Efficient Computation of General Modules for $\mathcal{ALC}$ Ontologies

Hui Yang[1] , **Patrick Koopmann**[2] , **Yue Ma**[1] and **Nicole Bidoit**[1]
[1]LISN, CNRS, Université Paris-Saclay
[2] Vrije Universiteit Amsterdam, The Netherlands
yang@lisn.fr, p.k.koopmann@vu.nl, ma@lisn.fr, nicole.bidoit@lisn.fr

Abstract

We present a method for extracting general modules for ontologies formulated in the description logic $\mathcal{ALC}$. A module for an ontology is an ideally substantially smaller ontology that preserves all entailments for a user-specified set of terms. As such, it has applications such as ontology reuse and ontology analysis. Different from classical modules, general modules may use axioms not explicitly present in the input ontology, which allows for additional conciseness. So far, general modules have only been investigated for lightweight description logics. We present the first work that considers the more expressive description logic $\mathcal{ALC}$. In particular, our contribution is a new method based on uniform interpolation supported by some new theoretical results. Our evaluation indicates that our general modules are often smaller than classical modules and uniform interpolants computed by the state-of-the-art, and compared with uniform interpolants, can be computed in a significantly shorter time. Moreover, our method can be used for, and in fact improves, the computation of uniform interpolants and classical modules.

1 Introduction

Ontologies are used to formalize terminological knowledge in many domains such as biology, medicine and the Semantic Web. Usually, they contain a set of statements (axioms) about concept and role names (unary and binary predicates). Using a formalization based on description logics (DLs) allows DL reasoners to infer implicit information from an ontology. Modern ontologies are often large and complex, which can make ontology engineering challenging. For example, as of 3 January 2023, the medical ontology SNOMED CT [Donnelly and others, 2006], used in the health-case systems of many countries, formalizes over 360,000 concepts, and the BioPortal repository of ontologies from the bio-medical domain [Noy *et al.*, 2009] currently hosts 1,043 ontologies that use over 14 million concepts. Often, one is not interested in the entire content of an ontology, but only in a fragment, for instance if one wants to reuse content from a larger ontology for a more specialized application, or to analyse what the ontology states about a given set of names. In particular, this set of names would form a *signature* Σ, got which we want to compute an ontology $\mathcal{M}$ that captures all the logical entailments of the original ontology $\mathcal{O}$ expressed using only the names in Σ. Our aim is to compute such an $\mathcal{M}$ that is as simple as possible, in terms of number and size of axioms. This problem has received a lot of attention in the past years, and for an $\mathcal{M}$ satisfying those requirements, the common approaches are *modules* and *uniform interpolants*.

Modules are *syntactical subsets* of the ontology $\mathcal{O}$ that preserve entailments within a given signature. There is a variety of notions of modules and properties they can satisfy that have been investigated in the literature [Grau *et al.*, 2008; Konev *et al.*, 2009]. *Semantic modules* preserve all models of the ontology modulo the given signature Σ. This makes them undecidable already for light-weight DLs such as $\mathcal{EL}$ [Konev *et al.*, 2013], which is why existing methods often compute approximations of minimal modules [Gatens *et al.*, 2014; Romero *et al.*, 2016]. A popular example are *locality-based modules*, which can be computed in a very short time [Grau *et al.*, 2008]. However, locality-based modules can be comparatively large, even if the provided signature is small [Chen *et al.*, 2014]. In contrast to semantic modules, *deductive modules* are decidable, and focus only on entailments in Σ that can be expressed in the DL under consideration. Practical methods to compute them are presented in [Koopmann and Chen, 2020; Yang *et al.*, 2023b]. However, while those modules is often half the size of the locality-based modules, for the more expressive DL $\mathcal{ALC}$, those methods are time-consuming, and can also not always guarantee minimality. An approximation of modules are *excerpts*, whose size is bounded by the user, but which may not preserve all entailments in the given signature [Chen *et al.*, 2017].

Since modules are always subsets of the original ontology, they may use names outside of the given signature. *Uniform interpolants* (UIs), on the other hand, *only use names from the provided signature* Σ, and are thus usually not syntactical subsets of the input ontology. This makes them useful also for applications other than ontology reuse, such as for logical difference [Ludwig and Konev, 2014], abduction [Del-Pinto and Schmidt, 2019], information hiding [Grau, 2010], and proof generation [Alrabbaa *et al.*, 2020]. The strict requirement on the signature means that UIs may not al-

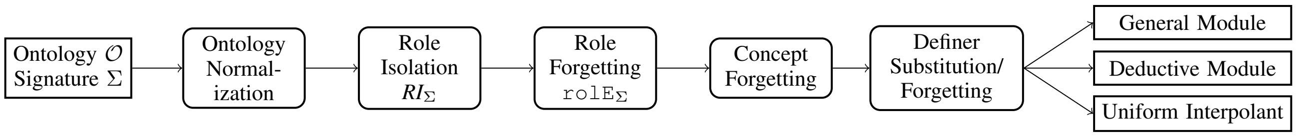

Figure 1: Overview of our unified method for computing general modules, deductive modules, and uniform interpolants

ways exist, and, in case of $\mathcal{ALC}$, can be of a size that is triple exponential in the size of the input [Lutz and Wolter, 2011]. Despite this high complexity, practical implementations for computing UIs exist [Zhao and Schmidt, 2018; Koopmann, 2020]. However, their computation times are much higher than for module extraction and can produce very complex axioms.

Both modules and UIs can be more complex than necessary. By dropping the syntactical requirements of those notions—being subsets of $\mathcal{O}$ and being within Σ respectively—we may obtain ontologies that are both smaller and simpler, and yet still preserve all relevant entailments, which would make them better suited for ontology reuse and analysis. In this paper, we present a method to compute such *general modules*, which are indeed often smaller and nicer than the corresponding classical modules and UIs. Our method can indeed also compute deductive modules and UIs, and does so in significantly shorter time than the state-of-the-art. While general modules have been investigated for the lightweight DLs $\mathcal{EL}$ and $\mathcal{ELH}$ [Nikitina and Glimm, 2012; Alghamdi *et al.*, 2021], to our knowledge, this is the first time they are investigated for $\mathcal{ALC}$.

The main steps of our approach are shown in Figure 1. Our method essentially works by performing uniform interpolation on a normalized version of the input ontology (Section 3). During normalization, so-called *definer names* are introduced, which are eliminated in the final step.This idea is inspired by the method for uniform interpolation in [Koopmann, 2020]. However, different from this approach, we put fewer constraints on the normal form and do not allow the introduction of definers after normalization, which changes the mechanism of uniform interpolation. As a result, our definer elimination step may reintroduce names eliminated during uniform interpolation, which is not a problem for the computation of general modules. In contrast, eliminating definers as done in [Koopmann, 2020] can cause an exponential blowup, and introduce concepts with the non-standard *greatest fixpoint* constructor [Calvanese and De Giacomo, 2003].

A particular challenge in uniform interpolation is the elimination of role names, for which existing approaches either rely on expensive calls to an external reasoner [Zhao *et al.*, 2019; Koopmann, 2020] or avoid the problem partially by introducing universal roles [Zhao and Schmidt, 2017; Koopmann and Chen, 2020], leading to results outside $\mathcal{ALC}$. A major contribution of this paper is a technique called *role isolation*, which allows to eliminate roles more efficiently, and explains our short computation times (Section 4).

Our evaluation shows that all our methods, including the one for uniform interpolation, can compete with the run times of locality-based module extraction, while at the same time resulting in subtantially smaller ontologies (Section 7).

Our main contributions are: 1) the first method dedicated to general modules in $\mathcal{ALC}$, 2) a formal analysis of some properties of the general modules we compute, 3) new methods for module extraction and uniform interpolation that significantly improve the state-of-the-art, 4) an evaluation on real-world ontologies indicating the efficiency of our technique.

For detailed proofs of the results, please refer to the extended version of the paper [Yang *et al.*, 2023a]. A prototype of our method can be found at https://hub.docker.com/r/yh1997/demo_gemo.

2 Preliminaries

We recall the DL $\mathcal{ALC}$ [Baader *et al.*, 2017]. Let $\mathsf{N_C}=\{A, B, \ldots\}$ and $\mathsf{N_R} = \{r, s, \ldots\}$ be pair-wise disjoint, countably infinite sets of *concept* and *role names*, respectively. A *signature* $\Sigma \subseteq \mathsf{N_C} \cup \mathsf{N_R}$ is a set of concept and role names. *Concepts* C are built according to the following grammar rules.

$$C ::= \top \mid A \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \exists r.C \mid \forall r.C \qquad (1)$$

For simplicity, we identify concepts of the form $\neg\neg C$ with C. In this paper, an *ontology* $\mathcal{O}$ is a finite set of *axioms* of the form $C \sqsubseteq D$, C and D being concepts. We denote by $sig(\mathcal{O})$/$sig(C)$ the set of concept and role names occurring in $\mathcal{O}$/C, and we use $sig_{\mathsf{C}}(*)$/$sig_{\mathsf{R}}(*)$ to refer to the concept/role names in $sig(*)$. For a signature Σ, a Σ-*axiom* is an axiom α s.t. $sig(\alpha) \subseteq \Sigma$.

An *interpretation* $\mathcal{I}=(\Delta^\mathcal{I}, \cdot^\mathcal{I})$ consists of a non-empty set $\Delta^\mathcal{I}$ and a function $\cdot^\mathcal{I}$ mapping each $A \in \mathsf{N}_C$ to $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$ and each $r \in \mathsf{N}_R$ to $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$. The interpretation function $\cdot^\mathcal{I}$ is extended to concepts as follows:

$$\top^\mathcal{I} = \Delta^\mathcal{I}, \quad (\neg C)^\mathcal{I} = \Delta^\mathcal{I} \setminus C^\mathcal{I},$$
$$(C \sqcap D)^\mathcal{I} = C^\mathcal{I} \cap D^\mathcal{I}, \quad (C \sqcup D)^\mathcal{I} = C^\mathcal{I} \cup D^\mathcal{I},$$
$$(\exists r.C)^\mathcal{I} = \left\{a \in \Delta^\mathcal{I} \mid \exists b \in C^\mathcal{I} : (a, b) \in r^\mathcal{I}\right\},$$
$$(\forall r.C)^\mathcal{I} = \left\{a \in \Delta^\mathcal{I} \mid \forall b : (a, b) \in r^\mathcal{I} \to b \in C^\mathcal{I}\right\}.$$

An axiom $C \sqsubseteq D$ is *satisfied* by an interpretation $\mathcal{I}$ ($\mathcal{I} \models C \sqsubseteq D$) if $C^\mathcal{I} \subseteq D^\mathcal{I}$. $\mathcal{I}$ is a *model* of an ontology $\mathcal{O}$ ($\mathcal{I} \models \mathcal{O}$) if $\mathcal{I}$ satisfies every axiom in $\mathcal{O}$. $\mathcal{O}$ *entails* an axiom α if $\mathcal{I} \models \alpha$ for every model $\mathcal{I}$ of $\mathcal{O}$. If α holds in every interpretation, we write $\models \alpha$ and call α an *tautology*.

The *length* $|*|$ *of concepts and axioms* is defined inductively by $|\top| = |A| = 1$, where $A \in \mathsf{N_C}$, $|C \sqcup D| = |C \sqcap D| = |C \sqsubseteq D| = |C| + |D|$, $|\forall r.C| = |\exists r.C| = |C| + 1$, and $|\neg C| = |C|$. Then, the *length of an ontology*, denoted $\|\mathcal{O}\|$, is defined by $\|\mathcal{O}\| = \sum_{\alpha \in \mathcal{O}} |\alpha|$.

A central notion for us is (deductive) inseparability [Konev *et al.*, 2009; Koopmann and Chen, 2020]. Given two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ and a signature Σ, $\mathcal{O}_1$ and $\mathcal{O}_2$ are Σ-*inseparable*, in symbols $\mathcal{O}_1 \equiv_\Sigma \mathcal{O}_2$, if for every Σ-axiom α,

$\mathcal{O}$:	$A_1 \sqsubseteq \exists r.\exists s.B_1 \sqcup \exists r.B_2$	$B_1 \sqcap B_3 \sqsubseteq \bot$	$A_2 \sqsubseteq A_3 \sqcup \forall s.B_3$	$B_4 \sqsubseteq A_4$	$B_2 \sqsubseteq B_4$
$cl(\mathcal{O})$:	$\neg A_1 \sqcup \exists r.D_1 \sqcup \exists r.D_3$	$\neg D_1 \sqcup \exists s.D_2$	$\neg D_2 \sqcup B_1$	$\neg D_3 \sqcup B_2$	$\neg A_2 \sqcup A_3 \sqcup \forall s.D_4$
	$\neg D_4 \sqcup B_3$	$\neg B_1 \sqcup \neg B_3$	$\neg B_2 \sqcup B_4$	$\neg B_4 \sqcup A_4$	
$RI_\Sigma(\mathcal{O})$:	$\neg A_1 \sqcup \exists r.D_1 \sqcup \exists r.D_3$	$\neg D_1 \sqcup \exists s.D_2$	$\neg D_3 \sqcup B_2$	$A_2 \sqcup A_3 \sqcup \forall s.D_4$	$\neg B_1 \sqcup \neg B_3$
	$\neg B_2 \sqcup B_4$	$\neg B_4 \sqcup A_4$	$\neg D_2 \sqcup \neg D_4$		
$\texttt{rolE}_\Sigma(RI_\Sigma(\mathcal{O}))$:	$\neg A_1 \sqcup \exists r.D_1 \sqcup \exists r.D_3$	$\neg D_3 \sqcup B_2$	$\neg B_1 \sqcup \neg B_3$	$\neg B_2 \sqcup B_4$	$\neg B_4 \sqcup A_4$
	$\neg D_2 \sqcup \neg D_4$	$\neg D_1 \sqcup \neg A_2 \sqcup A_3$			
$\texttt{conE}_\Sigma(\texttt{rolE}_\Sigma(RI_\Sigma(\mathcal{O})))$:	$\neg A_1 \sqcup \exists r.D_1 \sqcup \exists r.D_3$	$\neg D_1 \sqcup \neg A_2 \sqcup A_3$	$\neg D_3 \sqcup A_4$		
$gm_\Sigma(\mathcal{O})$:	$A_1 \sqsubseteq \exists r.\exists s.B_1 \sqcup \exists r.B_2$	$A_2 \sqcap \exists s.B_1 \sqsubseteq A_3$	$B_2 \sqsubseteq A_4$		
$gm^*_\Sigma(\mathcal{O})$:	$A_1 \sqsubseteq \exists r.(\neg A_2 \sqcup A_3) \sqcup \exists r.A_4$				

Table 1: Ontologies generated throughout the running example.

$\mathcal{O}_1 \models \alpha$ iff $\mathcal{O}_2 \models \alpha$. In this paper, we are concerned with the computation of general modules, defined in the following.

Definition 1 (General module)**.** *Given an ontology $\mathcal{O}$ and a signature Σ, an ontology $\mathcal{M}$ is a* general module *for $\mathcal{O}$ and Σ iff (i) $\mathcal{O} \equiv_\Sigma \mathcal{M}$ and (ii) $\mathcal{O} \models \mathcal{M}$.*

Every ontology is always a general module of itself, but we are interested in computing ones that are small in length and low in complexity. Two extreme cases of general modules are uniform interpolants and deductive modules.

Definition 2 (Uniform interpolant & deductive module)**.** *Let $\mathcal{O}$ be an ontology, Σ a signature, and $\mathcal{M}$ a general module for $\mathcal{O}$ and Σ. Then, (i) $\mathcal{M}$ is a* uniform interpolant *for $\mathcal{O}$ and Σ if $sig(\mathcal{M}) \subseteq \Sigma$, and (ii) $\mathcal{M}$ is a* deductive module *for $\mathcal{O}$ and Σ if $\mathcal{M} \subseteq \mathcal{O}$.*

3 Ontology Normalization

Our method performs forgetting on a normalized view of the ontology, which is obtained via the introduction of fresh names as in [Koopmann, 2015]. An ontology $\mathcal{O}$ is in *normal form* if every axiom is of the following form:

$$\top \sqsubseteq L_1 \sqcup \ldots \sqcup L_n \qquad L_i ::= A \mid \neg A \mid \mathsf{Q}r.A,$$

where $A \in \mathsf{N}_C$, and $\mathsf{Q} \in \{\forall, \exists\}$. We call the disjuncts L_i *literals*. For simplicity, we omit the "$\top \sqsubseteq$" on the left-hand side of normalized axioms, which are regarded as *sets*, in order to avoid dealing with duplicated literals and order. As an example, the axiom $A_2 \sqsubseteq A_3 \sqcup \forall s.B_3$ is equivalent to $\neg A_2 \sqcup A_3 \sqcup \forall s.B_3$ in normal form.

We assume a function *cl* that normalizes $\mathcal{O}$ usind standard transformations (see for example [Koopmann, 2015]). In particular, *cl* replaces concepts C occurring under role restrictions $\mathsf{Q}r.C$ by fresh names D taken from a set $\mathsf{N}_\mathsf{D} \subseteq \mathsf{N}_\mathsf{C} \setminus sig_\mathsf{C}(\mathcal{O})$ of *definers*. We use D, D', D_1, D_2, ... to denote definers. For each introduced definer D, we remember the concept C_D that was replaced by it. We assume that distinct occurrences of the same concept are replaced by distinct definers. Thus, in the resulting normalization of $\mathcal{O}$ denoted $cl(\mathcal{O})$, every literal $\mathsf{Q}r.D$ satisfies $D \in \mathsf{N}_\mathsf{D}$, and for every $D \in \mathsf{N}_\mathsf{D}$, $cl(\mathcal{O})$ contains at most one literal of the form $\mathsf{Q}r.D$. Every definer D has to occur only in literals of the form $\neg D$ or $\mathsf{Q}r.D$, that is, positive literals of the form D are not allowed. Obviously, we require $cl(\mathcal{O}) \equiv_{sig(\mathcal{O})} \mathcal{O}$.

Example 1. *Let $\mathcal{O}$ be the ontology defined in the first row of Table 1. By normalizing $\mathcal{O}$, we obtain the set $cl(\mathcal{O})$ shown in the second row of Table 1. The definers D_1, D_2 and D_3 in $cl(\mathcal{O})$ replace the concepts $C_{D_1} = \exists s.B_1$, $C_{D_2} = B_1$ and $C_{D_3} = B_2$, respectively.*

For a fixed normalization, we define a partial order $\preceq_d$ over all introduced definers, which is defined as the smallest reflexive-transitive relation over N_D s.t.

- $D' \preceq_d D$ if $\neg D \sqcup C \in cl(\mathcal{O})$ and $D' \in sig(C)$.

Intuitively, $D' \preceq_d D$ whenever $C_{D'}$ is contained in C_D. In Example 1, we have $D_2 \preceq_d D_1$, since $\neg D_1 \sqcup \exists s.D_2 \in cl(\mathcal{O})$. Our normalization ensures that $\preceq_d$ is acyclic.

In the following, we assume that the ontology $\mathcal{O}$ and the signature Σ do not contain definers, unless stated otherwise.

4 Role Forgetting

An ontology $\mathcal{M}$ is called a *role forgetting* for $\mathcal{O}$ and Σ iff $\mathcal{M}$ is a uniform interpolant for $\mathcal{O}$ and $\Sigma' = \Sigma \cup sig_\mathsf{C}(\mathcal{O})$. Existing methods to compute role forgetting either rely on an external reasoner [Zhao *et al.*, 2019; Koopmann, 2020] or use the *universal role* ∇ [Zhao and Schmidt, 2017; Koopmann and Chen, 2020]. The former approach can be expensive, while the latter produces axioms outside of $\mathcal{ALC}$. The normalization allows us to implement a more efficient solution within $\mathcal{ALC}$, which relies on an integrated reasoning procedure and an additional transformation step that produces so-called *role isolated ontologies*.

4.1 Role Isolated Ontologies

The main idea is to separate names $A \in \mathsf{N}_\mathsf{C}$ that occur with roles outside of the signature, using the following notations.

$$Rol(A, \mathcal{O}) = \{r \in sig(\mathcal{O}) \mid \mathsf{Q}r.A \text{ appears in } \mathcal{O},\ \mathsf{Q} \in \{\forall, \exists\}\}$$
$$Out_\Sigma(\mathcal{O}) = \{A \in sig(\mathcal{O}) \mid Rol(A, \mathcal{O}) \not\subseteq \Sigma\}$$

Definition 3 (Role-isolated ontology)**.** *An ontology $\mathcal{O}$ is* role isolated *for Σ if (i) $\mathcal{O}$ is in normal form, and (ii) every axiom $\alpha \in \mathcal{O}$ is of one of the following forms:*

(c1) $L_1 \sqcup \ldots \sqcup L_n$, $L_i := \neg A$ *with* $A \in Out_\Sigma(\mathcal{O})$ *for all* i;

(c2) $L_1 \sqcup \ldots \sqcup L_m$, $L_i := \mathsf{Q}r.A \mid B \mid \neg B$ *with* $r, A \in sig(\mathcal{O})$, $B \notin Out_\Sigma(\mathcal{O})$ *for all* i.

Thus, an axiom in a *role isolated* ontology falls into two disjoint categories: either (c1) it contains literals built only over concepts in $Out_\Sigma(\mathcal{O})$ or (c2) it contains role restrictions or literals built over concepts outside $Out_\Sigma(\mathcal{O})$.

$$\underline{A\text{-}Rule}: \quad \frac{C_1 \sqcup A_1 \qquad \neg A_1 \sqcup C_2}{C_1 \sqcup C_2}$$

$$\underline{r\text{-}Rule}: \quad \frac{C_1 \sqcup \exists r.D_1,\ \bigcup_{j=2}^{n}\{C_j \sqcup \forall r.D_j\},\ K_D}{C_1 \sqcup \ldots \sqcup C_n},$$

where $K_D = \neg D_1 \sqcup \ldots \sqcup \neg D_n$ or $\neg D_2 \sqcup \ldots \sqcup \neg D_n$.

Figure 2: Inference rules for computing $\mathcal{D}_\Sigma(\mathcal{O})$

Example 2 (Example 1 cont'd)**.** *For $\Sigma = \{r, A_1, A_2, A_3, A_4\}$, we have $Out_\Sigma(cl(\mathcal{O})) = \{D_2, D_4\}$. $cl(\mathcal{O})$ is not role isolated for Σ because of $\neg D_2 \sqcup B_1$.*

Given an ontology, we compute its role isolated form using the following definition.

Definition 4. *The* role isolated form $RI_\Sigma(\mathcal{O})$ *of $\mathcal{O}$ is defined as $RI_\Sigma(\mathcal{O}) := cl_\Sigma(\mathcal{O}) \cup \mathcal{D}_\Sigma(\mathcal{O})$, where*

- *$cl_\Sigma(\mathcal{O}) \subseteq cl(\mathcal{O})$ contains all $\alpha \in cl(\mathcal{O})$ s.t. if $\neg D$ is a literal of α, then $Rol(D', cl(\mathcal{O})) \subseteq \Sigma$ for all $D' \in \mathsf{N_D}$ s.t. $D \preceq_d D'$.*
- *$\mathcal{D}_\Sigma(\mathcal{O})$ is the set of axioms $\neg D_1 \sqcup \ldots \sqcup \neg D_n$ s.t. **(i)** $cl_\Sigma(\mathcal{O})$ contains axioms of the form $C_1 \sqcup \mathsf{Q}_1 r.D_1$, $C_2 \sqcup \forall r.D_2$, ..., $C_n \sqcup \forall r.D_n$, where $r \in \mathsf{N_R} \setminus \Sigma$, $\mathsf{Q}_1 \in \{\forall, \exists\}$, and **(ii)** $\{D_1 \ldots, D_n\}$ is a minimal set of definers s.t. $cl(\mathcal{O}) \models D_1 \sqcap \ldots \sqcap D_n \sqsubseteq \bot$.*

Intuitively, if a definer D appears in $cl_\Sigma(\mathcal{O})$, then it should not depend on definers in $Out_\Sigma(\mathcal{O})$.

Example 3 (Example 2 cont'd)**.** *We have:*

- *$cl_\Sigma(\mathcal{O}) = cl(\mathcal{O}) \setminus \{\neg D_2 \sqcup B_1, \neg D_4 \sqcup B_3\}$ because $Rol(D_2, cl(\mathcal{O})) = Rol(D_4, cl(\mathcal{O})) = \{s\} \not\subseteq \Sigma$ and,*
- *$\mathcal{D}_\Sigma(\mathcal{O}) = \{\neg D_2 \sqcup \neg D_4\}$.*

Theorem 1. *$RI_\Sigma(\mathcal{O})$ is role isolated for Σ and we have $\mathcal{O} \equiv_{\Sigma \cup sig_{\mathsf{C}}(\mathcal{O})} RI_\Sigma(\mathcal{O})$.*

To compute $\mathcal{D}_\Sigma(\mathcal{O})$, we saturate $cl(\mathcal{O})$ using the inference rules shown in Fig. 2, which is sufficient due to the following lemma.

Lemma 1. *Let $\mathcal{S}$ be the set of axioms $\neg D_1 \sqcup \ldots \sqcup \neg D_n$, obtained by applying the rules in Fig. 2 exhaustively on $cl(\mathcal{O})$. Then, for all $D_1, \ldots, D_n \in \mathsf{N_D}$, we have $cl(\mathcal{O}) \models D_1 \sqcap \ldots \sqcap D_n \sqsubseteq \bot$ iff $\neg D_{i_1} \sqcup \ldots \sqcup \neg D_{i_k} \in \mathcal{S}$ for some subset $\{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$.*

Example 4 (Example 3 cont'd)**.** *The axiom $\neg D_2 \sqcup \neg D_4$ in $\mathcal{D}_\Sigma(\mathcal{O})$ is obtained by applying two A-Rule inferences:*

$$\frac{\dfrac{\neg D_2 \sqcup B_1, \quad \neg B_1 \sqcup \neg B_3}{\neg D_2 \sqcup \neg B_3,} \qquad \neg D_4 \sqcup B_3}{\neg D_2 \sqcup \neg D_4}$$

4.2 Role Forgetting for Role Isolated Ontologies

If $\mathcal{O}$ is role isolated for Σ, a role forgetting for $\mathcal{O}$ and Σ can be obtained using the *r-Rule* in Figure 2. Our method applies to any ontology in normal form, not necessarily normalized using *cl*, which is why now the concept names $D_1, \ldots, D_n$ in the r-Rule can include also concept names outside $\mathsf{N_D}$.

Definition 5. *$\texttt{rolE}_\Sigma(\mathcal{O})$ is the ontology obtained as follows:*

1. *apply the* r-Rule *exhaustively for each $r \in sig_{\mathsf{R}}(\mathcal{O}) \setminus \Sigma$,*
2. *remove all axioms containing some $r \in sig_{\mathsf{R}}(\mathcal{O}) \setminus \Sigma$.*

The second step ensures that all role names in the resulting ontology $\texttt{rolE}_\Sigma(\mathcal{O})$ are in Σ and therefore, we have $sig(\texttt{rolE}_\Sigma(\mathcal{O})) \subseteq \Sigma \cup sig_{\mathsf{C}}(\mathcal{O})$.

Example 5 (Example 4 cont'd)**.** *For the ontology $RI_\Sigma(\mathcal{O})$, Table 1 (fourth row) shows $\texttt{rolE}_\Sigma(RI_\Sigma(\mathcal{O}))$ which is obtained through the following two steps:*

1. *The new axiom $\neg D_1 \sqcup \neg A_2 \sqcup A_3$ is generated by the r-Rule inference:*
$$\frac{\neg D_1 \sqcup \exists s.D_2,\ \neg A_2 \sqcup A_3 \sqcup \forall s.D_4,\ \neg D_2 \sqcup \neg D_4}{\neg D_1 \sqcup \neg A_2 \sqcup A_3}$$
2. *The two axioms $\neg D_1 \sqcup \exists s.D_2$, $\neg A_2 \sqcup A_3 \sqcup \forall s.D_4$ are removed because they contain $s \in sig(RI_\Sigma(\mathcal{O})) \setminus \Sigma$.*

Theorem 2. *If $\mathcal{O}$ is role isolated for Σ, then $\texttt{rolE}_\Sigma(\mathcal{O})$ is a role-forgetting for $\mathcal{O}$ and Σ.*

5 Computing General Modules via $\texttt{rolE}_\Sigma$

We compute a general module from $\texttt{rolE}_\Sigma(\mathcal{O})$ by forgetting also the concept names and eliminating all definers. The latter is necessary to obtain an ontology entailed by $\mathcal{O}$. Forgetting concept names is done to further simplify the ontology.

5.1 Concept Forgetting

We say that an ontology $\mathcal{M}$ is a *concept forgetting* for $\mathcal{O}$ and Σ iff $\mathcal{M}$ is a uniform interpolant for $\mathcal{O}$ and the signature $\Sigma' = \Sigma \cup sig_{\mathsf{R}}(\mathcal{O}) \cup \mathsf{N_D}$. A concept forgetting can be computed through the *A-Rule* in Figure 2.

Definition 6. *$\texttt{conE}_\Sigma(\mathcal{O})$ is the ontology obtained as follows:*

1. *apply the* A-Rule *exhaustively for each $A \in sig_{\mathsf{C}}(\mathcal{O}) \setminus \Sigma$,*
2. *delete every axiom α that contains A or $\neg A$, where $A \in \mathsf{N_C} \setminus \Sigma$ and no axiom contains $\mathsf{Q}r.A$ for $\mathsf{Q} \in \{\forall, \exists\}$ and $r \in \mathsf{N_R}$.*

Example 6 (Example 5 cont'd)**.** *Table 1 (the 5th row) shows the axioms in $\texttt{conE}_\Sigma(\texttt{rolE}_\Sigma(RI_\Sigma(\mathcal{O})))$ obtained as follows.*

1. *$\neg D_3 \sqcup B_4$, $\neg B_2 \sqcup A_4$, and $\neg D_3 \sqcup A_4$ are first generated by applying the A-Rule on B_2 and B_4.*
2. *Axioms containing B_i or $\neg B_i$, $i \in \{1, \ldots, 4\}$, are removed since $B_i \notin \Sigma$. $\neg D_2 \sqcup \neg D_4$ is also removed because there are no literals of the form $\mathsf{Q}r.D_2$ or $\mathsf{Q}r.D_4$.*

The following is a consequence of [Zhao and Schmidt, 2017, Theorem 1].

Theorem 3. *If $\mathcal{O}$ is in normal form, then $\texttt{conE}_\Sigma(\mathcal{O})$ is a concept forgetting for $\mathcal{O}$ and Σ.*

Theorems 1, 2 and 3, give us the following corollary.

Corollary 1. $\texttt{conE}_\Sigma(\texttt{rolE}_\Sigma(RI_\Sigma(\mathcal{O}))) \equiv_\Sigma \mathcal{O}$.

5.2 Constructing the General Module

Now, in order to obtain our general modules, we have to eliminate the definers from $\texttt{conE}_\Sigma(\texttt{rolE}_\Sigma(\mathit{RI}_\Sigma(\mathcal{O})))$. To improve the results, we delete *subsumed axioms* (i.e., axioms $\top \sqsubseteq C \sqcup D$ for which we also derived $\top \sqsubseteq C$) and also simplify the axioms.

Theorem 4. *Let $gm_\Sigma(\mathcal{O})$ be the ontology obtained from $\texttt{conE}_\Sigma(\texttt{rolE}_\Sigma(\mathit{RI}_\Sigma(\mathcal{O})))$ by*

- *deleting subsumed axioms,*
- *replacing each definer D by C_D, and*
- *exhaustively applying $C_1 \sqsubseteq \neg C_2 \sqcup C_3 \Rightarrow C_1 \sqcap C_2 \sqsubseteq C_3$ and $C_1 \sqsubseteq \mathsf{Q}r.\neg C_2 \sqcap C_3 \Rightarrow C_1 \sqcap \overline{\mathsf{Q}}r.C_2 \sqsubseteq C_3$, where $\overline{\exists} = \forall$ and $\overline{\forall} = \exists$.*

Then, $gm_\Sigma(\mathcal{O})$ is a general module for $\mathcal{O}$ and Σ.

Example 7 (Example 6 cont'd)**.** *Table 1 (the 8th row) shows the general module $gm_\Sigma(\mathcal{O})$, which has been obtained using $C_{D_1}{=}\exists s.B_1$ and $C_{D_3}{=}B_2$.*

Eliminating definers in this way may reintroduce previously forgotten names, which is why our general modules are in general not uniform interpolants. This has the advantage of avoiding the triple exponential blow-up caused by uniform interpolation (see Section 1). In contrast, the size of our result is at most single exponential in the size of the input.

Proposition 1. *For any ontology $\mathcal{O}$ and signature Σ, we have $\|gm_\Sigma(\mathcal{O})\| \leq 2^{O(\|cl(\mathcal{O})\|)}$. On the other hand, there exists a family of ontologies $\mathcal{O}_n$ and signatures Σ_n s.t. $\|\mathcal{O}_n\|$ is polynomial in $n \geq 1$ and $\|gm_{\Sigma_n}(\mathcal{O}_n)\| = n \cdot 2^{O(\|cl(\mathcal{O}_n)\|)}$.*

We will see in Section 7 that this theoretical bound is usually not reached in practice, and usually general modules are much smaller than the input ontology.

For some module extraction methods, such as for locality-based modules [Grau *et al.*, 2008], iterating the computation can lead to smaller modules. The following result shows that this is never the case for our method.

Proposition 2. *Let $(\mathcal{M}_i)_{i\geq 1}$ be the sequence of ontologies defined by (i) $\mathcal{M}_1 = gm_\Sigma(\mathcal{O})$ and (ii) $\mathcal{M}_{i+1} = gm_\Sigma(\mathcal{M}_i)$ for $i \geq 1$. Then, we have $\mathcal{M}_i \subseteq \mathcal{M}_{i+1}$ for $i \geq 1$. Moreover, there exists $i_0 \geq 0$ s.t. $\mathcal{M}^k = \mathcal{M}^{i_0}$ for all $k \geq i_0$.*

This property holds thanks to the substitution step of Theorem 4. This step may reintroduce in $gm_\Sigma(\mathcal{O})$ concept and role names outside of Σ. As a result, the repeated application of $\texttt{rolE}_\Sigma$ and $\texttt{conE}_\Sigma$ on $gm_\Sigma(\mathcal{O})$ can produce additional but unnecessary axioms. However, for ontologies in normal form, our method is *stable* in the sense that repeated applications produce the same ontology.

Proposition 3. *Let $\mathcal{O}$ be an ontology in normal form and $\mathcal{M} = gm_\Sigma(\mathcal{O})$. Then, $gm_\Sigma(\mathcal{M}) = \mathcal{M}$.*

5.3 Optimizing the Result

The general module $gm_\Sigma(\mathcal{O})$ may contain complex axioms since definers D can stand for complex concepts C_D. To make the result more concise, we eliminate some definers before substituting them. In particular, we use the following operations on $\texttt{conE}_\Sigma(\texttt{rolE}_\Sigma(\mathit{RI}_\Sigma(\mathcal{O})))$, inspired by [Sakr and Schmidt, 2022].

<u>*conD-Elim*</u>:

$$\frac{C_1 \sqcup \mathsf{Q}r.D_1, \bigcup_{j=2}^{n}\{C_j \sqcup \forall r.D_j\}, \neg D_1 \sqcup \ldots \sqcup \neg D_n}{C_1 \sqcup \ldots \sqcup C_n \sqcup \mathsf{Q}r.\bot}$$

<u>*D-Prop*</u> :

$$\frac{C_1 \sqcup \mathsf{Q}r.D, \quad \bigcup_{j=2}^{n}\{\neg D \sqcup C_j\}}{C_1 \sqcup \mathsf{Q}r.(C_2 \sqcap \ldots \sqcap C_n)}$$

where $\bigcup_{j=2}^{n}\{\neg D \sqcup C_j\}$ ($n \geq 1$) are all the axioms of the form $\neg D \sqcup C$. This rule is applicable only if no C_j contains definers.

Figure 3: Rules to eliminate definers

Op1. **Eliminating conjunctions of definers** aims to eliminate disjunctions of negative definers ($\neg D_1 \sqcup \ldots \sqcup \neg D_n$). This is done in two steps: (i) Applying the *conD-Elim* rule in Figure 3 on $\texttt{conE}_\Sigma(\texttt{rolE}_\Sigma(\mathit{RI}_\Sigma(\mathcal{O})))$, and then (ii) deleting all axioms of the form $\neg D_1 \sqcup \ldots \sqcup \neg D_n$.

Op2. **Eliminating single definers** aims to get rid of definers D that do not occur in axioms of the form $\neg D \sqcup \neg D_1 \sqcup C$. This is done in two steps: (i) applying the *D-Prop* rule of Figure 3 exhaustively and then (ii) deleting all axioms containing definers for which *D-Prop* has been applied.

Theorem 5. *Let $gm^*_\Sigma(\mathcal{O})$ be the ontology obtained by:*

- *successive application of Op1 and Op2 over $\texttt{conE}_\Sigma(\texttt{rolE}_\Sigma(\mathit{RI}_\Sigma(\mathcal{O})))$, followed by*
- *application of the steps described in Theorem 4.*

Then, $gm^*_\Sigma(\mathcal{O})$ is a general module for $\mathcal{O}$ and Σ.

Example 8. *Assume $\Sigma = \{r, A, A_1\}$ and*

$$\mathcal{O} = \{A \sqsubseteq \forall r.\exists s.B_1,\ A_1 \sqsubseteq \forall r.\forall s.B_2,\ B_1 \sqcap B_2 \sqsubseteq \bot\}.$$

Then, $\texttt{conE}_\Sigma(\texttt{rolE}_\Sigma(\mathit{RI}_\Sigma(\mathcal{O})))$ is:

$$\{\neg A \sqcup \forall r.D_1,\ \neg A_1 \sqcup \forall r.D_2,\ \neg D_1 \sqcup \neg D_2\},$$

where $C_{D_1} = \exists s.B_1$, $C_{D_2} = \forall s.B_2$. And thus, by replacing D_i by C_{D_i}, we obtain $gm_\Sigma(\mathcal{O}) =$

$$\{A \sqsubseteq \forall r.\exists s.B_1,\ A_1 \sqsubseteq \forall r.\forall s.B_2,\ \exists s.B_1 \sqcap \forall s.B_2 \sqsubseteq \bot\},$$

which is actually more intricate than $\mathcal{O}$. We can avoid this by applying the two optimizations described above.

The elimination of definer conjunctions (Op1) produces

$$\{\neg A \sqcup \forall r.D_1, \neg A_1 \sqcup \forall r.D_2, \neg A \sqcup \neg A_1 \sqcup \forall r.\bot\}. \quad (2)$$

(i) The first step of Op1 applies the conD-Elim inference:

$$\frac{\neg A \sqcup \forall r.D_1, \neg A_1 \sqcup \forall r.D_2, \neg D_1 \sqcup \neg D_2}{\neg A \sqcup \neg A_1 \sqcup \forall r.\bot}.$$

(ii) The second step of Op1 removes the axiom $\neg D_1 \sqcup \neg D_2$.

Then, the elimination of definers (Op2) produces

$$\{\neg A \sqcup \forall r.\top, \neg A_1 \sqcup \forall r.\top, \neg A \sqcup \neg A_1 \sqcup \forall r.\bot\} \quad (3)$$

by replacing D_1, D_2 by $\top$ as there is no axioms with negative $D_i, i = 1, 2$ in Equation (2). Note that the first two axioms in Equation (3) are tautologies and thus can be ignored.

Finally, we have $gm^*_\Sigma(\mathcal{O}) = \{A \sqcap A_1 \sqsubseteq \forall r.\bot\}$.

6 Deductive Modules and Uniform Interpolants

Deductive modules Depending on the situation, users might prefer the axioms in the original ontology $\mathcal{O}$ rather than newly introduced axioms (e.g., axioms in $gm_\Sigma(\mathcal{O})$ or $gm^*_\Sigma(\mathcal{O})$). For such situations, we can compute a deductive module for $\mathcal{O}$ and Σ by tracing back the inferences performed when computing the general module $gm^*_\Sigma(\mathcal{O})$.

Let $Res_\Sigma(\mathcal{O})$ be the set of all axioms generated by the computation progress of $gm^*_\Sigma(\mathcal{O})$. Clearly, $gm^*_\Sigma(\mathcal{O}) \subseteq Res_\Sigma(\mathcal{O})$. We iteratively construct a relation R on $Res_\Sigma(\mathcal{O})$ during the computation $gm^*_\Sigma(\mathcal{O})$ as follows: we start with $R = \emptyset$, and each time a new axiom β is generated from a premise set $\{\alpha_1, \ldots, \alpha_n\}$ (e.g., if β is obtained by applying r-Rule on $\{\alpha_1, \ldots, \alpha_n\}$), we add to R the relations $\alpha_1 R \beta$, ..., $\alpha_n R \beta$. Let R^* be the smallest transitive closure of R. Then the deductive module is defined as follows.

Theorem 6. *Let us define $dm_\Sigma(\mathcal{O})$ by*

$$dm_\Sigma(\mathcal{O}) = \{\alpha \in \mathcal{O} \mid \alpha R^* \beta \text{ for some } \beta \in gm^*_\Sigma(\mathcal{O})\}.$$

Then, $dm_\Sigma(\mathcal{O})$ is a deductive module for $\mathcal{O}$ and Σ.

Uniform interpolants While general modules can be a good alternative to uniform interpolants for ontology reuse, uniform interpolation has applications that require the ontology to be fully in the selected signature, as stated in the introduction. If instead of substituting definers D by C_D, we eliminate them using existing uniform interpolation tools, we can compute a uniform interpolant for the input.

When computing $gm^*_\Sigma(\mathcal{O})$ for an ontology $\mathcal{O}$ and signature Σ, if all definers have been eliminated by Op1 and Op2 from Section 5.3 (as in Example 8), then $gm^*_\Sigma(\mathcal{O})$ is indeed a uniform interpolant for $\mathcal{O}$ and Σ. Otherwise, we compute a uniform interpolant by forgetting the remaining definers using an existing uniform interpolation tool such as LETHE or FAME [Koopmann, 2020; Zhao and Schmidt, 2018]. As we see in Section 7, this allows us to compute uniform interpolants much faster than using the tool alone.

7 Evaluation

To show that our general modules can serve as a better alternative for ontology reuse and analysis, we compared them with the state-of-the-art tools implementing module extraction and uniform interpolation for $\mathcal{ALC}$. We were also interested in the impact of our optimization, and the performance of our technique for computing deductive modules and uniform interpolants. We implemented a prototype called GEMO in Python 3.7.4. As evaluation metrics, we looked at run time, length of computed ontologies, and length of largest axiom in the result. All the experiments were performed on a machine with an Intel Xeon Silver 4112 2.6GHz, 64 GiB of RAM, Ubuntu 18.04, and OpenJDK 11.

Corpus The ontologies used in our experiment are generated from the OWL Reasoner Evaluation (ORE) 2015 classification track [Parsia *et al.*, 2017] by the two following steps. First, we removed axioms outside of $\mathcal{ALC}$ from each ontology in ORE 2015. Then, we kept the ontologies $\mathcal{O}$ for which $cl(\mathcal{O})$ contained between 100 and 100,000 names. This resulted in 222 ontologies.

$\top\bot^*$-module	minM	LETHE	FAME	GEMO	gmLethe
100%	84.34%	85.27%	91.25%	97.34%	96.17%

Table 2: Success rate evaluation. The first (resp. second) best-performing method is highlighted in red (resp. blue).

Signatures For each ontology, we generated 50 signatures consisting of 100 concept and role names. As in [Koopmann and Chen, 2020], we selected each concept/role name with a probability proportional to their occurrence frequency in the ontology. In the following, a *request* is a pair consisting of an ontology and a signature.

Methods For each request $(\mathcal{O}, \Sigma)$, GEMO produced three different (general) modules $gm_\Sigma(\mathcal{O})$, $gm^*_\Sigma(\mathcal{O})$ and $dm_\Sigma(\mathcal{O})$, respectively denoted by gm, gm*, and dm. gmLethe denotes the uniform interpolation method described in Section 6, where we used GEMO for computing gm* and then LETHE for definer forgetting. In the implementation, for each request, we first extracted a locality based $\top\bot^*$-*module* [Grau *et al.*, 2008] to accelerate the computation. This is a common practice also followed by the uniform interpolation and deductive module extraction tools used in our evaluation. Since removing subsumed axioms as mentioned in Theorems 4 and 5 can be challenging, we set a time limit of 10s for this task.

We compared our methods with four different alternatives: **(i)** $\top\bot^*$-*modules* [Grau *et al.*, 2008] as implemented in the OWL API [Horridge and Bechhofer, 2011]; **(ii)** minM [Koopmann and Chen, 2020] that computes *minimal deductive modules* under $\mathcal{ALCH}^\nabla$-semantics; **(iii)** LETHE 0.6[1][Koopmann, 2020] and FAME 1.0[2] [Zhao and Schmidt, 2018] that compute uniform interpolants.

Success rate We say a method *succeeds* on a request if it outputs the expected results within 600s. Table 2 summarizes the success rate for the methods considered. After the $\top\bot^*$-modules, our method GEMO had the highest success rate.

Module length and run time Because some of the methods can change the shape of axioms, the number of axioms is not a good metric for understanding the quality of general modules. We thus chose to use ontology length as defined in Section 2, rather than size, for our evaluation. Table 3 shows the length and run time for the requests on which all methods were successful (78.45% of all requests).

We observe that dm and gmLethe have the best overall performance: their results had a substantially smaller average length and were computed significantly faster than others. Note that the average size of results for dm was even smaller than that for minM. The reason is that minM preserves entailments over $\mathcal{ALCH}^\nabla$, while we preserve only entailments over $\mathcal{ALC}$. Therefore, the minM results may contain additional axioms compared to the $\mathcal{ALC}$ deductive modules.

Comparing gm and gm* regarding length lets us conclude that the optimization in Section 5.3 is effective. On the other hand, minM produced results of small length but at the cost of long computation times. FAME and $\top\bot^*$-module were

[1] https://lat.inf.tu-dresden.de/~koopmann/LETHE/
[2] http://www.cs.man.ac.uk/~schmidt/sf-fame/

Methods		Resulting ontology length	Time cost
minM		2,355 / 392.59 / 264	595.88 / 51.82 / 8.86
⊤⊥*-module		4,008 / 510.77 / 364	5.94 / 1.03 / 0.90
FAME		9,446,325 / 6,661.01 / 271	526.28 / 3.20 / 1.17
LETHE		131,886 / 609.30 / 196	598.20 / 49.21 / 13.57
GEMO	gm	179,999 / 2,335.05 / 195	
	gm*	21,891 / 466.15 / 166	17.50 / 2.44 / 1.63
	dm	2,789 / 366.36 / 249	
gmLethe		21,891 / 364.10 /162	513.15 / 3.08 / 1.68

Table 3: Comparison of different methods (max. / avg. / med.).

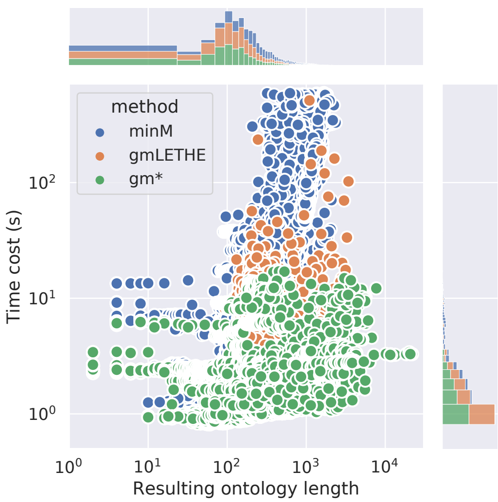

Figure 4: Comparison of minM, gmLETHE, and gm*.

quite time-efficient but less satisfactory in size, especially for FAME, whose results are often considerably larger than for the other methods. LETHE took more time than FAME, but produced more concise uniform interpolants on average.

For 87.87% of the requests reported in Table 3, gm* already computed a uniform interpolant, so that gmLethe did not need to perform any additional computations.

Figure 4 provides a detailed comparison of minM, gm* and gmLethe. gm* was often faster but produced larger results. In contrast, gmLethe produced more concise results at the cost of longer computation time. While minM avoided large modules, it was generally much slower than our methods.

Table 4 summarizes the results concerning all requests for which GEMO (resp. gmLethe) was successful. We see that the results of dm had a small average size. However, as for gm* and gmLethe, the median size of results was much smaller, which suggests that gm* and gmLethe perform better over relatively simple cases.

Uniform interpolants For 80.23% of requests where GEMO was successful, $gm^*_\Sigma(\mathcal{O})$ was already a uniform interpolants. In the other cases, the success rate for gmLethe was 93.96%. In the cases where LETHE failed, the success rate for gmLethe was 36.23%.

The comparison of LETHE with gmLethe in Figure 5 shows that gmLethe was significantly faster than LETHE in most of the cases.

Methods		Resulting ontology length	Time cost
GEMO	gm	17,335,040 / 35,008.2 / 310	
	gm*	2,318,878 / 2,978.77 / 214	585.97 / 4.89 / 1.75
	dm	18,218 / 638.74 / 309	
gmLethe		353,107 / 1,006.34 /192	579.70 / 7.56 / 2.02

Table 4: GEMO and gmLethe: Summary of results for all their own successful experiments (max. / avg. / med.).

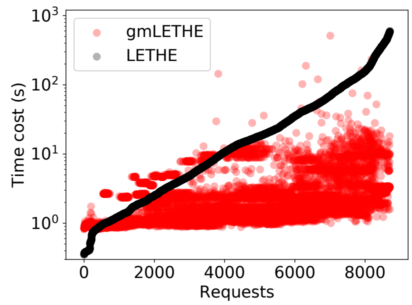

Figure 5: Run time comparison of LETHE and gmLethe.

Axiom size A potential shortcoming of general modules compared to classical modules is that they could contain axioms that are more complex than those of the input, and thus be harder to handle by human end-users. For the requests reported in Table 3, the largest axiom in the output of minM had length 352, while for gm*, it had length 5,815, and for gm, even only 56. In contrast, for the uniform interpolants computed by LETHE and FAME, the situation was much worse: here, the largest axiom had a length of 26,840 and 130,700, respectively, which is clearly beyond what can be understood by a human end-user. Besides these extreme cases, we can also observe differences wrt. the median values: for gm* the longest axiom had a median length of 3, which is even lower than the corresponding value for minM (5), and, as expected, lower than for LETHE (4) and FAME (6). This indicates that, in most cases, general modules computed by gm* are simple than for the other tools.

8 Conclusion

We presented new methods for computing general modules for $\mathcal{ALC}$ ontologies, which can also be used for computing deductive modules and uniform interpolants. Due to its higher syntactical flexibility, our general modules are often smaller and less complex than both classical modules and uniform interpolants computed with the state-of-the-art, which makes them particularly useful for applications such as ontology reuse and ontology analysis. Our method is based on a new role isolation process that enables efficient role forgetting and an easy definer elimination. The experiments on real-world ontologies validate the efficiency of our proposal and the quality of the computed general modules. In the future, we want to optimize the concept elimination step to obtain more concise general modules. Also, we would like to investigate how to generalize our ideas to more expressive DLs.

Acknowledgements

Hui Yang, Yue Ma and Nicole Bidoit are funded by the BPI-France (PSPC AIDA: 2019-PSPC-09) and ANR (EXPIDA: ANR-22-CE23-0017). Patrick Koopmann is partially funded by DFG Grant 389792660 as part of TRR 248—CPEC (see https://perspicuous-computing.science)

References

[Alghamdi *et al.*, 2021] Ghadah Alghamdi, Renate A. Schmidt, Warren Del-Pinto, and Yongsheng Gao. Upwardly abstracted definition-based subontologies. In Anna Lisa Gentile and Rafael Gonçalves, editors, *K-CAP '21: Knowledge Capture Conference*, pages 209–216. ACM, 2021.

[Alrabbaa *et al.*, 2020] Christian Alrabbaa, Franz Baader, Stefan Borgwardt, Patrick Koopmann, and Alisa Kovtunova. Finding small proofs for description logic entailments: Theory and practice. In Elvira Albert and Laura Kovács, editors, *LPAR 2020: 23rd International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, volume 73 of *EPiC Series in Computing*, pages 32–67. EasyChair, 2020.

[Baader *et al.*, 2017] Franz Baader, Ian Horrocks, Carsten Lutz, and Ulrike Sattler. *An Introduction to Description Logic*. Cambridge University Press, 2017.

[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. 2003.

[Chen *et al.*, 2014] Jieying Chen, Michel Ludwig, Yue Ma, and Dirk Walther. Evaluation of extraction techniques for ontology excerpts. In *Description Logics*, volume 1193 of *CEUR Workshop Proceedings*, pages 471–482, 2014.

[Chen *et al.*, 2017] Jieying Chen, Michel Ludwig, Yue Ma, and Dirk Walther. Zooming in on ontologies: Minimal modules and best excerpts. In *16th International Semantic Web Conference, Proceedings, Part I*, pages 173–189. Springer, 2017.

[Del-Pinto and Schmidt, 2019] Warren Del-Pinto and Renate A. Schmidt. ABox abduction via forgetting in $\mathcal{ALC}$. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI*, pages 2768–2775. AAAI Press, 2019.

[Donnelly and others, 2006] Kevin Donnelly et al. SNOMED-CT: The advanced terminology and coding system for eHealth. *Studies in health technology and informatics*, 121:279, 2006.

[Gatens *et al.*, 2014] William Gatens, Boris Konev, and Frank Wolter. Lower and upper approximations for depleting modules of description logic ontologies. In Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan, editors, *ECAI 2014 - 21st European Conference on Artificial Intelligence*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 345–350. IOS Press, 2014.

[Grau *et al.*, 2008] B Cuenca Grau, Ian Horrocks, Yevgeny Kazakov, and Ulrike Sattler. Modular reuse of ontologies: Theory and practice. *Journal of Artificial Intelligence Research*, 31:273–318, 2008.

[Grau, 2010] Bernardo Cuenca Grau. Privacy in ontology-based information systems: A pending matter. *Semantic Web*, 1(1-2):137–141, 2010.

[Horridge and Bechhofer, 2011] Matthew Horridge and Sean Bechhofer. The OWL API: a java API for OWL ontologies. *Semantic Web*, 2(1):11–21, 2011.

[Konev *et al.*, 2009] Boris Konev, Carsten Lutz, Dirk Walther, and Frank Wolter. Formal properties of modularisation. In Heiner Stuckenschmidt, Christine Parent, and Stefano Spaccapietra, editors, *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, volume 5445 of *Lecture Notes in Computer Science*, pages 25–66. Springer, 2009.

[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.

[Koopmann and Chen, 2020] Patrick Koopmann and Jieying Chen. Deductive module extraction for expressive description logics. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 1636–1643, 2020.

[Koopmann, 2015] Patrick Koopmann. *Practical uniform interpolation for expressive description logics*. PhD thesis, University of Manchester, UK, 2015.

[Koopmann, 2020] Patrick Koopmann. LETHE: Forgetting and uniform interpolation for expressive description logics. *KI—Künstliche Intelligenz*, 34(3):381–387, 2020.

[Ludwig and Konev, 2014] Michel Ludwig and Boris Konev. Practical uniform interpolation and forgetting for $\mathcal{ALC}$ tboxes with applications to logical difference. In Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014*. AAAI Press, 2014.

[Lutz and Wolter, 2011] Carsten Lutz and Frank Wolter. Foundations for uniform interpolation and forgetting in expressive description logics. In Toby Walsh, editor, *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 989–995. IJCAI/AAAI, 2011.

[Nikitina and Glimm, 2012] Nadeschda Nikitina and Birte Glimm. Hitting the sweetspot: Economic rewriting of knowledge bases. In Philippe Cudré-Mauroux, Jeff Heflin, Evren Sirin, Tania Tudorache, Jérôme Euzenat, Manfred Hauswirth, Josiane Xavier Parreira, Jim Hendler, Guus Schreiber, Abraham Bernstein, and Eva Blomqvist, editors, *The Semantic Web - ISWC 2012 - 11th International Semantic Web Conference, Proceedings, Part I*, volume 7649 of *Lecture Notes in Computer Science*, pages 394–409. Springer, 2012.

[Noy *et al.*, 2009] Natalya Fridman Noy, Nigam H. Shah, Patricia L. Whetzel, Benjamin Dai, Michael Dorf, Nicholas Griffith, Clément Jonquet, Daniel L. Rubin, Margaret-Anne D. Storey, Christopher G. Chute, and Mark A. Musen. BioPortal: ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Res.*, 37(Web-Server-Issue):170–173, 2009.

[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. Reason.*, 59(4):455–482, 2017.

[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.

[Sakr and Schmidt, 2022] Mostafa Sakr and Renate A. Schmidt. Fine-grained forgetting for the description logic $\mathcal{ALC}$. In Ofer Arieli, Martin Homola, Jean Christoph Jung, and Marie-Laure Mugnier, editors, *Proceedings of the 35th International Workshop on Description Logics (DL 2022)*, volume 3263 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2022.

[Yang *et al.*, 2023a] Hui Yang, Patrick Koopmann, Yue Ma, and Nicole Bidoit. Efficient computation of general modules for $\mathcal{ALC}$ ontologies (extended version), 2023. https://arxiv.org/abs/2305.09503.

[Yang *et al.*, 2023b] Hui Yang, Yue Ma, and Nicole Bidoit. Efficient extraction of $\mathcal{EL}$-ontology deductive modules. In *The 37th AAAI Conference on Artificial Intelligence, AAAI*. in press, 2023.

[Zhao and Schmidt, 2017] Yizheng Zhao and Renate A. Schmidt. Role forgetting for $\mathcal{ALCOQH}(\nabla)$-ontologies using an Ackermann-based approach. In Carles Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017*, pages 1354–1361. ijcai.org, 2017.

[Zhao and Schmidt, 2018] Yizheng Zhao and Renate A Schmidt. FAME: an automated tool for semantic forgetting in expressive description logics. In *International Joint Conference on Automated Reasoning*, pages 19–27. Springer, 2018.

[Zhao *et al.*, 2019] Yizheng Zhao, Ghadah Alghamdi, Renate A. Schmidt, Hao Feng, Giorgos Stoilos, Damir Juric, and Mohammad Khodadadi. Tracking logical difference in large-scale ontologies: A forgetting-based approach. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019*, pages 3116–3124. AAAI Press, 2019.