Semantic Forgetting in Expressive Description Logics

Mostafa Sakr[0000−0003−4296−5831] and Renate A. Schmidt[0000−0002−6673−3333]

University of Manchester, UK
{Mostafa.Sakr, Renate.Schmidt}@manchester.ac.uk

Abstract. Forgetting is an important ontology extraction tool. We present a semantic forgetting method for $\mathcal{ALC}$ ontologies, which captures the semantic content, possibly, by introducing helper concept symbols. Evaluation shows that the method performs well on large-scale ontologies.

1 Introduction

Important ontologies are often large-scale, formalizing several related topics of a certain domain. For example, the SNOMED CT ontology [37] contains more than 340 000 axioms modelled over 361 000 concept and role names, and formalizes several topics from the medical and biomedical domains such as diseases, symptoms, medicines, procedures, and organisms. Working with such large-scale ontologies is a challenge due to their sheer size, while in reality, only small extracts are needed. This presses for ontology extraction methods such as *forgetting* [8–12, 18, 19, 29].

The aim of forgetting is extracting an ontology, hereafter called a forgetting view, that is *inseparable* from the original ontology in terms of the captured knowledge relative to some *keep vocabulary* (the unforgotten subset of the vocabulary of the original ontology) [7, 19, 20, 30]. To this end, it uses reasoning to eliminate, or forget, the unwanted vocabulary and capture the knowledge relative to the keep vocabulary, possibly by introducing new axioms.

Two different languages $\mathcal{L}^V$ and $\mathcal{L}^C$ should be differentiated when considering forgetting. These are, respectively, the language of the extracted ontology, and the language of the captured consequences. Different forms of forgetting, that have been introduced in the literature, vary on their choices of $\mathcal{L}^V$ and $\mathcal{L}^C$ [7, 20]. A variant of forgetting of particular interest in this paper is *semantic forgetting* [26, 39–41] in which the meaning of the keep vocabulary is preserved in the semantic forgetting view. That is, for every model of the original ontology there is model of the semantic forgetting view and vice versa, such that both models interpret the keep vocabulary in the same way. It has been shown that in this case both ontologies agree on all second-order consequences [7, 19]. That is, $\mathcal{L}^C$ is the language of all second-order consequences of the original ontology relative to the keep signature.

Despite its high precision, semantic forgetting is not sufficiently studied in the literature. In the context of first-order logic (FOL), predicate symbols can be

forgotten by quantifying over them [13, 26, 27]. Since elimination of second-order quantifiers is not always successful [1, 14], the generated semantic forgetting view of a first-order theory is not always representable in first-order logic but is always representable in second-order logic. For the $\mathcal{ALC}$ fragment of FOL, examples can be constructed to show that the semantic forgetting view of an $\mathcal{ALC}$ ontology cannot be represented as a set of $\mathcal{ALC}$ axioms formulated over the keep vocabulary [22]. Attempts were made to formulate the semantic forgetting view of $\mathcal{ALC}$ ontologies using the description logic $\mathcal{ALCOQ}^{\neg,\sqcup,\sqcap}$ [45] which extends $\mathcal{ALC}$ with nominals ($\mathcal{O}$), qualified number restrictions ($\mathcal{Q}$), role negation ($\neg$), role disjunction ($\sqcup$), and role conjunction ($\sqcap$). However, even for acyclic ontologies, $\mathcal{ALCOQ}^{\neg,\sqcup,\sqcap}$ is not expressive enough to formulate the complete semantic forgetting view of $\mathcal{ALC}$ ontologies over the keep vocabulary. At present, there are no complete semantic forgetting methods for $\mathcal{ALC}$ ontologies.

Taking ideas from [26], we present a complete semantic forgetting method for the description logic $\mathcal{ALC}$. The method forgets concept names from $\mathcal{ALC}$ ontologies. As in [26], we have the perspective that forgetting symbols are second-order predicate symbols, and use fresh helper symbols to capture the complete semantic forgetting view. However, we use helper symbols differently. Whereas [26] replaces the forgetting symbols with fresh helper symbols, our proposed method uses them to represent the role fillers in which the forgetting symbols occur. The idea of our method is to eliminate the forgetting symbols and as many of the introduced helper symbols as possible. Eliminate the introduced helper symbols completely is however not possible in some cases. Thus, in general, the language $\mathcal{L}^V$ in which the semantic forgetting view is expressed is the language of all $\mathcal{ALC}$-axioms formulated over the keep vocabulary augmented with a set of fresh helper concept symbols. Using helper symbols in this way gives advantages over [26] in that our method is directly applicable to description logic syntax and is better tailored towards the purpose of ontology extraction. An advantage of our method over other methods in the literature [13, 15, 21–23, 28, 42–46] is that the semantic forgetting view preserves the form of the input ontology, thus making it more readable and convenient for ontology development, and debugging.

While the use of helper symbols is debatable because they are second-order, because they are existentially quantified, and because only $\mathcal{ALC}$ constructs are used to formulate the axioms of the semantic forgetting view, most standard reasoning tasks, such as satisfiability checking, classification, and query answering, can be performed on the semantic forgetting view using the standard $\mathcal{ALC}$ tools. Additionally, reduction in the complexity of these reasoning tasks is obtained when the number of the helper symbols is less than the number of symbols being forgotten. Helper symbols are increasingly being used in different applications to increase the expressive power of the used language [17, 21].

Complexity of forgetting is studied in [29] where it is shown that the forgetting view of an $\mathcal{ALC}$ ontology is in the worst case triple exponential in the size of the input ontology. We show that using our method the size of the semantic forgetting view is single exponential in the size of the input ontology and

double exponential in the number of forgetting symbols. This reduction in the theoretical size complexity can be attributed to the use of helper symbols.

2 Getting Started

The vocabulary of an ontology consists of concept and role names. Assume they belong respectively to two disjoint sets N_c and N_r of concept names and role names. $\mathcal{ALC}$ concepts are defined inductively by the grammar: $C, D := \top \mid \bot \mid A \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \exists r.C \mid \forall r.C$ where $A \in N_c$ and $r \in N_r$.

An ontology, or TBox, is a set of axioms of the forms $C \sqsubseteq D$ and $C \equiv D$, where C, D are concepts. Equivalence axiom $C \equiv D$ is short-hand for the two subsumption axioms $C \sqsubseteq D$ and $D \sqsubseteq C$.

Concepts can be referred to using their *position* inside the axiom [34]. A *position* is a word over natural numbers. Let $\alpha = C \odot D$ be an $\mathcal{ALC}$ axiom where $\odot \in \{\sqsubseteq, \equiv\}$, then the positions of C and D relative to α are 1 and 2 respectively, in symbols: $\alpha|1 = C$ and $\alpha|2 = D$. Consider a general $\mathcal{ALC}$ concept ψ. Let $pos(\psi)$ be the set of positions of the sub-concepts occurring in ψ. A sub-concept ϕ of ψ is referred to by $\alpha|i.\pi$ where i is position of ψ relative to the axiom α and $\pi \in pos(\psi)$ is the position of ϕ relative to ψ. The definition of $pos(\psi)$ is given inductively as follows:

1. $i.\pi \in pos(\psi)$ if $\psi = \phi_1 \odot \phi_2 \odot \cdots \odot \phi_n$ where $\odot \in \{\sqcup, \sqcap\}$, $\pi \in pos(\phi_i)$ and $1 \leq i \leq n$.
2. $1.\pi \in pos(\psi)$ if ψ takes any of the forms: $\neg\phi, \exists r.\phi$, or $\forall r.\phi$ and $\pi \in pos(\phi)$.

Example 1. Let α be the axiom $A \sqcap \exists r.B \equiv \neg D \sqcap (C_1 \sqcup \forall r.\forall r.C_2)$. The following table lists the subconcepts of α and their positions relative to α.

Concept	Position	Concept	Position
$A \sqcap \exists r.B$	$\alpha\|1$	$\neg D \sqcap (C_1 \sqcup \forall r.\forall r.C_2)$	$\alpha\|2$
A	$\alpha\|1.1$	$\exists r.B$	$\alpha\|1.2$
$\neg D$	$\alpha\|2.1$	$C_1 \sqcup \forall r.\forall r.C_2$	$\alpha\|2.2$
B	$\alpha\|1.2.1$	D	$\alpha\|2.1.1$
C_1	$\alpha\|2.2.1$	$\forall r.\forall r.C_2$	$\alpha\|2.2.2$
$\forall r.C_2$	$\alpha\|2.2.2.1$	C_2	$\alpha\|2.2.2.1.1$

We write $\alpha[\iota/\psi]$ to denote the axiom generated by replacing the sub-concept ϕ at position ι relative to α with the concept ψ.

The *polarity* of a concept occurring at position ι in a subsumption axiom α is denoted by $pol(\alpha, \iota)$, where $pol(\alpha, 1) = -1$, $pol(\alpha, 2) = 1$, and:

1. $pol(\alpha, i.\pi) = pol(\alpha, i)$ if $\alpha|i$ is a conjunction, disjunction, or a concept of the forms $\exists r.C$ or $\forall r.C$.
2. $pol(\alpha, i.\pi) = -pol(\alpha, i)$ if $\alpha|i$ is a concept of the form $\neg C$.

A concept is *positive* in an axiom if it occurs with polarity 1, and *negative* if it occurs with polarity -1.

Example 2. Consider the axiom α from Example 1. The concepts $\{A \sqcap \exists r.B, A, \exists r.B, B, D\}$ are negative, whereas the concepts $\{\neg D \sqcap (C_1 \sqcup \forall r.\forall r.C_2), \neg D, C_1 \sqcup \forall r.\forall r.C_2, C_1, \forall r.\forall r.C_2, \forall r.C_2, C_2\}$ are positive.

An interpretation $\mathcal{I}$ in $\mathcal{ALC}$ is a pair $\langle \Delta^{\mathcal{I}}, .^{\mathcal{I}} \rangle$ where the domain $\Delta^{\mathcal{I}}$ is a nonempty set, and $.^{\mathcal{I}}$ is an interpretation function that assigns to each concept symbol $A \in N_c$ a subset of $\Delta^{\mathcal{I}}$, and to each $r \in N_r$ a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. $\mathcal{ALC}$ concepts and axioms can be interpreted by translation to FOL [3, 16]. First, we interpret concept and role names respectively as unary and binary predicates in FOL. Then, we define two translation functions π_x and π_y that inductively translate $\mathcal{ALC}$ concepts to FOL formulae with the free variable x and y, respectively:

$$\begin{aligned}
&\pi_x(\top) = \pi_y(\top) = \text{TRUE} && \pi_x(\bot) = \pi_y(\bot) = \text{FALSE}\\
&\pi_x(A) = A(x) && \pi_y(A) = A(y)\\
&\pi_x(\neg C) = \neg \pi_x(C) && \pi_y(\neg C) = \neg \pi_y(C)\\
&\pi_x(C \sqcap D) = \pi_x(C) \wedge \pi_x(D) && \pi_y(C \sqcap D) = \pi_y(C) \wedge \pi_y(D)\\
&\pi_x(C \sqcup D) = \pi_x(C) \vee \pi_x(D) && \pi_y(C \sqcup D) = \pi_y(C) \vee \pi_y(D)\\
&\pi_x(\exists r.C) = \exists y\ (r(x,y) \wedge \pi_y(C)) && \pi_y(\exists r.C) = \exists x\ (r(y,x) \wedge \pi_x(C))\\
&\pi_x(\forall r.C) = \forall y\ (r(x,y) \rightarrow \pi_y(C)) && \pi_y(\forall r.C) = \forall x\ (r(y,x) \rightarrow \pi_x(C))
\end{aligned}$$

Let C and D be any $\mathcal{ALC}$ concepts. Define the translation function π which translates axioms to first-order formulae with one free variable.

$$\pi(C \equiv D) = (\pi_x(C) \leftrightarrow \pi_x(D)) \qquad \pi(C \sqsubseteq D) = (\pi_x(C) \rightarrow \pi_x(D))$$

An ontology $\mathcal{O}$, or a set of axioms, is translated as:

$$\pi(\mathcal{O}) = \bigwedge_{C \equiv D \in \mathcal{O}} \forall x\ \pi(C \equiv D) \wedge \bigwedge_{C \sqsubseteq D \in \mathcal{O}} \forall x\ \pi(C \sqsubseteq D) \tag{1}$$

In the presented forgetting method we use FOL formulas on the form (1). To simplify the notation, we allow the conjunction (disjunction) of axioms to mean the conjunction (disjunction) of their FOL translations:

$$\begin{aligned}
&(C_1 \sqsubseteq D_1) \wedge (C_2 \sqsubseteq D_2) && \text{means} && \pi(C_1 \sqsubseteq D_1) \wedge \pi(C_2 \sqsubseteq D_2)\\
&(C_1 \sqsubseteq D_1) \vee (C_2 \sqsubseteq D_2) && \text{means} && \pi(C_1 \sqsubseteq D_1) \vee \pi(C_2 \sqsubseteq D_2)
\end{aligned}$$

Let $\mathcal{O}$ be an ontology, and α an axiom. We say α is satisfiable with respect to $\mathcal{O}$ if there is a model $\mathcal{I}$ of $\mathcal{O}$ such that $\mathcal{I} \models \alpha$. We say α is a consequence of $\mathcal{O}$, in symbols, $\mathcal{O} \models \alpha$, if for every model $\mathcal{I}$ of $\mathcal{O}$ it holds that $\mathcal{I} \models \alpha$.

Let C be an $\mathcal{ALC}$ concept, we denote by $sig(C)$ the set of concept and role names appearing in C. For an ontology $\mathcal{O}$, $sig(\mathcal{O})$ is the set of concept and role names appearing in its axioms. That is, $sig(\mathcal{O}) = \bigcup_{C \odot D \in \mathcal{O}} sig(C) \cup sig(D)$, where $\odot \in \{\sqsubseteq, \equiv\}$. The *size* of $\mathcal{O}$, in symbols, size($\mathcal{O}$), is the number of axioms occurring on $\mathcal{O}$.

Definition 1. *Let $\mathcal{F}$ be a set of concept names, and $\mathcal{I}$ and $\mathcal{J}$ two models such that $\Delta^{\mathcal{I}} = \Delta^{\mathcal{J}}$. We write $\mathcal{I} \sim_{\mathcal{F}} \mathcal{J}$ if and only if $p^{\mathcal{I}} = p^{\mathcal{J}}$ for every concept and role name except, possibly, for $p \in \mathcal{F}$.*

Definition 2. *Let $\mathcal{O}$ be an $\mathcal{ALC}$ ontology and $\mathcal{F} \subseteq sig(\mathcal{O}) \cap N_c$ be the forgetting signature. We say that the $\mathcal{ALC}$ ontology $\mathcal{V}$ is a* semantic forgetting view *of $\mathcal{O}$ w.r.t. $\mathcal{F}$ iff the following holds,*

1. *$sig(\mathcal{V}) \subseteq sig(\mathcal{O}) \backslash \mathcal{F}$;*
2. *for every model $\mathcal{I}$ of $\mathcal{O}$, there is a model $\mathcal{J}$ of $\mathcal{V}$, and vice versa, such that $\mathcal{I} \sim_{\mathcal{F}} \mathcal{J}$.*

The keep vocabulary *is the set of concept and role symbols in $sig(\mathcal{O}) \backslash \mathcal{F}$.*

The following Lemma is useful in some proofs:

Lemma 1. *Let $\mathcal{I}, \mathcal{J}$ and $\mathcal{M}$ be three models such that $\Delta^{\mathcal{I}} = \Delta^{\mathcal{J}} = \Delta^{\mathcal{M}}$. Let S_1 and S_2 be two sets of concept names, then: $\mathcal{I} \sim_{S_1} \mathcal{J}$ and $\mathcal{J} \sim_{S_2} \mathcal{M}$ implies that $\mathcal{I} \sim_{S_1 \cup S_2} \mathcal{M}$.*

The following example suggests that the semantic view of an $\mathcal{ALC}$ ontology is not in general representable in $\mathcal{ALC}$.

Example 3. Let $\mathcal{O} = \{A \sqsubseteq \exists r.B \sqcap \exists r.\neg B\}$ and $\mathcal{F} = \{B\}$. In FOL with *Skolem* functions, the semantic forgetting view $\mathcal{V}$ of $\mathcal{O}$ with respect to $\mathcal{F}$ is:

$$\forall x \forall y((\neg A(x) \vee r(x, f(x))) \wedge (\neg A(y) \vee r(y, g(y))) \wedge (\neg A(x) \vee \neg A(y) \vee f(x) \not\approx g(y))) \quad (2)$$

This was computed with the second-order quantifer elimination method in [13]. f and g are function symbols introduced by *Skolemisation* [4, 13] to represent existential quantification. Since (2) cannot be back-translated to FOL without function symbols, representing (2) in the $\mathcal{ALC}$ fragment is not possible [6].

Even the more expressive description logic $\mathcal{ALCOQ}^{\neg,\sqcup,\sqcap}$ is insufficient to capture the semantic view [45]. The following example illustrates this observation.

Example 4. Consider $\mathcal{O}$ and $\mathcal{F}$ from Example 3. The forgetting view generated by the methods [22, 45] is $\mathcal{V} = \{A \sqsubseteq \geq 2r.\top\}$, which models the information that every element in A has at least two different successors via r. Let $\mathcal{I}$ be a model with domain of interpretation $\Delta = \{a_1, a_2, a_3, b_1, b_2, b_3\}$; and:

$A^{\mathcal{I}} = \{a_1, a_2, a_3\}$ and, $r^{\mathcal{I}} = \{(a_1, b_1), (a_1, b_2), (a_2, b_1), (a_2, b_3), (a_3, b_2), (a_3, b_3)\}$.

$\mathcal{I}$ is a model of $\mathcal{V}$ but there is no extension of $\mathcal{I}$ that is a model of $\mathcal{O}$ because no interpretation of B can separate $\{b_1, b_2, b_3\}$ into two disjoint sets where every element in A is connected to at least one element from each set.

The proposed method captures the semantic view by relaxing the first condition of Definition 2 and allowing fresh helper concept symbols to be used in the semantic forgetting view. Fresh here means that the introduced helper symbols do not occur in the vocabulary of the original ontology. The following example shows the idea.

Example 5. Consider $\mathcal{O}$ and $\mathcal{F}$ from Example 3. The ontology $\mathcal{V} = \{A \sqsubseteq \exists r.D_1 \sqcap \exists r.D_2, D_1 \sqcap D_2 \sqsubseteq \bot\}$ is a semantic forgetting view of $\mathcal{O}$ with respect to $\mathcal{F}$ where D_1 and D_2 are fresh concept symbols.

In the presented method, we use helper symbols to represent role fillers. In Example 5, D_1 and D_2 represent two disjoint subsets of the set $\{y \in \Delta^{\mathcal{I}} | (x, y) \in r^{\mathcal{I}} \wedge x \in A^{\mathcal{I}}\}$ where $\mathcal{I}$ is a model of $\mathcal{V}$. Precise interpretations of D_1 and D_2 cannot be given without knowledge of the forgotten symbol B. So, helper symbols can be seen as second-order existentially quantified symbols.

When ontology extraction is required to replace the original ontology with a smaller one in reasoning applications, these applications are unperturbed if the extracted ontology contains helper symbols in its vocabulary. This is at least correct for standard reasoning applications such as *classification*, *satisfiability checking* and *query answering*. The reasons that the original ontology can be safely replaced by the semantic forgetting view which uses helper symbols are:

1. The introduced helper symbols are implicitly existentially quantified. Thus, existing inference systems that operate on the original ontology are sufficient. That is, inference systems which allow for universally quantified symbols are not needed.
2. Using helper symbols, the semantic forgetting view can be formulated using $\mathcal{ALC}$ syntax. Thus, $\mathcal{ALC}$ reasoning methods can operate directly on the semantic forgetting view.

Let us explain this using a query answering problem. In query answering problems, forgetting can be used to replace the original ontology with a simpler forgetting view such that the vocabulary of the asked queries is subset of the vocabulary of the forgetting view [20].

Example 6. Consider the ontology $\mathcal{O} = \{A_1 \sqsubseteq \forall r.B, A_2 \sqsubseteq \forall r.\neg B\}$, and $\mathcal{V} = \{A_1 \sqsubseteq \forall r.D_1, A_2 \sqsubseteq \forall r.D_2, D_1 \sqcap D_2 \sqsubseteq \bot\}$ which is the semantic forgetting view with respect to $\{B\}$ where D_1 and D_2 are helper symbols. Both ontologies models the information that the r-successors of the elements in the interpretation of A_1 are disjoint from the r-successors of the elements in the interpretation of A_2. Suppose $\mathcal{O}$ is used in a boolean conjunctive query task $(\mathcal{O}, \mathcal{A}) \models r(a_2, b)$ where $\mathcal{A} = \{A_1(a_1), A_2(a_2), r(a_1, b)\}$ is a database, and $(\mathcal{O}, \mathcal{A})$ is the knowledge-base consisting of $\mathcal{O}$ and $\mathcal{A}$. An answer to this query is *yes* if $(\mathcal{O}, \mathcal{A}) \models r(a_2, b)$, and *no* if $(\mathcal{O}, \mathcal{A}) \not\models r(a_2, b)$. Evidently, the answer to this query is *no*, which can be computed by observing that $(\mathcal{O}, \mathcal{A}) \not\models r(a_2, b)$ iff $(\mathcal{O}, \mathcal{A}), \neg r(a_2, b) \not\models \bot$. Replacing $\mathcal{O}$ with $\mathcal{V}$, i.e., answering $(\mathcal{V}, \mathcal{A}) \models r(a_2, b)$, yields the same result, which also can be computed using the satisfiability checking method used above. In this computation, full interpretations of the helper symbols are not required.

3 Forgetting Method

We now present a method to compute the semantic forgetting view of $\mathcal{ALC}$ ontologies. The method uses helper symbols to capture the semantic forgetting

view. It proceeds in two stages. The first stage iteratively eliminates *defined* forgetting symbols. A concept symbol B is *defined* if an axiom of the form $B \equiv C$ exists, and $B \notin sig(C)$. Elimination of B is performed by replacing B everywhere in the ontology with C. We call this, *definition expansion*.

Example 7. Let $\mathcal{O} = \{A \sqsubseteq B, B \equiv C\}$ and $\mathcal{F} = \{B\}$. The semantic view of $\mathcal{O}$ with respect to $\mathcal{F}$ is $\mathcal{V} = \{A \sqsubseteq C\}$ obtained by replacing B by C in $\mathcal{O}$.

The second stage of the method eliminates the remaining forgetting symbols. Forgetting is defined by an adaptation of the following formula:

$$Forget(\mathcal{O}, B) = \mathcal{O}_B^{\top} \vee \mathcal{O}_B^{\bot} \tag{3}$$

often attributed to [5]. $\mathcal{O}_B^{\top}(\mathcal{O}_B^{\bot})$ denotes the result of replacing B by $\top(\bot)$ everywhere in $\mathcal{O}$. Applied iteratively, (3) provides a method to forget propositional variables from propositional theories [25]. It does not, however, extend to description logics [26, 11, 10, 9]. We integrate (3) with *structural transformation* [34] to perform forgetting for $\mathcal{ALC}$ ontologies.

Structural transformation extracts forgetting symbols appearing under role restriction and exposes them to resolution. Before applying structural transformation we replace every equivalence axiom $C \equiv D$ that contains a forgetting symbol by the two subsumption axioms $C \sqsubseteq D$ and $D \sqsubseteq C$. Structural transformation is applied as follows:

Definition 3. *Let B be a forgetting symbol and C be a concept such that $B \in sig(C)$. We say C is a* B-concept *if C is on the form of $B \odot E$ or $\neg B \odot E$ where E is an $\mathcal{ALC}$ concept and $\odot \in \{\sqcup, \sqcap\}$.*

Definition 4. *Let $\mathcal{O}$ be an ontology, B a forgetting symbol, α an axiom in $\mathcal{O}$, and $C = \alpha|_\iota$ be a concept of the form $\mathcal{Q}r.E$ where E is a B-concept, r is a role name, $\mathcal{Q} \in \{\exists, \forall\}$, and C is a concept that occurs in α at position ι. We say α is* structurally transformed *when replacing it with axiom $\alpha[\iota/\mathcal{Q}r.D]$, and adding the following axiom to the ontology:*

$$def(\iota, \alpha, D) = \begin{cases} D \sqsubseteq E & \text{if } pol(\alpha, \iota) = 1; \\ E \sqsubseteq D & \text{if } pol(\alpha, \iota) = -1, \end{cases}$$

where D is a fresh concept symbol, i.e., $D \notin sig(\mathcal{O})$. We say $\mathcal{O}$ is structurally transformed *if the above transformation is applied exhaustively on all axioms until no forgetting symbol appears under role restriction.*

Example 8. Let $\mathcal{O} = \{A \sqsubseteq \exists r.(B \sqcap C), \exists r.B \sqsubseteq E, E \sqsubseteq \neg\exists r.F\}$, and $\mathcal{F} = \{B, F\}$. Then $\mathcal{O}$ is transformed into $\{A \sqsubseteq \exists r.D_1, D_1 \sqsubseteq B \sqcap C, \exists r.D_2 \sqsubseteq E, B \sqsubseteq D_2, E \sqsubseteq \exists r.D_3, F \sqsubseteq D_3\}$ where D_1, D_2 and D_3 are fresh concept symbols.

The concept symbols introduced by structural transformation are called *helper symbols*. These must be fresh. That is, they are not part of the signature of the ontology when they are introduced. We denote by N_h the set of introduced helper symbol.

Lemma 2. *Let $\mathcal{O}_1$ be an ontology, and $\mathcal{O}_2$ the result of applying the above structural transformation with respect to some forgetting signature $\mathcal{F}$. Let N_h be the set of introduced helper symbols. Then, for every model $\mathcal{I}$ of $\mathcal{O}_1$ there is a model $\mathcal{J}$ of $\mathcal{O}_2$, and vice versa, such that $\mathcal{I} \sim_{N_h} \mathcal{J}$.*

Proof. Consider a single application of the transformation in Definition 4. Let B be a forgetting symbol and $\bar{\mathcal{O}}_1$ is the ontology after applying the transformation. Consider first the positive case when C is replaced by $\mathcal{Q}r.D$, and the axiom $D \sqsubseteq E$ is appended to the ontology. Let $\mathcal{I}$ be a model of $\bar{\mathcal{O}}_1$. Since $\mathcal{I} \models D \sqsubseteq E$, we have $d \in \exists r.D$ implies that $d \in \exists r.E$ for every domain element $d \in \Delta^{\mathcal{I}}$. Thus, $\mathcal{I} \models \mathcal{O}_1$. Let $\mathcal{I}$ be a model of $\mathcal{O}_1$. We construct a new model $\mathcal{J}$ such that $\Delta^{\mathcal{J}} = \Delta^{\mathcal{I}}$. We set $\mathcal{J}$ equal to $\mathcal{I}$ on all interpretations and additionally interpret D as: $D^{\mathcal{J}} = \{y \in E^{\mathcal{I}} | (x, y) \in r^{\mathcal{I}}\}$. It follows that for every domain element $d \in \Delta^{\mathcal{J}}$ we have $d \in (\mathcal{Q}r.E)^{\mathcal{I}}$ iff $e \in (\mathcal{Q}r.D)^{\mathcal{J}}$. Hence, $\mathcal{J} \models \mathcal{O}_1$ and $\mathcal{J} \models \bar{\mathcal{O}}_1$. Also, since all other symbols are interpreted the same in both models, we have $\mathcal{I} \sim_D \mathcal{J}$. Second, consider the negative case when C is replaced by $\mathcal{Q}r.D$, and the axiom $E \sqsubseteq D$ is appended to the ontology. Let $\mathcal{I}$ be a model of $\bar{\mathcal{O}}_1$. We need to show that for every domain element d we have $d \in (\neg\mathcal{Q}r.D)^{\mathcal{I}}$ implies $d \in (\neg\mathcal{Q}r.E)$. Suppose $d \in (\neg\mathcal{Q}r.D)^{\mathcal{I}}$, then $d \in (\bar{\mathcal{Q}}r.\neg D)^{\mathcal{I}}$ where $\bar{\mathcal{Q}} = \{\exists, \forall\} \backslash \mathcal{Q}$. Since $\mathcal{I} \models \neg D \sqsubseteq \neg E$ (because $E \sqsubseteq D$ is equivalent to $\neg D \sqsubseteq \neg E$), it follows then that $d \in (\bar{\mathcal{Q}}r.\neg E)^{\mathcal{I}}$. Hence, $d \in (\neg\mathcal{Q}r.E)^{\mathcal{I}}$, and so $\mathcal{I} \models \mathcal{O}_1$. Let $\mathcal{I}$ be a model of $\mathcal{O}_1$. We construct a new model $\mathcal{J}$ such that $\Delta^{\mathcal{J}} = \Delta^{\mathcal{I}}$ and set $\mathcal{J}$ equal to $\mathcal{I}$ on all interpretations and additionally interpret D as: $D^{\mathcal{J}} = E^{\mathcal{J}}$. It follows directly that: $\mathcal{J} \models \mathcal{O}_1$, $\mathcal{J} \models \bar{\mathcal{O}}_1$, and $\mathcal{I} \sim_D \mathcal{J}$. Repeating the transformation exhaustively on $\mathcal{O}_1$, it follows by Lemma 1 that for every model $\mathcal{I}$ of $\mathcal{O}_1$, there is a model $\mathcal{J}$ of $\mathcal{O}_2$ and vice versa such that $\mathcal{I} \sim_{\{D_1\} \cup \cdots \cup \{D_n\}} \mathcal{J}$. That is, $\mathcal{I} \sim_{N_h} \mathcal{J}$.

Following structural transformation, the forgetting symbols $\mathcal{F}$ are forgotten iteratively using the rules of the calculus given in Fig. 1. In each iteration, the subset of axioms of the ontology containing the forgetting symbol B is extracted and viewed as a first-order formula ϕ_B. The *Resolution* rule in Fig. 1 is applied to ϕ_B. The rule creates a disjunction of the two first-order formulas $\phi_B^{\top}$ and $\phi_B^{\bot}$ where the forgetting symbol is replaced by $\top$ and $\bot$, respectively. The *Normalization* and the *Disjunction Elimination* rules eliminate the disjunction between $\phi_B^{\top}$ and $\phi_B^{\bot}$; so prepare the result of the *Resolution* rule to be restored to a set of DL axioms. The *Normalization* rule distributes disjunction between $\phi_B^{\top}$ and $\phi_B^{\bot}$ inwards. The *Disjunction Elimination* rule then converts the disjunction of two formulas into a single formula that can be expressed as a DL axiom. In addition to the rules in Fig. 1, we eagerly eliminate tautologous axioms and formulas.

Definition 5. *An axiom* {*formula*} *is a* tautology *if it takes any of the following forms:*

1. $\bot \sqsubseteq C$ $\{\pi(\bot \sqsubseteq C)\}$
2. $C \sqsubseteq \top$ $\{\pi(C \sqsubseteq \top)\}$*; or*
3. $A \sqcap C \sqsubseteq A \sqcup D$.

where A, C and D are general concepts.

Resolution (Res)

$$\frac{\phi_B}{\phi_B^{\top} \vee \phi_B^{\bot}}$$

where ϕ_B is a first-order formula representing the set of axioms of the ontology that contain the symbol B.

Normalization (Norm)

$$\frac{(A_1^1 \wedge \cdots \wedge A_n^1) \vee (A_1^2 \wedge \cdots \wedge A_m^2)}{(A_1^1 \vee A_1^2) \wedge \cdots \wedge (A_n^1 \vee A_m^2)}$$

where $A_1^1, \ldots, A_n^1, A_1^2, \ldots, A_m^2$ are formulas.

Disjunction Elimination (DElim)

$$\frac{(C_1 \sqsubseteq D_1) \vee (C_2 \sqsubseteq D_2)}{(C_1 \sqcap C_2) \sqsubseteq D_1 \sqcup D_2}$$

where C_1, D_1, C_2, and D_2 are concepts.

Fig. 1. Forgetting Calculus

A tautologous axiom is eliminated by removing it from the ontology, and a tautologous formula is eliminated by replacing it with $\top$.

Example 9. Consider the ontology $\mathcal{O} = \{A \sqsubseteq \exists r.D_1, D_1 \sqsubseteq B \sqcap C, \exists r.D_2 \sqsubseteq E, B \sqsubseteq D_2\}$ after structural transformation, and suppose we want to forget B.

$$\begin{aligned}
&\phi_B = (D_1 \sqsubseteq B \sqcap C) \wedge (B \sqsubseteq D_2) \\
&\phi_B^{\top} = (D_1 \sqsubseteq C) \wedge (\top \sqsubseteq D_2), \quad \phi_B^{\bot} = (D_1 \sqsubseteq \bot) \wedge (\bot \sqsubseteq D_2) \equiv D_1 \sqsubseteq \bot \\
&\phi_B^{\top} \vee \phi_B^{\bot} = ((D_1 \sqsubseteq C) \wedge (\top \sqsubseteq D_2)) \vee (D_1 \sqsubseteq \bot) && \text{Res} \\
&\equiv ((D_1 \sqsubseteq C) \vee (D_1 \sqsubseteq \bot)) \wedge ((\top \sqsubseteq D_2) \vee (D_1 \sqsubseteq \bot)) && \text{Norm} \\
&\equiv (D_1 \sqsubseteq C) \wedge (D_1 \sqsubseteq D_2) && \text{DElim}
\end{aligned}$$

The resulting formula is translated to the set of DL axioms $\{D_1 \sqsubseteq C, D_1 \sqsubseteq D_2\}$ and added to the remaining axioms of $\mathcal{O}$. The semantic forgetting view is therefore $\mathcal{V} = \{A \sqsubseteq \exists r.D_1, \exists r.D_2 \sqsubseteq E, D_1 \sqsubseteq C, D_1 \sqsubseteq D_2\}$.

Definition 6. *Suppose B is the forgetting symbol. Consider the forgetting rules in Fig. 1. We say that a rule is* sound *if and only if for every model $\mathcal{I}$ of the premise there is a model $\mathcal{J}$ of the conclusion, and vice versa, such that $\mathcal{I} \sim_B \mathcal{J}$.*

Theorem 1. *The rules in Fig. 1 are sound.*

Proof. The output of the *Normalization* rule is a syntactic variant of the premise. For the *Disjunction Elimination* rule, in FOL notations, the premise of the rule

is:

$$\begin{aligned}&(\pi_x(C_1) \to \pi_x(D_1)) \vee (\pi_x(C_2) \to \pi_x(D_2))\\ &\equiv (\pi_x(C_1) \wedge \pi_x(C_2)) \to (\pi_x(D_1) \vee \pi_x(D_2))\\ &\equiv \pi((C_1 \sqcap C_2) \sqsubseteq (D_1 \sqcup D_2))\end{aligned}$$

The conclusion is a syntactic veriant of the premise, so the premise and the conclusion coincide on all models, and the rule is sound. For the *Resolution* rule, suppose the forgetting symbol is B. The premise ϕ_B is the first-order formula on the form: $\phi_B = \forall x\ \pi(A_1) \wedge \forall x\ \pi(A_2) \wedge \cdots \wedge \forall x\ \pi(A_n)$ where $A_1, \ldots, A_n$ are axioms of the ontology that contain the symbol B. We rewrite ϕ_B as $\forall x\ F_1(x) \wedge \forall x\ F_2(x) \wedge \cdots \wedge \forall x\ F_n(x)$ where $F_i(x)$ is the formula $\pi(A_i)$ with one free variable x. Let $\mathcal{I}$ be an arbitrary model of ϕ_B, we have:

$$\begin{aligned}\mathcal{I} &\models \forall x F_1(x) \wedge \forall x F_2(x) \wedge \cdots \wedge \forall x F_n(x)\\ &\equiv \forall x(F_1(x) \wedge \cdots \wedge F_n(x))\\ &\equiv \forall x((B(x) \wedge F_1(x) \wedge \cdots \wedge F_n(x)) \vee (\neg B(x) \wedge F_1(x) \wedge \cdots \wedge F_n(x)))\\ &\models \forall x((F_1(x) \wedge \cdots \wedge F_n(x))[B(x)/\top] \vee (F_1(x) \wedge \cdots \wedge F_n(x))[B(x)/\bot])\\ &\equiv \forall x(\phi_B^{\top} \vee \phi_B^{\bot})\end{aligned}$$

Therefore, $\mathcal{I}$ is also a model of the conclusion. Let $\mathcal{J}$ be an arbitrary model of the conclusion. We move to a new model $\mathcal{I}$ which coincides with $\mathcal{J}$ on everything and, additionally, interprets the predicate B as follows:

$$\mathcal{I} \models B(a) \text{ iff } \mathcal{I} \models \phi_B^{\top}[x/a]$$

for every element a in $\Delta^{\mathcal{I}}$. It follows directly that $\mathcal{I}$ is a model of ϕ_B. Also, by construction of $\mathcal{I}$, we have $\mathcal{I} \sim_B \mathcal{J}$. Therefore, the *Resolution* rule is sound. Finally, *Tautology Elimination* is a standard equivalence preserving operation. To see this, let $\mathcal{O}$ be an ontology and α be the axiom eliminated from $\mathcal{O}$. Let $\mathcal{I}$ be a model of $\mathcal{O}$, then $\mathcal{I}$ is a model of $\mathcal{O}\backslash\{\alpha\}$ since removing axioms does not invalidate the model. Let $\mathcal{I}$ be an arbitrary model of $\mathcal{O}\backslash\{\alpha\}$. Since α is a tautology, then $\mathcal{I} \models \alpha$, and it follows that $\mathcal{I} \models \mathcal{O}$.

Theorem 2. *The size of the semantic forgetting view is, in the worst case, exponential in the size of the original ontology and double exponential in the size of the forgetting signature.*

Proof. The result of the first stage of the forgetting method, i.e., definition expansion, can be exponential in the size of the original ontology [32, 2].

Let n be the size of the original ontology, and m be the size of the ontology after definition expansion. We have, $m \leq \mathcal{O}(2^n)$. Suppose k is the size of the forgetting signature not eliminated by definition expansion. Structural transformation is bounded by the number of role restrictions in which the forgetting symbols occurs. Therefore, it can only result in a linear expansion. The size of the output of one forgetting iteration is influenced mainly by the *Normalisation*

rule in Fig. 1. An application of this rule produces, at most, a quadratic number of axioms. So the result of one iteration of forgetting is $\mathcal{O}(m^2)$. Repeating this for k symbols gives $\mathcal{O}(m^{2^k})$, or $\mathcal{O}(2^{n.2^k})$.

4 Eliminating Helper Symbols

In the second stage of the method, the goal is to attempt to eliminate the helper symbols introduced in the vocabulary of the semantic forgetting view. Since it is not always possible to represent the semantic forgetting view in $\mathcal{ALC}$, a complete elimination of helper symbols is not guaranteed. The elimination of helper symbols is done by reversing the structural transformation process.

Definition 7. *Let D be a helper symbol. We denote by* $\text{def}(D)$ *the set of all axioms that can be put in the forms $D \sqsubseteq C$ or $C \sqsubseteq D$.*

Definition 8. *Let D be a helper symbol and consider any $\alpha \in def(D)$. Then, α is* improper *if:*

1. *α can be put equivalently in the forms $D \sqcap \bar{D} \sqsubseteq C$ or $C \sqsubseteq D \sqcup \bar{D}$ for any $\bar{D} \in N_h$ where $\bar{D} \neq D$ and C is a general $\mathcal{ALC}$ concept; and*
2. *α can be put equivalently in the forms $D \sqsubseteq C$ or $C \sqsubseteq D$ and $\mathcal{Q}r.D$ occurs in C where $\mathcal{Q} \in \{\exists, \forall\}$, and r is any role name.*

Otherwise, the definition axiom is proper. *$def(D)$ is* proper *if all axioms $\alpha \in def(D)$ are proper, otherwise, $def(D)$ is* improper.

Example 10. Let D_1 and D_2 be two helper symbols. Consider the axioms $\alpha_1 = D_1 \sqsubseteq C, \alpha_2 = D_1 \sqcap D_2 \sqsubseteq C, \alpha_3 = D_1 \sqsubseteq D_2, \alpha_4 = \forall r.D_1 \sqsubseteq D_1$. The axioms α_1 is a proper axiom of $\text{def}(D_1)$, and α_3 is a proper of $\text{def}(D_1)$ and $\text{def}(D_2)$. The axiom α_2 is improper because it satisfies the first condition of Definition 8. The axiom α_4 is also improper because it satisfies the second condition of Definition 8.

A helper symbol D for which $\text{def}(D)$ is improper cannot be eliminated. Improper axioms satisfying 1 of Definition 8 have been investigated in [35] where it was shown that eliminating these helper symbols leads to loss of semantic information that we want to preserve. Definition 8.2 is the case when the ontology contains a cycle over a subset of the forgetting vocabulary. The helper symbol in this case functions as a witness to this cycle. Eliminating this helper symbol requires extending the language of the semantic view with fixpoints [21, 33, 38].

A helper symbol D for which $def(D)$ is proper can be eliminated. The elimination rules of D are presented in Fig. 2. The *NDef* and the *PDef* rules replace the set $def(D)$ in the ontology with the semantically equivalent conclusion axioms. The *Negative Ackermann Substitution* and the *Positive Ackermann Substitution* rules eliminate D using the Ackermann approach [1, 36]. The premise $C \sqsubseteq D$ ($D \sqsubseteq C$) of the Positive (Negative) Ackermann Substitution rule is usually the result generated by the PDef (NDef) rule.

NDef

$$\frac{(D \sqcap C_1 \sqsubseteq E_1) \wedge \cdots \wedge (D \sqcap C_n \sqsubseteq E_n)}{D \sqsubseteq ((\neg C_1 \sqcup E_1) \sqcap \cdots \sqcap (\neg C_n \sqcup E_n))}$$

where D is a helper symbol, and the axioms $D \sqcap C_i \sqsubseteq E_i$ where $1 \leq i \leq n$ are proper axioms in def(D).

PDef

$$\frac{(C_1 \sqsubseteq D \sqcup E_1) \wedge \cdots \wedge (C_n \sqsubseteq D \sqcup E_n)}{((\neg C_1 \sqcap \neg E_1) \sqcap \cdots \sqcap (\neg C_n \sqcap \neg E_n)) \sqsubseteq D}$$

where D is a helper symbol, and the axioms $D \sqcap C_i \sqsubseteq E_i$ where $1 \leq i \leq n$ are proper axioms in def(D).

Negative Ackermann Substitution (NAck)

$$\frac{\mathcal{O} \cup (D \sqsubseteq C)}{\mathcal{O}[D/C]}$$

where D is a helper symbol and may occur in $\mathcal{O}$ only positively

Positive Ackermann Substitution (PAck)

$$\frac{\mathcal{O} \cup (C \sqsubseteq D)}{\mathcal{O}[D/C]}$$

where D is a helper symbol and may occur in $\mathcal{O}$ only negatively.

Fig. 2. Helper symbol elimination rules

Lemma 3. *Let $\mathcal{O}_1$ be an ontology and $\mathcal{O}_2$ the ontology after eliminating a helper symbol D using the rules in Fig.2. Then, for every model $\mathcal{I}$ of $\mathcal{O}_1$ there is a model $\mathcal{J}$ of $\mathcal{O}_2$, and vice versa, such that $\mathcal{I} \sim_D \mathcal{J}$.*

Proof. The result of the NDef (PDef) rule is a syntactic variant of the premises. Since the NAck and PAck rules achieve the reverse of *structural transformation*, it follows by the the proof of Lemma 2 that the NAck and PAck rules preserve the interpretations of all symbols up to the eliminated helper symbols.

Theorem 3. *Let $\mathcal{O}$ be an ontology, $\mathcal{F}$ a forgetting signature, and $\mathcal{V}$ be the semantic forgetting view generated as described in sections 3 and 4. For every model $\mathcal{I}$ of $\mathcal{O}$ there is a model $\mathcal{J}$ of $\mathcal{V}$, and vice versa, such that $\mathcal{I} \sim_{\mathcal{F} \cup N_h} \mathcal{J}$, where N_h is the set of helper symbols in $\mathcal{V}$.*

Proof. Let $\mathcal{O}_1$ be the ontology $\mathcal{O}$ after structural transformation, $\mathcal{O}_2$ the result of applying the calculus in Fig. 1 exhaustively on $\mathcal{O}_1$, and $\mathcal{V}$ the result of eliminating helper symbols from $\mathcal{O}_2$ using the rules in Fig. 2. By Lemma 2, the interpretations of all symbols of $\mathcal{O}$ are preserved in the models of $\mathcal{O}_1$. By Theorem 1, the models of $\mathcal{O}_1$ and $\mathcal{O}_2$ may only differ on the interpretation of the symbols $\mathcal{F}$. Finally, by Lemma 3, the models of $\mathcal{O}_2$ and $\mathcal{V}$ may only differ on the interpretation of

the helper symbols N_h. It follows from the above and Lemma 1 that for every model $\mathcal{I}$ of $\mathcal{O}$ there is a model $\mathcal{J}$ of $\mathcal{V}$, and vice versa, such that $\mathcal{I} \sim_{\mathcal{F} \cup N_h} \mathcal{J}$.

5 Discussion of the Forgetting Method

The presented forgetting method implicitly views the forgetting symbols as existentially quantified second-order symbols. In order to completely eliminate the forgetting symbols, the method introduces helper symbols which can also be seen as existentially quantified second-order symbols. Therefore, the language of the generated semantic view can be seen as a fragment of second-order logic (SOL). An alternative method [26] which also generates a semantic forgetting view in SOL is to quantify over the forgetting symbols. Using this approach, the semantic forgetting view of the ontology $\mathcal{O}$ with respect to $\{B\}$ is $\exists \bar{B}\ \mathcal{O}[B/\bar{B}]$, where $\mathcal{O}[B/\bar{B}]$ is the result of replacing B by the helper symbol $\bar{B}$ everywhere in $\mathcal{O}$. Since $\bar{B}$ can be renamed to B, this is equivalent to $\exists B\ \mathcal{O}$. Therefore, a simpler form of the semantic forgetting view generated by [26] is $\exists B_1 \exists B_2 \ldots \exists B_n\ \mathcal{O}$ where $B_1, B_2, \ldots, B_n$ are the forgetting symbols.

Our presented method can be seen as a non-trivial extension of the method in [26], because it introduces further existentially quantified symbols. Extending the forgetting method in this way is necessary to make the method suitable for ontology extraction and applicable to DL syntax. We show this through the following example.

Example 11. Let $\mathcal{O} = \{A \sqsubseteq \exists r.(B \sqcup C) \sqcap \exists r.(\neg B \sqcup \neg C)\}$, and $\mathcal{F} = \{B\}$. The semantic forgetting view $\mathcal{V}_1$ obtained by quantifying over B is $\exists B\ (A \sqsubseteq \exists r.(B \sqcup C) \sqcap \exists r.(\neg B \sqcup \neg C))$. The equivalent semantic forgetting view is $\mathcal{V}_2 = \{A \sqsubseteq \exists r.\top\}$ which is generated by our method. In essence, $\mathcal{V}_1$ is syntactically the same as $\mathcal{O}$. So, only quantifying over the forgetting symbol returns the original ontology and fails to extract the expected ontology. With our method, we can obtain it the expected semantic forgetting view.

The above example shows that the method in [26] is not suitable for ontology extraction. To make it suitable to this purpose, augmenting the method in [26] with a second-order quantifier elimination method such as formula (3) is required. While second-order quantifier elimination is sufficient in the syntax of FOL [13, 14], it is not directly applicable to DL ontologies.

Example 12. Let $\mathcal{O} = \{A \sqsubseteq \exists r.B \sqcap \exists r.\neg B\}$ and $\mathcal{F} = \{B\}$. By quantifying over B, we get the semantic forgetting view $\mathcal{V}_1 = \exists B\ (A \sqsubseteq \exists r.B \sqcap \exists r.\neg B)$ Applying the calculus in Fig. 1 directly on $\mathcal{V}_1$ gives: $\mathcal{V}_2 = \{A \sqsubseteq \exists r.\top\}$ which is not the correct semantic forgetting view of $\mathcal{O}$ with respect to $\mathcal{F}$. Our method solves this problem by applying structural transformation prior to the calculus in Fig. 1. The correct semantic forgetting view generated by our method is $\{A \sqsubseteq \exists r.D_1 \sqcap \exists r.D_2, D_1 \sqcap D_2 \sqsubseteq \bot\}$.

6 Preserving Ontology Structure

A common criticism of existing forgetting methods, e.g. [13, 28, 21–23, 42, 43, 45, 44], is that they do not preserve the *form* of the original ontology. This reduces the readability of the extracted ontology, and makes it harder for human inspection and debugging.

Example 13. Let $\mathcal{O}_1 = \{A \sqsubseteq B \sqcap C \sqcap D\}$, $\mathcal{O}_2 = \{A \equiv B, B \equiv C \sqcap D\}$, and $\mathcal{O}_3 = \{\forall r.B \sqsubseteq C, A \sqsubseteq B\}$. Consider the forgetting signature $\mathcal{F} = \{B\}$. The forgetting views generated by the methods [21–23, 42, 43, 45, 44] of $\mathcal{O}_1, \mathcal{O}_2$, and $\mathcal{O}_3$ respectively with respect to $\mathcal{F}$ are: $\mathcal{V}_1 = \{\top \sqsubseteq \neg A \sqcup C, \top \sqsubseteq \neg A \sqcup D\}$, $\mathcal{V}_2 = \{\top \sqsubseteq \neg A \sqcup C, \top \sqsubseteq \neg A \sqcup D, \top \sqsubseteq \neg C \sqcup \neg D \sqcup A\}$, and $\mathcal{V}_3 = \{\top \sqsubseteq C \sqcup \exists r.\neg A\}$. Our forgetting method produces: $\mathcal{V}'_1 = \{A \sqsubseteq C \sqcap D\}$ (instead of $\mathcal{V}_1$), $\mathcal{V}'_2 = \{A \equiv C \sqcap D\}$ (instead of $\mathcal{V}_2$), and $\mathcal{V}'_3 = \{\forall r.A \sqsubseteq C\}$ (instead of $\mathcal{V}_3$).

It may be observed that the forgetting views generated by our method are more readable and preserve the forms of the original ontologies. They:

1. *Reuse concepts of the original ontology*: $\mathcal{V}'_1$ consists of one subsumption axiom which reuses the concept $C \sqcap D$ whereas the concept $C \sqcap D$ does not occur in $\mathcal{V}_1$.
2. *Preserve equivalence axioms*: $\mathcal{V}'_2$ consists of one equivalence axiom whereas $\mathcal{V}_2$ consists of three subsumption axioms.
3. *Preserve position and polarity of concepts*: Similar to $\mathcal{O}_3$, $\mathcal{V}'_3$ has the universal quantifier on the left hand side of the output axiom, whereas $\mathcal{V}_3$ contains an existential quantifier on the right hand side of the axiom. In general, axioms of $\mathcal{V}_1, \mathcal{V}_2$, and $\mathcal{V}_3$ appear on the form $\top \sqsubseteq E$ which is not the case for $\mathcal{V}'_1, \mathcal{V}'_2$, and $\mathcal{V}'_3$.

These properties are achieved in the following ways: (1) The initial definition expansion step avoids unnecessary conversion of equivalence axioms into subsumption axioms. (2) The forgetting calculus does not require the input axioms to be transformed to a special normal or clausal form (like other methods [13, 28, 21–23, 42, 43, 45, 44]). For example, while traditional resolution requires the premises in the disjunctive clausal form, we use a non-traditional form of resolution which only replaces a forgetting symbol with $\top$ and $\bot$. (3) The *Disjunction Elimination* rule reuses the concept expressions of the premise axioms in the generated axiom, and preserves their position relative to the subsumption operator. (4) Although effort has been put into eliminating the helper symbols, some helper symbols may be present in the returned semantic forgetting view only to preserve the form of the original ontology. We illustrate this last point through the following examples.

Example 14. Consider the ontology $\mathcal{O}_1 = \{\forall r.B \sqsubseteq \exists r.B\}$ and the forgetting signature $\mathcal{F} = \{B\}$. The semantic forgetting view of $\mathcal{O}_1$ with respect to $\mathcal{F}$ computed by e.g. the method of [45] is $\mathcal{V}_1 = \{\top \sqsubseteq \exists r.\top\}$. However, the result generated by our method is $\{\forall r.D \sqsubseteq \exists r.D\}$ where D is a helper symbol. The helper symbol D allows the form of the original ontology to be preserved.

Table 1. Statistics of the used BioPortal ontologies and forgetting signatures.

	Maximum	Minimum	Average	Median
Ontology Size (axioms)	133290	40	24760	4653
Forgetting Signature Size (concepts)	36697	1	3526	771

In the next example, we show that even for semantically equivalent ontologies, the number of helper symbols in the language of the semantic view may differ.

Example 15. Consider the ontology $\mathcal{O}_2 = \{\top \sqsubseteq \exists r.B \sqcup \exists r.\neg B\}$ which is a syntactic variant of $\mathcal{O}_1$ in Example 14. Let $\mathcal{F} = \{B\}$. The semantic forgetting view of $\mathcal{O}_2$ with respect to $\mathcal{F}$ generated by the proposed method is: $\mathcal{V}_2 = \{\top \sqsubseteq \exists r.D_1 \sqcup \exists r.D_2, D_1 \sqcap D_2 \sqsubseteq \bot\}$ where D_1 and D_2 are helper symbols. Thus, the method uses two helper symbols to represent the semantic forgetting view instead of one helper symbol as in Example 14. Note that the semantic forgetting view of $\mathcal{O}_2$ generated by [45] is $\mathcal{V}_1$ as in Example 14.

The above examples show that the preservation of the original ontology form is valued by the proposed method over the elimination of the helper symbols. This is an advantage of our method over the other methods [13, 21–23, 42, 43, 45, 44] as it preserves the syntactic modeling choices made in the original ontology. This is beneficial for domains such as medical ontologies where often strict modelling guidelines must be followed.

7 Empirical Evaluation

We conducted experiments on popular ontologies from the NCBO BioPortal repository [31]. Starting with 40 ontologies, three experiments were run on each ontology in a total of 120 experiments. The forgetting signatures $\mathcal{F}_1, \mathcal{F}_2$, and $\mathcal{F}_3$ of these three experiments respectively were selected such that $\mathcal{F}_1 \subset \mathcal{F}_2 \subset \mathcal{F}_3$. Statistics about the sizes of the ontologies and forgetting signatures are presented in Table 1. Each experiment was assigned 2GB memory and ran on a x64-based processor Intel(R) Core(TM) i5 CPU @ 2.7GHz with a 64-bit operating system (macOS Catalina 10.15.7). Statistics gathered from the experiments are shown in Fig. 3. Chart A of Fig. 3 shows the average running time of the experiments across the three settings of forgetting 10%, 30%, and 50% of the signature. The chart shows a polynomial increase in the average running times (369, 699, and 1091 seconds in the 10%, 30%, and 50% settings respectively).

A breakdown of the average running times is shown in chart B of Fig. 3. The chart shows the average time consumed for eliminating the forgetting symbols using the rules in Fig. 1, and the average time consumed for eliminating the helper symbol using the rules in Fig. 2, relative to the average execution time of the whole forgetting process. We found that the time consumed for eliminating the helper symbols occupied the majority of the time of the whole forgetting

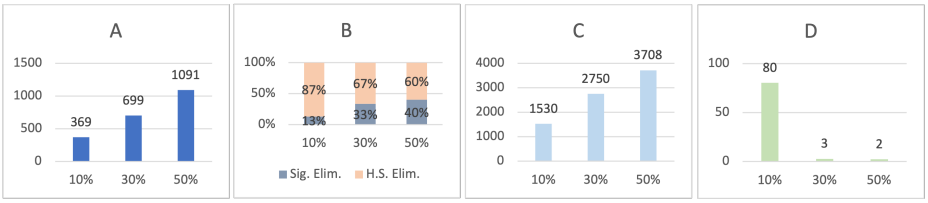

Fig. 3. Chart A: Average execution times (seconds) of the proposed method. Chart B: Breakdown of the execution time into the time consumed eliminating forgetting symbols and the time consumed eliminating helper symbols. Chart C: Average number of helper symbols introduced by *structural transformation*. Chart D: Average number of helper symbols remaining in the semantic forgetting view.

process. However, we noticed a decreasing trend in the time consumed for eliminating the helper symbols. Execution of the rules in Fig. 1 consumed on average 13%, 33%, and 40% of the time of the whole forgetting process in the 10%, 30%, and 50% settings respectively, whereas execution of the rules in Fig. 2 consumed on average 87%, 67%, and 60% of the time. This suggests that helper symbols not only simplify the forgetting method, but also the cost of eliminating them becomes less significant when forgetting large subsets of the vocabulary.

We measured in chart C of Fig. 3 the trend of introducing helper symbols by structural transformation. We observed a linear increase in the average number of introduced helper symbols. On average there were 1530, 2750, and 3708 introduced helper symbols in the 10%, 30%, and 50% settings respectively. This increase was expected given the increase in the number of forgetting symbols. We also measured the number of introduced helper symbols as a ratio to the forgetting signature. On average, the helper symbols introduced by the structural transformation process were 107%, 56%, and 47% of the forgetting signature in the 10%, 30%, and 50% settings respectively. We may conclude that a factor of the downward trend in the time consumed by the elimination of helper symbols is the relative decrease in the number forgetting symbols which caused new helper symbols to be introduced.

We observed that in some cases the above factor also reduced the number of helper symbols that remain in the semantic forgetting view. The following example shows the idea.

Example 16. Let $\mathcal{O} = \{A_1 \sqsubseteq C \sqcup \exists r.B, A_2 \sqsubseteq \exists r.\neg B\}$, $\mathcal{F}_1 = \{B\}$, and $\mathcal{F}_2 = \{B, C\}$. The semantic forgetting view of $\mathcal{O}$ with respect to $\mathcal{F}_1$ is $\mathcal{V}_1 = \{A_1 \sqsubseteq C \sqcup \exists r.D_1, A_2 \sqsubseteq \exists r.D_2, D_1 \sqcap D_2 \sqsubseteq \bot\}$. The semantic forgetting view of $\mathcal{O}$ with respect to $\mathcal{F}_2$ is $\mathcal{V}_2 = \{A_2 \sqsubseteq \exists r.\top\}$. The concept name C does not appear under role restriction in $\mathcal{O}$, and does not result in introducing a helper symbol. If only B is forgotten from $\mathcal{O}$, then the helper symbols D_1 and D_2 appear in the semantic forgetting view. Forgetting additionally the concept name C prevents D_1 and D_2 from occurring the semantic forgetting view.

The observation illustrated by the example explains chart D in Fig. 3. Chart D shows the average number of helper symbols appearing the vocabulary of the final forgetting view. While on average 80 helper symbols appeared in the forgetting views of the 10% setting, only 3 and 2 helper symbols on average appeared respectively in the forgetting views of the 30% and 50% settings.

Next we compared the performance of our method with the Fame(Q) tool [46], a java implementation of the method of [45]. While Fame(Q) does not produce the complete semantic view (see Example 4), it captures the semantic information that is representable in $\mathcal{ALCOQ}^{\neg,\sqcup,\sqcap}$. We tried executing the above experiments using Fame. However, we observed memory issues in 52 experiments. This happened more often in experiments with large forgetting signatures and ontologies. To overcome this problem we constructed our own ontologies and experiments. The ontologies are constructed such that: (1) They are extracted from real life ontologies; and (2) they model one or several related topics. Construction was done in the following way: (1) Collect a set of random concept names Σ from the *Interlinking Ontology for Biological Concepts (IOBC)* ontology [24]. (2) Retrieve from the *NCBO Bioportal* the related ontologies where the symbols in Σ appear more frequently. (3) From the retrieved ontologies, extract and combine into a single ontology, the subsets of the axioms containing symbols in Σ.

The construction guaranteed that each generated ontology contained information that described the concept names (topics) extracted from the IOBC ontology. Since these concepts were extracted from the same ontology, i.e., the IOBC ontology, they were semantically related. In total, 28 ontologies were constructed using the method described above. In each construction, at least 25% of the input ontologies were expressed in $\mathcal{ALC}$ or a more expressive logic. Next, we used the constructed ontologies in three different experimental settings: *Low*, *Moderate*, and *High* with a total of 84 experiments. Each setting corresponded to a different choice of the forgetting signature. The *Low* setting meant that at least half of the axioms of the ontology contained a forgetting symbol, and the probability of two forgetting symbols appearing in the same axiom was low. The *Moderate* setting meant that all axioms of the ontology contained a forgetting symbol, and at least one axiom contained two or more forgetting symbols. The *High* setting meant that all axioms of the ontology contained a forgetting symbol, and at least half of the axioms contained two or more forgetting symbols.

The occurrence of several forgetting symbols in the same axioms makes the axioms generated from one forgetting round input to subsequent rounds, because each round in the proposed method and Fame(Q) eliminates only one forgetting symbol. This made the experiments more computationally challenging for both methods, and avoided the drawbacks of reducing the size of the input ontologies.

Statistics about the ontologies and the forgetting signatures are presented in Table 2. Statistics gathered from the experiments are shown in Fig. 4. Chart A in Figure 4 compares the average execution times of Fame(Q) and our proposed method. In general we found the execution time of Fame(Q) stable across the *Low*, *Moderate*, and *High* settings. The average execution times of Fame(Q) were

Table 2. Statistics of the constructed ontologies and forgetting signatures.

	Maximum	Minimum	Average	Median
Ontology Size (axioms)	1422	78	302	201
Forgetting Signature Size (concepts)	324	1	27	10

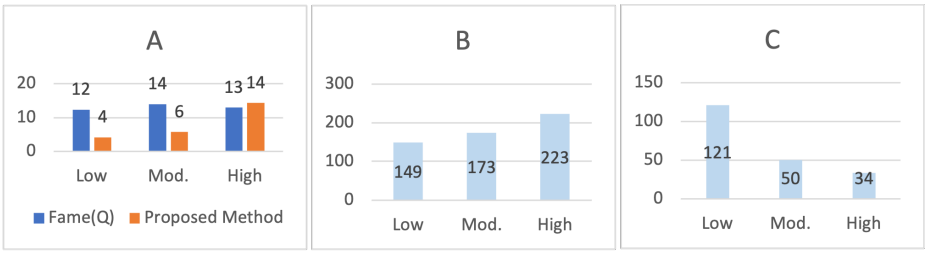

Fig. 4. Chart A: Execution time(seconds) comparison with Fame(Q). Chart B: Average number of helper symbols introduced by *structural transformation*. Chart C: Average number of the introduced helper symbols as a ratio to the number of forgetting symbols.

12, 14, 13 seconds respectively in the three settings. In contrast, the time consumed by our method grew polynomially over the different settings. This growth was expected since more more semantic information was captured by our method as the forgetting problem becomes harder. Since Fame(Q) is not complete, we do not see the same increase in the time consumed by Fame. We noticed in the *Low* and *Moderate* settings that our method consumes less time than Fame, which is not expected since it should still preserve more information than Fame. We found that Fame(Q) used a reasoner to perform some optimizations the output, which resulted in consuming more time than our method.

We measured the number of introduced helper symbols across the three settings in the same way as before. Chart B in Fig. 4 shows an increase in the number introduced helper symbols as we moved to harder settings. On average, 149, 173, and 223 helper symbols were introduced by *structural transformation* in the *Low, Moderate*, and *High* settings respectively. This increase is expected since the number of forgetting symbols increase. However, as shown in Chart C of Fig. 4, the average number of introduced helper symbols as a ratio to the number of forgetting symbols decreased as we go to harder settings. This agreed with the ratios of 107%, 56%, and 47% that were found before when forgetting was done directly on the original BioPortal ontologies.

8 Conclusions

We discussed the problem of semantic forgetting in the context of the description logic $\mathcal{ALC}$. The method captures the semantic forgetting view by possibly allowing helper symbols in the vocabulary of the generated semantic view. An

important feature of our method is preserving the syntactic form of the original ontology, which increases the readability of the semantic view.

Evaluation of our method showed that our method performs well on large-scale ontologies when forgetting 10%, 30%, and 50% of the vocabulary. It also showed that the number of helper symbols that are used in the extracted semantic forgetting view decreases as the number of forgetting symbols increases. We compared our method with Fame, a tool that forgets concept and role names in $\mathcal{ALCOQ}^{\neg,\sqcup,\sqcap}$, in three settings of hardness: *Low*, *Moderate*, and *High*. While Fame(Q) does not capture the complete semantic forgetting view, Experiments showed that our method performed better in the *Low*, and *Moderate* settings. In the *High* setting, Fame(Q) performed marginally better than our method.

References

1. Ackermann, W.: Untersuchungen über das eliminationsproblem der mathematischen logik. Mathematische Annalen **110**, 390–413 (1935)
2. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F.: The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
3. Baader, F., Horrocks, I., Sattler, U.: Description logics. In: Handbook of Knowledge Representation, pp. 135 – 179. Elsevier (2007)
4. Baaz, M., Egly, U., Leitsch, A.: Normal form transformations. In: Handbook of Automated Reasoning, pp. 275–332. North-Holland (12 2001)
5. Boole, G.: An Investigation of the Laws of Thought: On Which Are Founded the Mathematical Theories of Logic and Probabilities. Cambridge University Press (1854)
6. Borgida, A.: On the relative expressiveness of description logics and predicate logics. Artificial Intelligence **82**(1-2), 353–367 (apr 1996)
7. Botoeva, E., Konev, B., Lutz, C., Ryzhikov, V., Wolter, F., Zakharyaschev, M.: Inseparability and conservative extensions of description logic ontologies: A survey. In: Reasoning Web: Logical Foundation of Knowledge Graph Construction and Query Answering: 12th International Summer School. pp. 27–89. Springer (2017)
8. Chen, J., Alghamdi, G., Schmidt, R.A., Walther, D., Gao, Y.: Ontology extraction for large ontologies via modularity and forgetting. In: Kejriwal, M., Szekely, P.A., Troncy, R. (eds.) Proceedings of the 10th International Conference on Knowledge Capture (K-CAPâ€™19). pp. 45–52. ACM (2019)
9. Delgrande, J.: A Knowledge Level Account of Forgetting. Journal of Artificial Intelligence Research **60**, 1165–1213 (2017)
10. Delgrande, J.P.: Towards a Knowledge Level Analysis of Forgetting. In: Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning. pp. 606–609. AAAI Press (2014)
11. Ditmarsch, H., Herzig, A., Lang, J., Marquis, P.: Introspective forgetting. In: Proceedings of the 21st Australasian Joint Conference on Artificial Intelligence. pp. 18–29. Springer (2008)
12. Eiter, T., Kern-Isberner, G.: A Brief Survey on Forgetting from a Knowledge Representation and Reasoning Perspective. KI - Künstliche Intelligenz **33**(1), 9–33 (mar 2019)

13. Gabbay, D.M., Ohlbach, H.J.: Quantifier elimination in second-order predicate logic. In: Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning. pp. 425–435. Morgan Kaufmann (1992)
14. Gabbay, D.M., Schmidt, R.A., Szałas, A.: Second-Order Quantifier Elimination: Foundations, Computational Aspects and Applications, Studies in Logic: Mathematical Logic and Foundations, vol. 12. College Publications (2008)
15. Herzig, A., Mengin, J.: Uniform interpolation by resolution in modal logic. In: Logics in Artificial Intelligence. pp. 219–231. Springer (2008)
16. Hustadt, U., Schmidt, R.A., Georgieva, L.: A Survey of Decidable First-Order Fragments and Description Logics. Journal of Relational Methods in Computer Science **1**, 251–276 (2004)
17. Jung, J., Lutz, C., Pulcini, H., Wolter, F.: Logical separability of incomplete data under ontologies. In: 17th International Conference on Principles of Knowledge Representation and Reasoning. pp. 517–528 (2020)
18. Konev, B., Lutz, C., Walther, D., Wolter, F.: Formal properties of modularisation. In: Modular Ontologies, pp. 25–66. Springer (05 2009)
19. Konev, B., Lutz, C., Walther, D., Wolter, F.: Model-theoretic inseparability and modularity of description logic ontologies. Artificial Intelligence **203**, 66–103 (2013)
20. Konev, B., Walther, D., Wolter, F.: Forgetting and uniform interpolation in large-scale description logic terminologies. In: Proceedings of the 21st International Jont Conference on Artifical Intelligence. p. 830–835. Morgan Kaufmann (2009)
21. Koopmann, P., Schmidt, R.A.: Uniform interpolation of $\mathcal{ALC}$-ontologies using fixpoints. In: Proceedings of the 9th International Symposium on Frontiers of Combining Systems. Lecture Notes in Artificial Intelligence, vol. 8152, pp. 87–102. Springer (2013)
22. Koopmann, P., Schmidt, R.A.: Count and forget: Uniform interpolation of $\mathcal{SHQ}$-ontologies. In: Automated Reasoning. Lecture Notes in Artificial Intelligence, vol. 8562, pp. 434–448. Springer (2014)
23. Koopmann, P., Schmidt, R.A.: Saturation-based forgetting in the description logic $\mathcal{SIF}$. In: Proceedings of the 28th International Workshop on Description Logics. vol. 1350. CEUR-WS.org (2015)
24. Kushida, T., Kozaki, K., Kawamura, T., Tateisi, Y., Yamamoto, Y., Takagi, T.: Interconnection of biological knowledge using nikkajirdf and interlinking ontology for biological concepts. New Generation Computing **37**, 1–25 (09 2019)
25. Lang, J., Liberatore, P., Marquis, P.: Propositional Independence: Formula-Variable Independence and Forgetting. Journal of Artificial Intelligence Research **18**, 391–443 (2003)
26. Lin, F., Reiter, R.: Forget it! In: Proc. AAAI 1994. pp. 154–159 (1994)
27. Lin, F., Reiter, R.: How to progress a database (and why) i. logical foundations. In: Principles of Knowledge Representation and Reasoning, pp. 425 – 436. Morgan Kaufmann (1994)
28. Ludwig, M., Konev, B.: Towards practical uniform interpolation and forgetting for $\mathcal{ALC}$ tboxes. In: Proceedings of the 26th International Workshop on Description Logics. AAAI Press (2013)
29. Lutz, C., Wolter, F.: Foundations for uniform interpolation and forgetting in expressive description logics. In: IJCAI International Joint Conference on Artificial Intelligence (2011)
30. Lutz, C., Wolter, F.: Deciding inseparability and conservative extensions in the description logic $\mathcal{EL}$. Journal of Symbolic Computation **45**, 194–228 (02 2010)

31. Matentzoglu, N., Bail, S., Parsia, B.: A snapshot of the owl web. In: Alani, H., Kagal, L., Fokoue, A., Groth, P., Biemann, C., Parreira, J.X., Aroyo, L., Noy, N., Welty, C., Janowicz, K. (eds.) The Semantic Web – ISWC 2013. pp. 331–346. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
32. Nebel, B.: Terminological reasoning is inherently intractable. Artificial Intelligence **43**(2), 235–249 (1990)
33. Nonnengart, A., Szalas, A.: A fixpoint approach to second-order quantifier elimination with applications to correspondence theory. Logic at Work: Essays Dedicated to the Memory of Helena Rasiowa (1999) **24** (01 1998)
34. Nonnengart, A., Weidenbach, C.: Computing small clause normal forms. In: Handbook of Automated Reasoning, pp. 335 – 367. North-Holland (2001)
35. Sakr, M., Schmidt, R.A.: Fine-grained forgetting for the description logic $\mathcal{ALC}$ (2021), http://www.cs.man.ac.uk/∼schmidt/publications/SakrSchmidt21a.pdf, Manuscript, submitted for publication
36. Schmidt, R.A.: The Ackermann approach for modal logic, correspondence theory and second-order reduction. Journal of Applied Logic **10**(1), 52–74 (2012)
37. Spackman, K.: Managing clinical terminology hierarchies using algorithmic calculation of subsumption: Experience with snomed-rt. Journal of the American Medical Informatics Association (01 2000)
38. Tarski, A.: A lattice-theoretical fixpoint theorem and its applications. Pacific journal of Mathematics **5**(2), 285–309 (1955)
39. Wernhard, C.: Projection and scope-determined circumscription. Journal of Symbolic Computation **47**, 1089–1108 (sep 2012)
40. Wernhard, C.: Application patterns of projection/forgetting. In: Workshop on Interpolation: From Proofs to Applications, iPRA 2014 (2014)
41. Zhang, Y., Zhou, Y.: Forgetting Revisited. Twelfth International Conference on the Principles of Knowledge Representation and Reasoning (apr 2010)
42. Zhao, Y., Schmidt, R.A.: Concept forgetting in $\mathcal{ALCOI}$-ontologies using an Ackermann approach. In: The Semantic Web, 14th International Semantic Web Conference. Lecture Notes in Computer Science, vol. 9366, pp. 587–602. Springer (2015)
43. Zhao, Y., Schmidt, R.A.: Forgetting concept and role symbols in $\mathcal{ALCOIH}\mu^+(\nabla, \sqcap)$-ontologies. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. pp. 1345–1352. AAAI Press/IJCAI (2016)
44. Zhao, Y., Schmidt, R.A.: Role forgetting for $\mathcal{ALCOQH}(\nabla)$-ontologies using an Ackermann approach. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence. pp. 1354–1361. AAAI Press/IJCAI (2017)
45. Zhao, Y., Schmidt, R.A.: On concept forgetting in description logics with qualified number restrictions. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence. pp. 1984–1990. AAAI Press/IJCAI (2018)
46. Zhao, Y., Schmidt, R.A.: FAME(Q): An automated tool for forgetting in description logics with qualified number restrictions. In: Automated Deduction—CADE-27. Lecture Notes in Artificial Intelligence, vol. 11716. Springer (2019)