a prototype of our approach which we evaluate and compare with existing methods, both for computing random modules and *atomic decompositions*, a structure representing all self-contained modules of an ontology. Since uniform interpolation can be an expensive operation, we may not always be able to compute optimal modules in short time. Furthermore, our algorithm might require reasoning capabilities that are not supported by state-of-the-art DL reasoners. For those cases, our implementation allows for a flexible way to approximate modules: the more time given, the smaller the modules, and we know when the module is minimal. Our results indicate that in most cases, an approximation is not necessary, and the modules computed by our method are significantly smaller than those computed with existing tools. For proofs and details omitted due to space limitations, we refer to the extended technical report [Koopmann and Chen, 2020].

2 Description Logics and Deductive Modules

We recall the DLs relevant in this paper [Baader *et al.*, 2007], as well as the notions of deductive modules.

Let $\mathsf{N_C}$ and $\mathsf{N_R}$ be two disjoint, countably infinite, sets of respectively *concept names* and *role names*. A *signature* $\Sigma \subseteq \mathsf{N_C} \cup \mathsf{N_R}$ is a finite set of concept names and role names. A *role* is an element $r \in \mathsf{N_R} \cup \{\nabla\}$, where ∇ denotes the *universal role*. Concepts are built according to the syntax rule $C ::= A \mid \neg C \mid C \sqcup C \mid \exists r.C$, where $A \in \mathsf{N_R}$ and r is a role. Further concepts are introduced as abbreviations: $C \sqcap D = \neg(\neg C \sqcup \neg D)$, $\forall r.C = \neg\exists r.\neg C$, $\top = A \sqcup \neg A$ and $\bot = \neg\top$. A *concept inclusion* (CI) is an expression $C \sqsubseteq D$, where C, D are concepts. A *role inclusion* (RI) is an expression $r \sqsubseteq s$, where r, s are roles. CIs and RIs are collectively called *axioms*. The DL $\mathcal{ALCH}^\nabla$ uses all constructors introduced so far. Without RIs, we obtain the DL $\mathcal{ALC}^\nabla$, and without the universal role, we obtain the DLs $\mathcal{ALCH}$ and $\mathcal{ALC}$ respectively. Given a concept/axiom/ontology X, we denote by $\mathsf{sig}(X)$ the *signature of* X, i.e. the concept and role names occurring in X.

The semantics of $\mathcal{ALCH}^\nabla$ 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 $A \in \mathsf{N_C}$ to $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$, every $r \in \mathsf{N_R}$ to a binary relation $r^\mathcal{I} \subseteq \Delta^\mathcal{I}$, that satisfies $\nabla^\mathcal{I} = \Delta^\mathcal{I} \times \Delta^\mathcal{I}$, and is extended to concepts by $(\neg C)^\mathcal{I} := \Delta^\mathcal{I} \backslash C^\mathcal{I}$, $(C \sqcup D)^\mathcal{I} := C^\mathcal{I} \cup D^\mathcal{I}$, and $(\exists r.C)^\mathcal{I} := \{x \in \Delta^\mathcal{I} \mid \exists y \in C^\mathcal{I} : (x, y) \in r^\mathcal{I}\}$. An interpretation $\mathcal{I}$ satisfies a CI $C \sqsubseteq D$ iff $C^\mathcal{I} \subseteq D^\mathcal{I}$, and an RI $r \sqsubseteq s$ if $r^\mathcal{I} \subseteq s^\mathcal{I}$. For an axiom α, we write $\mathcal{I} \models \alpha$ if $\mathcal{I}$ satisfies α. An interpretation $\mathcal{I}$ is a *model* of an ontology $\mathcal{O}$ if $\mathcal{I} \models \alpha$ for all $\alpha \in \mathcal{O}$. 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 $\mathcal{I} \models \alpha$.

Definition 1. *Let $\mathcal{L}$ be a DL, $\mathcal{O}$ an ontology and Σ a signature. Then, a subset $\mathcal{M} \subseteq \mathcal{O}$ is a* deductive $\langle\mathcal{L}, \Sigma\rangle$-module *of $\mathcal{O}$ iff for every $\mathcal{L}$-axiom α with $\mathsf{sig}(\alpha) \subseteq \Sigma$, $\mathcal{O} \models \alpha$ iff $\mathcal{M} \models \alpha$.*

If the DL $\mathcal{L}$ is clear from the context, we may refer to an $\langle\mathcal{L}, \Sigma\rangle$-module simply as Σ-module. Note that deductive modules are related to *conservative extensions*: namely, if $\mathcal{M}$ is a deductive $\langle\mathcal{L}, \Sigma\rangle$-module of $\mathcal{O}$, then $\mathcal{O}$ is a conservative extension of $\mathcal{M}$.

In this paper, we consider the input ontologies to be expressed in $\mathcal{ALC}$ and $\mathcal{ALCH}$. Still, there might be entailments using universal roles that we want to preserve by the module. This is particularly the case for $\mathcal{ALC}$/$\mathcal{ALC}^\nabla$, due to following result from [Konev *et al.*, 2009].

$$\mathrm{R}A \quad \frac{C_1 \sqcup A \quad C_2 \sqcup \neg A}{C_1 \sqcup C_2} \qquad \mathrm{R}\exists \quad \frac{C_1 \sqcup \exists r.D}{C_1 \sqcup \exists \nabla.D}$$

$$\mathrm{RQ} \quad \frac{C_1 \sqcup \mathsf{Q}r.D_1 \quad C_2 \sqcup \forall r.D_2}{C_1 \sqcup C_2 \sqcup \mathsf{Q}r.D_{12}} \qquad \mathsf{Q} \in \{\exists, \forall\}$$

Figure 1: Uniform interpolation calculus.

Theorem 1. *Let $\mathcal{O}$ be an $\mathcal{ALC}$-ontology, Σ a signature, and $\mathcal{M}$ an $\langle \mathcal{ALC}^{\nabla}, \Sigma\rangle$-module of $\mathcal{O}$. Then, for every $\mathcal{ALC}$-ontology $\mathcal{O}'$ with $(sig(\mathcal{O}') \cap sig(\mathcal{O})) \subseteq \Sigma$ and every $\mathcal{ALC}$-axiom α s.t. $sig(\alpha) \subseteq \Sigma$, we have $(\mathcal{O} \cup \mathcal{O}') \models \alpha$ iff $(\mathcal{M} \cup \mathcal{O}') \models \alpha$.*

These $\langle \mathcal{ALC}^{\nabla}, \Sigma\rangle$-modules are called *robust under replacements*, as they can serve as a replacement of the original ontology in a reusing context. Intuitively, if we want to use terms from the signature Σ in a new ontology $\mathcal{O}'$, we can replace $\mathcal{O}$ by its $\langle \mathcal{ALC}^{\nabla}, \Sigma\rangle$-module. We refer to these modules as *robust modules* in the following.

3 Resolution-Based Uniform Interpolation

Uniform interpolants [Lutz and Wolter, 2011] cover exactly the entailments a deductive module should preserve.

Definition 2. *Let $\mathcal{L}$ be a DL, $\mathcal{O}$ an ontology and Σ a signature. Then, a* uniform $\langle \mathcal{L}, \Sigma\rangle$-interpolant *of $\mathcal{O}$ is an ontology $\mathcal{O}^{\Sigma}$ s.t. $sig(\mathcal{O}^{\Sigma}) \subseteq \Sigma$ and for every $\mathcal{L}$-axiom α with $sig(\alpha) \subseteq \Sigma$, $\mathcal{O}^{\Sigma} \models \alpha$ iff $\mathcal{O} \models \alpha$.*

Our method for computing deductive modules uses uniform interpolation to track entailments in the target signature back to the axioms in the original ontology. Our arguments are easier to follow in perspective of the resolution-based method for computing uniform interpolants from in [Koopmann and Schmidt, 2013]. To capture entailments of robust modules, we need to compute uniform $\langle \mathcal{ALC}^{\nabla}, \Sigma\rangle$-interpolants. The method in [Koopmann and Schmidt, 2013] only supports $\mathcal{ALCH}$, which is why we adapt the calculus used by the method to support for universal roles. The rest of the method remains the same. Due to space restrictions, we only give a brief overview of the important steps.

The method proceeds in three steps, which we discuss one after the other. Let $\mathsf{N_D} \subseteq \mathsf{N_C}$ be a special set of concept names not used in the input.

Step 1 transforms all axioms of the ontology into the following form:

$$\top \sqsubseteq C_1 \sqcup \ldots \sqcup C_n \qquad C_i := A \mid \neg A \mid \mathsf{Q}r.D,$$

where $A \in \mathsf{N_C}$, $\mathsf{Q} \in \{\forall, \exists\}$, $r \in \mathsf{N_R}$ and $D \in \mathsf{N_D}$, and at most one literal of the form $\neg D$, $D \in \mathsf{N_D}$, occurs in each axiom. Here, we treat disjunctions as sets, that is, duplicates are silently removed and the order is not important. Furthermore, as it is the same for all axioms, we discard the leading $\top \sqsubseteq$, and call the normalised axiom *clause*.

In **Step 2**, inferences on the names outside the target signature are performed using the rules shown in Figure 1. Here, D_{12} refers to a possibly new definer representing $D_1 \sqcap D_2$, which we introduce, if such a definer does not exist yet, by adding $\neg D_{12} \sqcup D_1$, $\neg D_{12} \sqcup D_2$ to the current clause set. The difference to the calculus in [Koopmann and Schmidt, 2013] is the R$\exists$-rule: since they compute uniform interpolants for a DL without universal roles, instead a more expensive *role resolution* rule is used that makes use of an external reasoner, which is not necessary in our case. Still, in order to eliminate a role name r from a clause, we need to consider all inferences of the RQ-rule on that role name before applying the R$\exists$-rule.

Rules are only applied if the result contains at most one literals of the form $\neg D$, where $D \in \mathsf{N_D}$. In order to perform all relevant inferences on the names outside the target signature, usually further inferences on names inside the target signature are necessary. This is typically the case for the RQ-rule: this rule might trigger the introduction of a new definer D_{12} with the clauses $\neg D_{12} \sqcup D_1$, $\neg D_{12} \sqcup D_2$. If we have also two clauses $\neg D_1 \sqcup C_1' \sqcup A$, $\neg D_2 \sqcup C_2' \sqcup \neg A$, and A is a name we want to perform inferences on, new resolution steps on A become possible only due to the new definers. How to determine which rules have to be used is described in more detail in [Koopmann, 2020]. After all inferences on a name X outside of the signature are performed, we filter out all occurrences of that name.

Finally, in **Step 3**, the introduced definers are eliminated as follows.

1. For every definer D, the clauses $\neg D \sqcup C_1, \ldots, \neg D \sqcup C_n$ are grouped into a single axiom $D \sqsubseteq C_1 \sqcap \ldots \sqcap C_n$.
2. We then repeat following steps as long as as possible:
   (a) Replace $D \sqsubseteq C[D]$, where $C[D]$ is a concept in which D occurs positively, by $D \sqsubseteq \nu X.C[X]$, where $C[X]$ is obtained from $C[D]$ by replacing D by X.
   (b) For every CI $D \sqsubseteq C$ where $D \notin \text{sig}(C)$, remove that axiom and replace all occurrences of D by C.

$\nu X.C[X]$ is a special concept constructor called the *greatest fixpoint*, which corresponds to the limit of the infinite sequence of concepts $C[\top]$, $C[C[\top]]$, $C[C[C[\top]]]$, ... For the formal semantics, we refer to [Calvanese and De Giacomo, 2003]. There exist inputs for which no uniform interpolant without fixpoint operators exists [Lutz and Wolter, 2011].

Lemma 1. *Let $\mathcal{O}$ be an $\mathcal{ALC}$-ontology and Σ a signature. The method always terminates and computes a uniform $\langle \mathcal{ALC}^{\nabla}, \Sigma\rangle$-interpolant of $\mathcal{O}$.*

4 Complete Robust Deductive Modules

For ontology analysis, it might be desirable to see all axioms in an ontology that contribute to entailments in a signature of interest. "Contributing to an entailment" is formally captured by the notion of justifications [Baader and Peñaloza, 2007]: given an ontology $\mathcal{O}$ and an axiom α s.t. $\mathcal{O} \models \alpha$, a *justification for $\mathcal{O} \models \alpha$* is a subset $\mathbf{J} \subseteq \mathcal{O}$ s.t. $\mathbf{J} \models \alpha$ and $\mathbf{J}$ is minimal w.r.t. $\subseteq$. If $\mathcal{O} \not\models \alpha$, there is no justification for $\mathcal{O} \models \alpha$.

Definition 3. *Let $\mathcal{L}$ be a DL, $\mathcal{O}$ an ontology and Σ a signature. An $\langle \mathcal{L}, \Sigma\rangle$-module $\mathcal{M}$ of $\mathcal{O}$ is* complete *iff for every $\mathcal{L}$-*

axiom α s.t. sig$(\alpha) \subseteq \Sigma$ and every justification $\mathbf{J}$ *for* $\mathcal{O} \models \alpha$, $\mathbf{J} \subseteq \mathcal{M}$.

Complete modules contain all axioms that contribute to entailments in the signature of interest, and are thus useful for ontology analysis tasks. Similarly motivated is the *set of* Σ*-essential axioms*, as defined in [Grau *et al.*, 2008]: Given a signature Σ, the set of Σ-essential axioms in $\mathcal{O}$ is the intersection of all subset-minimal Σ-modules in $\mathcal{O}$. However, in the presence of redundancy, Σ-essential axiom sets may "miss" axioms, as already the following simple example illustrates. Take the ontology $\mathcal{O} = \{A \sqsubseteq B \sqcap C, A \sqsubseteq B\}$ and the signature $\Sigma = \{A, B, C\}$. Clearly, both axioms contribute to the entailment $A \sqsubseteq B$, but only the first axiom occurs in a minimal module for Σ. Thus, $A \sqsubseteq B \sqcap C$ is the only Σ-essential axiom, while the complete module would also contain $A \sqsubseteq B$. While this is a very simply example, we note that redundancy is a common phenomenon in real-life ontologies [Grimm and Wissmann, 2011].

To compute complete deductive modules, we track the axioms used during uniform interpolation. For this, we use *axiom annotation*.

Definition 4. *Let $\mathcal{O}$ be an ontology. The* annotation $\mathcal{O}_a$ *of $\mathcal{O}$ is defined as*

$$\mathcal{O}_a = \{\mathsf{A}_{C \sqsubseteq D} \sqcap C \sqsubseteq D \mid C \sqsubseteq D \in \mathcal{O}\},$$

where each $\mathsf{A}_{C \sqsubseteq D}$ is fresh.

We call the concept names $\mathsf{A}_{C \sqsubseteq D}$ *labels*. The following result can be shown using the rule applications used when computing the uniform interpolant as described in Section 3.

Theorem 2. *Let $\mathcal{O}$ be an ontology, Σ a signature, and $\mathcal{O}_a$ the annotation of $\mathcal{O}$. Additionally, let $\mathcal{O}_a^\Sigma$ be the uniform $\langle \mathcal{ALC}^\nabla, \Sigma' \rangle$-interpolant of $\mathcal{O}_a$ for $\Sigma' = \Sigma \cup \{\mathsf{A}_\alpha \mid \alpha \in \mathcal{O}\}$. Then, the ontology*

$$\mathcal{M} = \{\alpha \mid \mathsf{A}_\alpha \in \mathit{sig}(\mathcal{O}_a^\Sigma)\}$$

is a complete deductive $\langle \mathcal{ALC}^\nabla, \Sigma \rangle$-module of $\mathcal{O}$ for Σ.

In the above definition, we call $\mathcal{O}_a^\Sigma$ the *annotated uniform $\langle \mathcal{ALC}^\nabla, \Sigma \rangle$-interpolant of $\mathcal{O}$*. In addition in helping us find a complete module, the annotated uniform interpolant provides for a link between the module and the uniform interpolant that allows us to better understand *how* the axioms contribute to entailments in the target signature.

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

$$\alpha_1 = \exists r.\top \sqsubseteq A \sqcup B \quad \alpha_2 = \exists r.A \sqsubseteq B \quad \alpha_3 = \exists r.B \sqsubseteq A$$

The uniform interpolant of $\mathcal{O}$ for $\Sigma = \{A, r\}$, consisting of the following axioms, captures all entailments of $\mathcal{O}$ over Σ:

$$\exists r.\exists r.A \sqsubseteq A \qquad \exists r.(\exists r.\top \sqcap \neg A) \sqsubseteq A$$

To understand which axioms contribute to these entailments, we first annotate $\mathcal{O}$:

$$\mathsf{A}_{\alpha_1} \sqcap \exists r.\top \sqsubseteq A \sqcup B$$
$$\mathsf{A}_{\alpha_2} \sqcap \exists r.A \sqsubseteq B \qquad \mathsf{A}_{\alpha_3} \sqcap \exists r.B \sqsubseteq A$$

Algorithm 1 Computing minimal deductive modules

Input: ontology $\mathcal{O}$, signature Σ, integer n
Output: a minimal deductive module $\mathcal{M}$

```
Let M = O, Q = divide(O, n)
while L ≠ ∅ do
    B := head(Q), Q := tail(Q)
    if (M \ B)^Σ ⊨ O^Σ then
        M := M \ B
    else
        L := divideInHalf(B) :: Q
    end if
end while
return M
```

The annotated uniform interpolant of $\mathcal{O}$ for Σ is then the following:

$$\mathsf{A}_{\alpha_3} \sqcap \exists r.(\mathsf{A}_{\alpha_2} \sqcap \exists r.A) \sqsubseteq A$$
$$\mathsf{A}_{\alpha_3} \sqcap \exists r.(\mathsf{A}_{\alpha_1} \sqcap \exists r.\top \sqcap \neg A) \sqsubseteq A$$

We can directly read off the complete Σ-module for $\mathcal{O}$, which is $\mathcal{O}$ itself. We can furthermore understand how the axioms contribute to entailments over Σ: α_3 is needed for every non-trivial entailment over Σ, while α_2 and α_1 each contribute to what is under the different existential role restrictions.

5 Minimised Robust Deductive Modules

Both for *ontology reuse* or *ontology understanding*, if we are not interested in completeness, we would prefer a module that does not contain more axioms than necessary.

Definition 5. *A deductive $\langle \mathcal{L}, \Sigma \rangle$-module $\mathcal{M}$ of an ontology $\mathcal{O}$ is* minimal *if no subset $\mathcal{M}' \subset \mathcal{M}$ is a deductive $\langle \mathcal{L}, \Sigma \rangle$-module of $\mathcal{O}$* .

We note that the number of minimal modules can be exponential in the size of the ontology. We aim neither to compute the complete set of minimal modules, nor the module that contains the smallest number of axioms, but just to compute any minimal module. Our method for minimising modules guarantees minimality if the computed uniform interpolants does not contain fixpoints. However, as our evaluation shows, it still often results in modules that are much smaller than the complete module.

Minimal deductive modules can be seen as *justifications* for the corresponding uniform interpolant. Computing justifications has been studied in the literature under the name of axiom pinpointing [Baader and Peñaloza, 2007], and there exist implementations even as part of the OWL API [Horridge *et al.*, 2008]. However, those methods are optimised for justifying smaller axioms, and generally do not perform well for large entailments such as complex uniform interpolants. Our algorithm uses ideas from axiom pinpointing, but is optimised for our particular use-case.

Algorithm 1 computes a minimal deductive module by removing all superfluous axioms from $\mathcal{O}$. Similar algorithms can be found in [Kontchakov *et al.*, 2010; Chen *et al.*, 2018].

Since entailment checks can be expensive, instead of checking axioms one by one, we check entailment of a set of axioms, called *bubble*, in each step. *divide*$(\mathcal{O}, n)$ partitions the axioms in $\mathcal{O}$ into a sequence $\mathcal{Q}$ of bubbles of size n plus the remainder, while *divideInHalf*$(\mathcal{B})$ splits a bubble into two equally sized bubbles to be attached at the beginning of the current sequence $\mathcal{Q}$.

Note that in Line 4, we could also test for $\mathcal{M} \setminus \mathcal{B} \models \mathcal{O}^\Sigma$, in which case our algorithm would directly correspond to an algorithm for computing justifications. However, this test can be computationally expensive, because $\mathcal{O}^\Sigma$ can be large, or even impossible, if $\mathcal{O}^\Sigma$ can contains fixpoints which are not supported by DL reasoners. Note that in most cases, there will be no syntactical overlap between $\mathcal{O}^\Sigma$ and $\mathcal{M}$, since $\mathcal{O}^\Sigma$ has to be fully in Σ. In contrast, the interpolants $(\mathcal{M} \setminus \mathcal{B})^\Sigma$ and $\mathcal{O}^\Sigma$ use the same signature, and thus can overlap syntactically, if computed wisely. As we can decide $(\mathcal{M} \setminus \mathcal{B})^\Sigma \models \mathcal{O}^\Sigma$ by only testing entailment of the axioms in $\mathcal{O}^\Sigma \setminus (\mathcal{M} \setminus \mathcal{B})^\Sigma$, those syntactical overlaps can reduce reasoning times significantly, and we may not even have to consider all fixpoint expressions occurring in $\mathcal{O}^\Sigma$ for reasoning. If greatest fixpoints occur only in $(\mathcal{M} \setminus \mathcal{B})^\Sigma$, these can be simulated using auxiliary names [Koopmann and Schmidt, 2013].

To compute interpolants of the different subsets $(\mathcal{M} \setminus \mathcal{B})$ of the ontology fast, we use the following lemma.

Lemma 2. *Let $\mathcal{O}$ be an $\mathcal{ALC}$-ontology, Σ a signature and $\mathcal{O}_a^\Sigma$ an annotated uniform $\langle \mathcal{ALC}^\nabla, \Sigma\rangle$-interpolant of $\mathcal{O}$. Additionally, let $\mathcal{O}_1 \subseteq \mathcal{O}$. Then, the ontology*

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

is a uniform $\langle \mathcal{ALC}^\nabla, \Sigma\rangle$-interpolant of $\mathcal{O}_1$ for Σ.

Thus, before entering Algorithm 1, we compute the annotated uniform interpolant of the input ontology, from which then all other required uniform interpolants are cheaply obtained by replacing the labels with $\top$ and $\bot$. We further use some syntactical simplification rules to replace tautological as well as contradictory subconcepts by $\top$ and $\bot$, and to remove tautological axioms. The interpolant $\mathcal{O}^\Sigma$ is replaced by the logically equivalent $\mathcal{M}^\Sigma$ when it turns out to be smaller.

Example 2. *Continuing on Example 1, we obtain for $\{\alpha_1, \alpha_2\}$ after simplification the uniform interpolant*

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

of which the second axiom would be removed by simplification. For $\{\alpha_2, \alpha_3\}$, we similarly obtain

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

of which the second axiom would be removed. For $\{\alpha_1, \alpha_2\}$ we only obtain tautological axioms. None of these interpolants entails the uniform interpolant of $\mathcal{O}$, which is why $\mathcal{O}$ is also a minimal module of itself.

6 Deductive Modules for Pure $\mathcal{ALC}$

The previous sections considered uniform interpolants and deductive modules for $\mathcal{ALC}^\nabla$. For applications in ontology analysis where robustness under replacements is not a requirement, it might be sufficient to preserve entailments in pure $\mathcal{ALC}$, for which the modules can be smaller. Our method for uniform interpolation introduces universal roles only under existential restrictions. This allows for a simple way of generating corresponding uniform interpolants for $\mathcal{ALC}$.

Lemma 3. *Let $\mathcal{O}$ be an $\mathcal{ALC}^\nabla$ ontology in which the universal role only occurs in concepts of the form $\exists\nabla.C$, and where every $\exists\nabla.C$ occurs only positively. Additionally, let $\mathcal{O}'$ the result of replacing every $\exists\nabla.C$ in $\mathcal{O}$ by $\bot$ if $\mathcal{O} \models C \sqsubseteq \bot$, and otherwise by $\top$. Then, for every $\mathcal{ALC}$-axiom α, $\mathcal{O} \models \alpha$ iff $\mathcal{O}' \models \alpha$.*

When we apply this lemma on an annotated uniform interpolant, we lose however all annotations that occur in expressions of the form $\exists\nabla.C$. We thus have to extend our mechanism for tracking inferences using annotations. Let $\mathcal{O}$ be an ontology, Σ a signature and $\mathcal{O}_a$ the annotation of $\mathcal{O}$. Then, the *extended annotation* of $\mathcal{O}$ for Σ is defined as

$$\mathcal{O}_{ea} = \mathcal{O}_a \cup \{\exists r.\mathsf{A}_\alpha \sqsubseteq \mathsf{A}_\alpha \mid r \in \mathrm{sig}(\mathcal{O}), \alpha \in \mathcal{O}\}.$$

Correspondingly, we obtain the *annotated uniform $\langle \mathcal{ALC}, \Sigma\rangle$-interpolant* based on this extended annotation, in the same way as we obtain the annotated uniform $\langle \mathcal{ALC}^\nabla, \Sigma\rangle$-interpolant from the annotation.

The additional axioms make sure that the label information is preserved when eliminating a role name r. Specifically, they make sure that for every justification $\mathbf{J}$ of an entailment $\mathcal{O} \models C \sqsubseteq D$, we also have $\mathcal{O}_{ea} \models \bigsqcap_{\alpha \in \mathbf{J}} A_\alpha \sqcap C \sqsubseteq D$. In $\mathcal{O}_a$, there would be a corresponding entailment $\mathcal{O}_a \sqsubseteq C' \sqsubseteq D'$, in which however the labels can occur under role restrictions in C' and D'. Intuitively, the additional axioms in $\mathcal{O}_{ea}$ allow to "pull" these labels out of the role restrictions to make sure that $\mathcal{O}_{ea} \models \bigsqcap_{\alpha \in \mathbf{J}} A_\alpha \sqcap C \sqsubseteq D$.

With this adaptation, we can use the methods presented in Section 4 and 5 for computing complete deductive modules as well as minimised deductive modules for entailments in $\mathcal{ALC}$, based on the following results.

Theorem 3. *Let $\mathcal{O}$ be an ontology, Σ a signature, and $\mathcal{O}_{ea}$ the extended annotation of $\mathcal{O}$ for Σ. Additionally, let $\mathcal{O}_{ea}^\Sigma$ be the uniform $\langle \mathcal{ALC}, \Sigma'\rangle$-interpolant of $\mathcal{O}_{ea}$, where $\Sigma' = \Sigma \cup \{\mathsf{A}_\alpha \mid \alpha \in \mathcal{O}\}$. Then, $\mathcal{M} = \{\alpha \mid \mathsf{A}_\alpha \in sig(\mathcal{O}_{ea}^\Sigma)\}$ is a complete deductive $\langle \mathcal{ALC}, \Sigma\rangle$-module of $\mathcal{O}$ for Σ.*

Lemma 4. *Let $\mathcal{O}$ be an ontology, Σ a signature and $\mathcal{O}_{ea}^\Sigma$ an annotated uniform $\langle \mathcal{ALC}, \Sigma\rangle$-interpolant of $\mathcal{O}$ for Σ. Additionally, let $\mathcal{O}_1 \subseteq \mathcal{O}$. Then, the ontology*

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

is a uniform $\langle \mathcal{ALC}, \Sigma\rangle$-interpolant of $\mathcal{O}_1$ for Σ.

7 Handling Role Hierarchies

Though we presented one uniform interpolation method in this paper, in general, all methods presented here can be combined with any uniform interpolation method for the respective DL. This approach fails however for $\mathcal{ALCH}$, because we cannot annotate RIs. Here, we have to modify the uniform interpolation procedure so that it tracks usage of role axioms itself. The rules shown in Figure 2 are based on rules from the

$$\mathsf{R}\exists\sqsubseteq \quad \frac{C \sqcup \exists r.D \qquad r \sqsubseteq s \in \mathcal{O}}{\neg \mathsf{A}_{r\sqsubseteq s} \sqcup C \sqcup \exists s.D}$$

$$\mathsf{R}\forall\sqsubseteq \quad \frac{C \sqcup \forall r.D \qquad s \sqsubseteq r \in \mathcal{O}}{\neg \mathsf{A}_{r\sqsubseteq s} \sqcup C \sqcup \forall s.D}$$

Figure 2: Additional uniform interpolation rules for $\mathcal{ALCH}$ tracking role hierarchy axioms.

original calculus in [Koopmann and Schmidt, 2013], and extend the calculus presented in Section 3 by explicitly tracking used role inclusion axioms. Note that the resulting method is not sound anymore for computing uniform interpolants, and can thus only be used for computing annotated interpolants.

In our implementation, these rules are either used on their own, when the aim is to eliminate a role name r, or in combination with the RQ-rule. In the latter case, we do not perform the individual inferences of these rules explicitly, but determine in one step which additional disjuncts have to be added to the newly derived clause, and which role is used in the role restriction. For instance, if we have the clauses $C_1 \sqcup \forall r.D_1$ and $C_2 \sqcup \exists s.D_2$ and there exist a justification $\mathbf{J}$ for $\mathcal{O} \models s \sqsubseteq r$, we derive

$$C_1 \sqcup C_2 \sqcup \exists s.D_{12} \sqcup \bigsqcup_{r_1 \sqsubseteq r_2 \in \mathbf{J}} \mathsf{A}_{r_1 \sqsubseteq r_2},$$

in one step. The resulting method is applied on the (extended) annotation of an ontology as before, resulting in the annotated uniform $\langle \mathcal{L}, \Sigma\rangle$-interpolant for respectively $\mathcal{L} = \mathcal{ALCH}^{\nabla}$ and $\mathcal{L} = \mathcal{ALCH}$.

Entailments of RIs cannot be handled in this way, but are not hard to obtain from the set of RIs. For a set $\mathcal{R}$ of RIs, the uniform Σ-interpolant $\mathcal{R}^{\Sigma} = \{r \sqsubseteq s \mid r, s \in \Sigma, \mathcal{O} \models r \sqsubseteq s\}$ can be easily computed by following the RIs in $\mathcal{R}$. We thus compute the uniform interpolant of the RIs directly.

Theorem 4. *Let $\mathcal{L} \in \{\mathcal{ALCH}, \mathcal{ALCH}^{\nabla}\}$, $\mathcal{O}$ be an $\mathcal{ALCH}$-ontology with RIs $\mathcal{R}$, Σ a signature and $\mathcal{O}_a^{\Sigma}$ an annotated uniform $\langle \mathcal{L}, \Sigma\rangle$-interpolant of $\mathcal{O}$ for Σ. Additionally, let $\mathcal{R}^{\Sigma}$ be a uniform Σ-interpolant of $\mathcal{R}$. Then, the ontology*

$$\mathcal{M} = \{\alpha \in \mathcal{O} \mid \mathsf{A}_\alpha \in sig(\mathcal{O}_a^{\Sigma})\} \cup \bigcup_{\beta \in \mathcal{R}^{\Sigma}} \mathbb{J}(\mathcal{R} \models \beta)\},$$

where $\mathbb{J}(\mathcal{R} \models \beta)$ is the union of all justifications for $\mathcal{R} \models \beta$, is a complete deductive $\langle \mathcal{L}, \Sigma\rangle$-module of $\mathcal{O}$.

Lemma 5. *Let $\mathcal{L} \in \{\mathcal{ALCH}, \mathcal{ALCH}^{\nabla}\}$, $\mathcal{O}$ be an $\mathcal{ALCH}$-ontology with RIs $\mathcal{R}$, Σ a signature and $\mathcal{O}_a^{\Sigma}$ an annotated uniform $\langle \mathcal{L}, \Sigma\rangle$-interpolant of $\mathcal{O}$ for Σ. Additionally, let $\mathcal{O}_1 \subseteq \mathcal{O}$. Then, the ontology*

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

is a uniform $\langle \mathcal{L}, \Sigma\rangle$-interpolant of $\mathcal{O}_1$.

We use this approach for computing minimised deductive modules with Algorithm 1. For the complete deductive module, we just need to additionally collect the justifications of the entailed RIs.

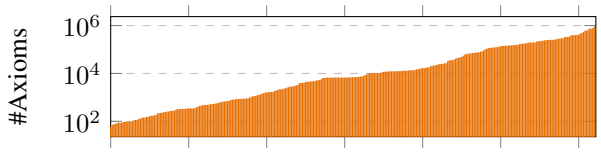

Figure 3: Distribution of ontology-sizes in the corpus.

8 Proof of Concept Evaluation

We implemented our technique for extracting complete and minimal $\mathcal{ALCH}^{\nabla}$-modules in Java and evaluated it on a varied corpus of ontologies. For this, we implemented the calculus consisting of the rules in Figures 1 and 2, as well as the module extraction techniques, as extension to the uniform interpolation tool LETHE [Koopmann, 2020].[1] To keep the modules small, we used the redundancy elimination rules presented in [Koopmann and Schmidt, 2013], and applied a set of syntactic, equivalence-preserving simplification rules on the computed uniform interpolants. Uniform interpolation is an expensive operation, which becomes more challenging in the presence of labels. However, we observed that even for the hardest cases, the majority of names could usually be eliminated fast. To acknowledge this, we implemented a timeout within the uniform interpolation procedure: once the timeout is reached, we output and simplify the annotated uniform interpolant computed so far. For checking entailments, we used the reasoner HermiT [Glimm *et al.*, 2014].

We evaluated our method on ontologies in the DL Classification track of the OWL Reasoner Evaluation (ORE) 2015 [Parsia *et al.*, 2017]. This corpus contains a balanced and varied mix of ontologies in expressive DLs, which have been taken from the corpora MOWLCorp, BioPortal and the Oxford Ontology Repository, and thus contains many real-life ontologies. From each ontology, we removed axioms that could not be equivalently expressed in $\mathcal{ALCH}$, and removed from the resulting set ontologies that had less than 100 names or more than 100,000 axioms, resulting in a set of 227 ontologies. The sizes of the ontologies, measured in number of axioms, can be seen in Figure 3. All experiments were performed on an Intel Core i5-4590 CPU with 3.30GHz and 32 GB RAM.

8.1 Complete and Minimised Deductive Modules

To evaluate the performance of our method for small modules, we randomly generated 60 signatures per ontology of 100 names each, including both concept and role names, where each name was selected with a probability proportional to its number of occurrences in the ontology. We used a timeout of 5 minutes for the uniform interpolation procedure, and a timeout of 10 minutes in complete. We compared the results with modules computed by the two other alternatives for computing modules in expressive DLs: $\top\bot^*$-modules as implemented in the OWL-API [Grau *et al.*, 2008], and AMEX modules [Gatens *et al.*, 2014].

In 90.9% of cases, uniform interpolation did not cause a timeout, and in 96.4% of cases, we could generate a (possi-

[1] https://lat.inf.tu-dresden.de/~koopmann/LETHE (version 0.65)

	Category 1	Category 2
$\top\bot^*$	0 / 65,566 / 1,022 / 196	0 /1,109 / 232 / 236
AMEX	0 / 2,333 / 258 / 142	N/A
CompMod.	0 / 1,272 / 241 / 139	0 / 948 / 156 / 190
MinMod.	0 / 1,179 / 179 / 119	0 / 630 / 122 / 148

Table 1: Size of different modules (min. / max. / avg. /med.)

bly approximated) module. When the uniform interpolation step was stopped due to the timeout, an average of 2 names was not eliminated. In 83.4% of cases, we were able to also minimise the module within the time limit. As was to be expected, our method was more expensive than the other methods for module extraction. On average, computing the complete and minimised module took respectively 22 and 44 seconds, while each $\top\bot^*$-module was computed within 5 seconds. Computing the AMEX-module based on the precomputed $\top\bot^*$-modules always took less than a second.

Statistics on the sizes of modules are shown in Table 1. As AMEX only supports acyclic $\mathcal{ALC}$ ontologies and cannot handle role hierarchies, 143 ontologies could not be processed by AMEX. Category 1 regards the cases that were supported by AMEX, and Category 2 the remaining cases. As we can see in Table 1, the size of minimised modules is significantly smaller than $\top\bot^*$-modules and, on average, considerably smaller than AMEX-modules and complete modules.

8.2 Self-Contained Modules and Atomic Decompositions

A *self-contained* module $\mathcal{M}$ for a signature Σ is a module that is also a module for its own signature, i.e. $\Sigma \cup \text{sig}(\mathcal{M})$. Our method of computing complete modules can be easily extended to compute self-contained modules, where after the computation of the module, we update the signature with the signature of the computed module, and repeat the procedure until the size of modules does not change anymore. While in Section 8.1, we had to use random samples of signatures of a specified size, it is possible to construct a structure that represents all complete self-contained modules, thus allowing for more significant insights on how module sizes behave in general. Specifically, we use the *atomic decomposition* (AD) [Vescovo *et al.*, 2011], a structure initially developed for locality-based modules such as $\top\bot^*$-modules, and represents all possible modules. The AD of an ontology partitions the ontology into a set of *atoms*, so that every module is the union of atoms. Here, an atom $\mathfrak{a}$ of an ontology $\mathcal{O}$ is a set of axioms such that for every module $\mathcal{M}$ of $\mathcal{O}$, $\mathfrak{a} \subseteq \mathcal{M}$ or $\mathfrak{a} \cap \mathcal{M} = \emptyset$ w.r.t. any signature. Larger atoms can lead to larger modules, as a module with one axiom from an atom always contains all other axioms in that atom. Thus, the smaller the atoms, the more fine-grained is the structure of the atomic decomposition, and the smaller are modules in general.

We computed atomic decompositions for the 171 ontologies in our corpus that contained less than 10,000 axioms, using both $\top\bot^*$-modules and complete self-contained modules computed by our method, where this time, we used a timeout of 30 seconds for uniform interpolation, so that the AD was approximated in some cases. Our method produced much less large atoms, indicating a finer granularity of self-contained complete axioms compared to $\top\bot^*$-modules. In the following, we call atoms generated by our method *complete-atoms*, and atoms generated using $\top\bot^*$-modules *star-atoms*. For 70.1% of the ontologies, each complete-atom contained at most 4 axioms, which was only the case for 39.8% of the ontologies regarding star-atoms. The 90% quantile for the maximal atom size per ontology was 9 for complete-atoms, and 118 for star-atoms. This shows that self-contained modules computed by our method can be much smaller than $\top\bot^*$-modules in general.

9 Conclusion and Future Work

We presented a practical method for computing deductive modules in $\mathcal{ALC}$ and $\mathcal{ALCH}$ ontologies based on uniform interpolation. Our evaluation showed that deductive modules are often considerably smaller than other types of modules. In the future, we would like to investigate the impact timeouts in the uniform interpolation method have on the module size, and whether precomputed atomic decompositions can be used to speed up the computation. Furthermore, we want to investigate whether uniform interpolation methods for more expressive DLs, such as the approaches in [Zhao and Schmidt, 2018; Koopmann and Schmidt, 2014], can be used. While for some more expressive DLs, deductive modules are undecidable, it could still be possible that approximations result in smaller modules than for existing module types.

Acknowledgements

Patrick Koopmann is partially supported by the DFG grant 389793660 as part of TRR 248 (see https://www.perspicuous-computing.science). Jieying Chen is supported by the SIRIUS centre, which is funded by the Norwegian Research Council, project number 237898 and is co-funded by its partner companies. We would like to thank Boris Konev for his help with the AMEX system, and Marco Benedetti for providing the solver sKizzo used by AMEX.

References

[Baader and Peñaloza, 2007] Franz Baader and Rafael Peñaloza. Axiom pinpointing in general tableaux. In *Proc. of TABLEAUX'07*, pages 11–27. Springer, 2007.

[Baader *et al.*, 2007] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The description logic handbook: theory, implementation, and applications*. Cambridge University Press, 2007.

[Calvanese and De Giacomo, 2003] Diego Calvanese and Giuseppe De Giacomo. Expressive description logics. In *The Description Logic Handbook: Theory, Implementation, and Applications*, pages 178–218. Cambridge University Press, 2003.

[Chen *et al.*, 2017] Jieying Chen, Michel Ludwig, Yue Ma, and Dirk Walther. Zooming in on ontologies: Minimal modules and best excerpts. In *Proc. of ISWC'17*, pages 173–189. Springer, 2017.

[Chen *et al.*, 2018] Jieying Chen, Michel Ludwig, and Dirk Walther. Computing minimal subsumption modules of ontologies. In *Proc. of GCAI'18*, pages 41–53. EasyChair, 2018.

[Chen *et al.*, 2019] Jieying Chen, Michel Ludwig, Yue Ma, and Dirk Walther. Computing minimal projection modules for $\mathcal{ELH}^r$-terminologies. In *Proc. of JELIA'19*, volume 11468, pages 355–370. Springer, 2019.

[Gatens *et al.*, 2014] William Gatens, Boris Konev, and Frank Wolter. Lower and upper approximations for depleting modules of description logic ontologies. In *Proc. of ECAI'14*, pages 345–350. IOS Press, 2014.

[Ghilardi *et al.*, 2006] Silvio Ghilardi, Carsten Lutz, and Frank Wolter. Did I damage my ontology? A case for conservative extensions in description logics. In *Proc. of KR'06*, pages 187–197. AAAI Press, 2006.

[Glimm *et al.*, 2014] Birte Glimm, Ian Horrocks, Boris Motik, Giorgos Stoilos, and Zhe Wang. HermiT: An OWL 2 reasoner. *J. Autom. Reasoning*, 53(3):245–269, 2014.

[Grau *et al.*, 2008] Bernardo Cuenca Grau, Ian Horrocks, Yevgeny Kazakov, and Ulrike Sattler. Modular reuse of ontologies: Theory and practice. *J. Artif. Intell. Res.*, 31:273–318, 2008.

[Grimm and Wissmann, 2011] Stephan Grimm and Jens Wissmann. Elimination of redundancy in ontologies. In *Proceedings of ESWC 2011*, volume 6643 of *LNCS*, pages 260–274. Springer, 2011.

[Horridge *et al.*, 2008] Matthew Horridge, Bijan Parsia, and Ulrike Sattler. Laconic and precise justifications in OWL. In *Proc. of ISWC'08*, volume 5318 of *LNCS*, pages 323–338. Springer, 2008.

[Jiménez-Ruiz and Grau, 2011] Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. Logmap: Logic-based and scalable ontology matching. In *Proc. of ISWC'11*, pages 273–288. Springer, 2011.

[Jiménez-Ruiz *et al.*, 2008] Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, Ulrike Sattler, Thomas Schneider, and Rafael Berlanga Llavori. Safe and economic re-use of ontologies: A logic-based methodology and tool support. In *Proc. of ESWC'08*, pages 185–199. STII, 2008.

[Konev *et al.*, 2009] Boris Konev, Carsten Lutz, Dirk Walther, and Frank Wolter. Formal properties of modularisation. In *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, volume 5445 of *LNCS*, pages 25–66. Springer, 2009.

[Konev *et al.*, 2012] Boris Konev, Michel Ludwig, Dirk Walther, and Frank Wolter. The logical difference for the lightweight description logic $\mathcal{EL}$. *J. Artif. Intell. Res.*, 44:633–708, 2012.

[Konev *et al.*, 2013] Boris Konev, Carsten Lutz, Dirk Walther, and Frank Wolter. Model-theoretic inseparability and modularity of description logic ontologies. *Artif. Intell.*, 203:66–103, 2013.

[Kontchakov *et al.*, 2010] Roman Kontchakov, Frank Wolter, and Michael Zakharyaschev. Logic-based ontology comparison and module extraction, with an application to DL-Lite. *Artif. Intell.*, 174(15):1093–1141, 2010.

[Koopmann and Chen, 2020] Patrick Koopmann and Jieying Chen. Deductive module extraction for expressive description logics (extended version). LTCS-Report 20-06, Chair for Automata Theory, Institute for Theoretical Computer Science, Technische Universität Dresden, Dresden, Germany, 2020. See https://lat.inf.tu-dresden.de/research/reports.html.

[Koopmann and Schmidt, 2013] Patrick Koopmann and Renate A. Schmidt. Forgetting concept and role symbols in $\mathcal{ALCH}$-ontologies. In *Proc. of LPAR'13*, pages 552–567. Springer, 2013.

[Koopmann and Schmidt, 2014] Patrick Koopmann and Renate A. Schmidt. Count and forget: Uniform interpolation of $\mathcal{SHQ}$-ontologies. In *Proc. of IJCAR'14*, volume 8562 of *LNCS*, pages 434–448. Springer, 2014.

[Koopmann, 2020] Patrick Koopmann. LETHE: Forgetting and uniform interpolation for expressive description logics. *KI - Künstliche Intelligenz, German Journal of Artificial Intelligence*, 2020.

[Ludwig, 2014] Michel Ludwig. Just: a tool for computing justifications w.r.t. $\mathcal{ELH}$ ontologies. In *Proc. of ORE'14*, pages 1–7. CEUR-WS.org, 2014.

[Lutz and Wolter, 2007] Carsten Lutz and Frank Wolter. Conservative extensions in the lightweight description logic $\mathcal{EL}$. In *Proc. of CADE'07*, volume 4603, pages 84–99. Springer, 2007.

[Lutz and Wolter, 2011] Carsten Lutz and Frank Wolter. Foundations for uniform interpolation and forgetting in expressive description logics. In *Proc. of IJCAI'11*, pages 989–995. AAAI Press, 2011.

[Parsia *et al.*, 2017] Bijan Parsia, Nicolas Matentzoglu, Rafael S. Gonçalves, Birte Glimm, and Andreas Steigmiller. The OWL Reasoner Evaluation (ORE) 2015 competition report. *J. Autom. Reasoning*, 59(4):455–482, 2017.

[Romero *et al.*, 2012] Ana Armas Romero, Bernardo Cuenca Grau, and Ian Horrocks. MORe: Modular combination of OWL reasoners for ontology classification. In *Proc. of ISWC'12*, pages 1–16. Springer, 2012.

[Romero *et al.*, 2016] Ana Armas Romero, Mark Kaminski, Bernardo Cuenca Grau, and Ian Horrocks. Module extraction in expressive ontology languages via datalog reasoning. *J. Artif. Intell. Res.*, 55:499–564, 2016.

[Vescovo *et al.*, 2011] Chiara Del Vescovo, Bijan Parsia, Ulrike Sattler, and Thomas Schneider. The modular structure of an ontology: Atomic decomposition. In *Proc. of IJCAI'11*, pages 2232–2237. AAAI Press, 2011.

[Zhao and Schmidt, 2018] Yizheng Zhao and Renate A. Schmidt. FAME: an automated tool for semantic forgetting in expressive description logics. In *Proc. of IJCAR'18*, volume 10900 of *LNCS*, pages 19–27. Springer, 2018.