Fine-Grained Forgetting for the Description Logic ALC

Mostafa Sakr, Renate A. Schmidt

The University of Manchester, Manchester, UK

Abstract

Forgetting is an important ontology extraction technique. A variant of forgetting which has received significant attention in the literature is *deductive forgetting*. While *deductive forgetting* (or *uniform interpolation*) is attractive as it generates the forgetting view in a language with the same complexity as the language of the original ontology, it is known that with a slightly extended target language using definer symbols more information can be preserved. In this paper, we study *deductive forgetting* of concept names with the aim of understanding the unpreserved information. We present a system that performs *deductive forgetting* and produces a set Δ of axioms representing the unpreserved information in the forgetting view. Our system allows a new fine-grained ontology extraction process that gives the user the option to enhance the informativeness of the deductive forgetting view by appending to it axioms from Δ.

1. Introduction

Forgetting is an important ontology extraction technique. It eliminates from an ontology a given subset of its vocabulary. The result is a focused ontology, called *forgetting view*, which preserves the content of the ontology relative to the non-forgotten vocabulary. Forgetting offers solutions for many applications such as: *computing logical difference* [1], *information hiding* [2], *abduction* [3], *resolving conflicts* [4], *relevance* [5, 6, 7], and *forgetting actions in planning* [8].

A variant of forgetting that has been studied in the literature is *deductive forgetting* (or *uniform interpolation*). Given an $\mathcal{ALC}$ ontology and a forgetting signature, deductive forgetting produces a view of the ontology which preserves only the information expressible in $\mathcal{ALC}$ over the non-forgetting signature [9, 10, 11]. Deductive forgetting is however not precise, because information expressible with more expressivity may not be preserved. Yet, deductive forgetting remains an appealing variant of forgetting because when performed on $\mathcal{ALC}$ ontologies, the generated forgetting views are (infinitely) representable in $\mathcal{ALC}$, or finitely representable if fixpoint operators are allowed [12, 13].

When a deductive view is computed, the following questions arise: *Does the deductive view preserve all information of the non-forgotten vocabulary? If not, what information is not preserved and what does this information represent? Can some of this information be partially preserved, i.e., can a view more informative than the deductive forgetting view be computed?* In this paper, we

DL 2022: 35th International Workshop on Description Logics, August 7–10, 2022, Haifa, Israel
Mostafa.Sakr@manchester.ac.uk (M. Sakr); Renate.Schmidt@manchester.ac.uk (R. A. Schmidt)
0000-0003-4296-5831 (M. Sakr); 0000-0002-6673-3333 (R. A. Schmidt)

aim to address these questions, gain a better understanding of the information not preserved by deductive forgetting, and provide a practical tool to compute this information. We focus on concept forgetting for ontologies in the description logic $\mathcal{ALC}$, the basic logic in the family of expressive description logics [14].

The main contribution is a novel forgetting method. (1) The method converts the input ontology into an intermediate ontology in which the forgetting signature has been eliminated. This intermediate ontology is semantically equivalent to the input ontology with respect to the non-forgotten vocabulary. That is, the reducts of the models of both ontologies to the non-forgotten vocabulary coincide. The intermediate ontology may use foreign concept symbols, or definers, to represent subsets of role successors. (2) The method obtains from the intermediate ontology two sets $\mathcal{O}^{red}$ and Δ of axioms. The set $\mathcal{O}^{red}$ approximates the deductive view by allowing the use of foreign symbols. We present a method to eliminate these foreign symbols from $\mathcal{O}^{red}$ and obtain the final deductive view in $\mathcal{ALC}$. Complete elimination of the foreign symbols may not however succeed when the deductive view does not exist finitely in $\mathcal{ALC}$ due to cycles occurring over forgetting symbols. The set Δ represents the *information difference* between the intermediate ontology and $\mathcal{O}^{red}$. If all foreign symbols are successfully eliminated from $\mathcal{O}^{red}$, then Δ also represents the information difference between the input ontology and the deductive view.

Several benefits are obtained from our forgetting method. (1) In two different evaluations an implementation of our forgetting method was compared against the state-of-the-art deductive forgetting tool Lethe [15]. Our implementation was found to be faster than Lethe. Our analysis shows that this improvement can be attributed to a novelty of the language of the intermediate ontology as it allows for avoiding time-consuming operations performed by Lethe. (2) By inspecting the set Δ, we now understand the difference between the input ontology and the deductive view is information on the conjunctions of different subsets of role successors. (3) An empty Δ indicates that the deductive view coincides semantically with the input ontology with respect to the non-forgotten vocabulary. (4) By incrementing $\mathcal{O}^{red}$ with axioms from Δ, our method allows for a fine-grained forgetting framework where views of the original ontology that are more informative than the deductive view can be obtained based on user requirements.

All proofs are provided in the long version https://github.com/e73898ms/FineGrainedForgetting.

2. History and Related Work

Forgetting can be traced back to Boole who referred to it as *elimination of the middle terms*. In propositional logic, it was studied in relation to *relevance, independence,* and *variable elimination* [6, 7]. A variant of forgetting which preserves semantic equivalences is *semantic forgetting*. In the context of first-order logic (FOL), semantic forgetting was viewed as a *second-order quantifier elimination problem* [16, 17] finding that the semantic view of a FO theory is not in general expressible in FOL but is always expressible in second-order logic (SOL). A way to view the foreign symbols in the intermediate ontology created by our method is as second-order existentially quantified concept symbols. This is because they are used to represent subsets of role successors whose first-order definition, as we show, cannot be computed in general.

Therefore, our intermediate ontology adheres to results in the literature [9, 16], and can be viewed as an approximation to the semantic view of the input ontology. We show that standard reasoning operations can be performed on the intermediate ontology using standard $\mathcal{ALC}$ reasoning methods.

Deductive forgetting was considered in [18] under the name *weak forgetting*. The proposal in [18] builds on previous work in modal logics which views deductive forgetting as *uniform interpolation* [19, 20], i.e., forgetting a signature $\mathcal{F}$ from an ontology $\mathcal{O}$ is equivalent to computing the uniform interpolant over the remaining vocabulary of $\mathcal{O}$ after excluding $\mathcal{F}$. This allows characterizing the relationship between the original ontology and the deductive view in terms of *bisimulation* over the non-forgotten symbols [10, 9, 21]. Deductive and semantic forgetting are also closely related to the notions of *concept inseparability* and *model inseparability* [22, 23, 24, 25]. Several deductive forgetting methods were proposed in [13, 20, 1].

Deciding the existence of a finitely representable deductive view is 2ExpTime, and its size is, at most, triple exponential in the size of the original ontology [10]. For $\mathcal{ALC}$ ontologies, the deductive view can be captured, possibly infinitely, in $\mathcal{ALC}$. Infinite forgetting views occur when cycles over some forgetting symbols exist [13]. In this case, finite representations may be approximated by fixpoint operators [12, 13, 26, 27] or by using foreign symbols to witness these cycles [13]. The latter representation can be converted to the former [13].

3. Basic Definitions

Let N_c, N_r be two disjoint sets of concept symbols and role symbols. Concepts in $\mathcal{ALC}$ are of the following forms: $\bot \mid A \mid \neg C \mid C \sqcap D \mid \exists r.C$ where $A \in N_c, r \in N_r$ and C and D are $\mathcal{ALC}$ concepts. We also allow the following abbreviations: $\top \equiv \neg\bot, \forall r.C \equiv \neg\exists r.\neg C, C \sqcup D \equiv \neg(\neg C \sqcap \neg D)$. An interpretation in $\mathcal{ALC}$ is a pair $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ where the domain $\Delta^{\mathcal{I}}$ is a nonempty set and $\cdot^{\mathcal{I}}$ 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 $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The language constructs are interpreted as follows: $\bot^{\mathcal{I}} := \emptyset, \quad (\neg C)^{\mathcal{I}} := \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}, (C \sqcap D)^{\mathcal{I}} := C^{\mathcal{I}} \cap D^{\mathcal{I}}, (\exists r.C)^{\mathcal{I}} := \{x \in \Delta^{\mathcal{I}} | \exists y : (x, y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$.

A TBox, or an ontology, is a set of axioms of the form $C \sqsubseteq D$, where C and D are concepts. $\mathcal{I}$ is model of an ontology $\mathcal{O}$ if all axioms $C \sqsubseteq D \in \mathcal{O}$ are true in $\mathcal{I}$, in symbols $\mathcal{I} \models C \sqsubseteq D$. And, $\mathcal{I} \models C \sqsubseteq D$ if and only if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. We say that $C \sqsubseteq D$ is satisfiable with respect to $\mathcal{O}$ if and only if $\mathcal{I} \models C \sqsubseteq D$ for some model $\mathcal{I}$ of $\mathcal{O}$. We also say that $C \sqsubseteq D$ is a consequence of $\mathcal{O}$, in symbols $\mathcal{O} \models C \sqsubseteq D$, if and only if $\mathcal{I} \models C \sqsubseteq D$ for every model $\mathcal{I}$ of $\mathcal{O}$.

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

Definition 1. *Two models $\mathcal{I}$ and $\mathcal{J}$ Σ-*coincide *iff $\Delta^{\mathcal{I}} = \Delta^{\mathcal{J}}$ and $p^{\mathcal{I}} = p^{\mathcal{J}}$ for every concept or role symbol $p \in \Sigma$.*

Definition 2. *Let $\mathcal{O}_1$ and $\mathcal{O}_2$ be two ontologies and Σ a set of symbols where $\Sigma \subseteq N_c \cup N_r$. We say $\mathcal{O}_1$ and $\mathcal{O}_2$ are* semantically Σ-equivalent*, in symbols $\mathcal{O}_1 \equiv_{\Sigma}^{\mathcal{M}} \mathcal{O}_2$, iff for every model $\mathcal{I}_1$ of $\mathcal{O}_1$ there is a model $\mathcal{I}_2$ of $\mathcal{O}_2$, and vice versa, such that $\mathcal{I}_1$ and $\mathcal{I}_2$ Σ-coincide.*

Resolution (Res)

$$\frac{C_1 \sqcup A \quad C_2 \sqcup \neg A}{C_1 \sqcup C_2}$$

where A is a forgetting symbol and C_1, C_2 are general concept expressions.

Figure 1: Binary resolution rule

Definition 3. *Let $\mathcal{O}_1$ and $\mathcal{O}_2$ be two $\mathcal{ALC}$ ontologies, and let Σ a set of symbols where $\Sigma \subseteq N_c \cup N_r$. We say $\mathcal{O}_1$ and $\mathcal{O}_2$ are* deductively Σ-equivalent*, in symbols $\mathcal{O}_1 \equiv^C_\Sigma \mathcal{O}_2$, iff for every $\mathcal{ALC}$ concept inclusion α, where $sig(\alpha) \subseteq \Sigma$, we have $\mathcal{O}_1 \models \alpha$ iff $\mathcal{O}_2 \models \alpha$.*

Deductive forgetting is defined using deductive equivalence [10].

Definition 4. *Let $\mathcal{O}$ be an $\mathcal{ALC}$ ontology, and let $\mathcal{F} \subseteq sig(\mathcal{O}) \cap N_c$ be a forgetting signature. An ontology $\mathcal{V}$ is a* deductive forgetting view *of $\mathcal{O}$ w.r.t. $\mathcal{F}$ iff $sig(\mathcal{V}) \subseteq sig(\mathcal{O}) \backslash \mathcal{F}$, and $\mathcal{O} \equiv^C_{sig(\mathcal{O}) \backslash \mathcal{F}} \mathcal{V}$.*

4. Computing the Intermediate Ontology

The first stage of our method is to compute the intermediate ontology $\mathcal{O}^{int}$ of the input ontology $\mathcal{O}$ w.r.t. the given forgetting signature $\mathcal{F}$. The method applies resolution to the input ontology written in clausal form. $\mathcal{O}^{clausal}$ is computed by: (1) converting $\mathcal{O}$ into negation normal form (NNF), with negation applied only to concept names, (2) miniscoping, i.e., replacing $\exists r.C \sqcup \exists r.D$ with the semantically equivalent $\exists r.(C \sqcup D)$, and $\forall r.C \sqcap \forall r.D$ with the semantically equivalent $\forall r.(C \sqcap D)$, (3) applying structural transformations to extract the formulas under role restriction that contain the forgetting symbols by introducing fresh concept symbols (called definers) [28], and (4) converting the result to conjunctive normal form (CNF).

Example 1. *Consider the axiom $A \sqsubseteq \exists r.(B \sqcap C)$ where B is a forgetting symbol. It is first converted to NNF by eliminating the connective $\sqsubseteq$, giving $S_1 = \{\neg A \sqcup \exists r.(B \sqcap C)\}$. Structural transformation is applied to extract $B \sqcap C$, giving $S_2 = \{\neg A \sqcup \exists r.D_1, \neg D_1 \sqcup (B \sqcap C)\}$ where $D_1 \in N_d$ is a definer symbol. Finally, S_2 is converted to CNF, giving $S_3 = \{\neg A \sqcup \exists r.D_1, \neg D_1 \sqcup B, \neg D_1 \sqcup C\}$.*

The forgetting symbols in $\mathcal{F}$ are then eliminated from $\mathcal{O}^{clausal}$ by iteratively eliminating them using the *Resolution* rule in Figure 1. When all possible resolution inferences have been performed on a concept symbol in $\mathcal{F}$, clauses that contain this concept symbol are removed in a *purity deletion* step. Additionally, the following operations are applied eagerly: (1) Tautology deletion: clauses of the form $C \sqcup \neg C \sqcup E$ are deleted, where C and E are $\mathcal{ALC}$ concepts. (2) Purification: if a forgetting symbol A occurs only positively or only negatively in $\mathcal{O}$, then A is replaced everywhere by $\top$ and $\bot$ respectively. Assume $\mathcal{O}^{int}$ is the set of clauses that remain.

Example 2. *Let* $\mathcal{O} = \{A \sqsubseteq \forall r.B \sqcap \forall s.\neg B, G \sqsubseteq \exists r.(\neg B \sqcup C), B \sqsubseteq H\}$*, and* $\mathcal{F} = \{B\}$*. The method starts by generating* $\mathcal{O}^{clausal} = \{\neg A \sqcup \forall r.D_1, \neg A \sqcup \forall s.D_2, \neg G \sqcup \exists r.D_3, \neg D_1 \sqcup B, \neg D_2 \sqcup \neg B, \neg D_3 \sqcup \neg B \sqcup C, \neg B \sqcup H\}$*, where* D_1, D_2*, and* D_3 *are fresh definers. Then, it resolves on the concept symbol* B *using the* Resolution *rule in Figure 1 generating additionally the clauses* $\{\neg D_1 \sqcup \neg D_2, \neg D_1 \sqcup \neg D_3 \sqcup C, \neg D_1 \sqcup H\}$*. Finally, the clauses* $\{\neg D_1 \sqcup B, \neg D_2 \sqcup \neg B, \neg D_3 \sqcup \neg B \sqcup C, \neg B \sqcup H\}$ *are removed by purity deletion. So the intermediate ontology* $\mathcal{O}^{int}$ *is* $\{\neg A \sqcup \forall r.D_1, \neg A \sqcup \forall s.D_2, \neg G \sqcup \exists r.D_3, \neg D_1 \sqcup \neg D_2, \neg D_1 \sqcup \neg D_3 \sqcup C, \neg D_1 \sqcup H\}$*.*

Theorem 1. $\mathcal{O} \equiv^{\mathcal{M}}_{sig(\mathcal{O})\setminus\mathcal{F}} \mathcal{O}^{int}$.

Theorem 2. *The size of* $\mathcal{O}^{int}$ *is in the worst case exponential in the size of the given ontology* $\mathcal{O}$ *and double exponential in the number of forgetting symbols.*

Definers are used in the intermediate ontology $\mathcal{O}^{int}$ to represent subsets of role successors. Precise definitions of definers cannot always be given without knowledge of the forgetting symbols. For instance in Example 1, D_1 is interpreted as $D_1^{\mathcal{I}} = \{y \in B^{\mathcal{I}} \cap C^{\mathcal{I}} \mid (x, y) \in r^{\mathcal{I}}, x \in A^{\mathcal{I}}\}$ where $\mathcal{I}$ is a model of S. Since the user of the forgetting view may not be aware of B, a definition of D_1 in terms of B is not conveyed in $\mathcal{O}^{int}$. Thus, definers may be viewed as second-order existentially quantified concept symbols because they represent some subsets of the domain whose definitions cannot be precisely captured. One can observe that $\mathcal{O}^{int}$ can be viewed as an approximation to the semantic forgetting view of $\mathcal{O}$ with respect to $\mathcal{F}$.

5. Extracting Δ and $\mathcal{O}^{red}$

The second stage of the method is to obtain the sets Δ and $\mathcal{O}^{red}$ from $\mathcal{O}^{int}$, where $\mathcal{O}^{red}$ is an ontology semantically equivalent to the deductive view and Δ is the information difference between $\mathcal{O}^{int}$ and $\mathcal{O}^{red}$. The *Reduction* rule in Figure 2 removes from $\mathcal{O}^{int}$ the clauses with two or more negative definers. These clauses constitute the set Δ.

The *Role Propagation* rule in Figure 2 computes the $\mathcal{ALC}$ consequences that are otherwise lost when removing the clauses of Δ from $\mathcal{O}^{int}$ by the *Reduction* rule. Therefore, we require it to be applied before removing these clauses. The premises of the *Role Propagation* rule start with the clause $P_0 \sqcup C_0$, where P_0 takes the form $\neg D_0 \sqcup \neg D_1 \sqcup \cdots \sqcup \neg D_n$. The second premise is a set of clauses $P_j \sqcup C_j$. Here, the concepts P_j takes the same form as the concept P_0, i.e., is a disjunction of negative definers, but also Definers$(P_j) \subseteq$ Definers(P_0) where Definers(P) denotes the set of definer symbols in $sig(P)$. The intuition here is that $P_j \sqsubseteq P_0$. Therefore, every domain element that is not in the interpretation of P_0, consequently P_j, must be in the interpretation of C_0 and C_j. The clauses in the third and the fourth premises take the same form, except that existential role restriction is only allowed in the third premise. By the third and fourth premises, every domain element must be in the interpretation of $\bigsqcup_{i=0}^{n} E_i$ or $\mathcal{Q}r.(\bigsqcap_{i=0}^{n} D_i)$. But the latter can be rewritten as $\mathcal{Q}r.\neg P_0$, which is subsumed by $\mathcal{Q}r.(\bigsqcap_{j=0}^{m} C_j)$ as concluded by the rule.

Example 3. *Continuing with Example 2, the* Role Propagation *rule applies with its four premises being:*

1. $P_0 \sqcup C_0 = \neg D_1 \sqcup \neg D_3 \sqcup C$

Role Propagation

$$\frac{P_0 \sqcup C_0,\ \bigcup_{j=1}^{m} \{P_j \sqcup C_j\}, E_0 \sqcup \mathcal{Q}r.D_0,\ \bigcup_{i=1}^{n} \{E_i \sqcup \forall r.D_i\}}{(\bigsqcup_{i=0}^{n} E_i) \sqcup \mathcal{Q}r.(\bigsqcap_{j=0}^{m} C_j)}$$

where $P_0 = \bigsqcup_{i=0}^{n} \neg D_i$, P_j is any sub-concept of P_0, $\mathcal{Q} \in \{\exists, \forall\}$, and C_0 and C_j do not contain a definer.

Reduction

$$\frac{\mathcal{O} \cup \{\neg D_1 \sqcup ... \sqcup \neg D_n \sqcup C\}}{\mathcal{O}}$$

where C is a concept expression that does not contain a negative definer, $D_1, ..., D_n$ are definer symbols, and $n \geq 2$.

Figure 2: $\mathcal{ALC}$ reduction rules.

2. $\bigcup_{j=1}^{m} \{P_j \sqcup C_j\} = \{\neg D_1 \sqcup H\}$
3. $E_0 \sqcup \mathcal{Q}r.D = \neg G \sqcup \exists r.D_3$
4. $\bigcup_{i=1}^{n} \{E_i \sqcup \forall r.D_i\} = \{\neg A \sqcup \forall r.D_1\}$

The conclusion is $\neg A \sqcup \neg G \sqcup \exists r.(C \sqcap H)$*. Note that the generated conclusion preserves information that would otherwise be lost when the clause* $\neg D_1 \sqcup \neg D_3 \sqcup C$ *is removed by the* Reduction *rule.*

After the *Role Propagation* rule has been exhaustively applied, the clauses of Δ are removed. The remaining clauses constitute $\mathcal{O}^{red}$.

Denote by $\mathcal{O}^{rp}$ the ontology obtained from $\mathcal{O}^{int}$ by applying the *Role Propagation* rule.

Example 4. *Continuing with Example 2. We have* $\mathcal{O}^{rp} = \mathcal{O}^{int} \cup \{\neg A \sqcup \neg G \sqcup \exists r.(C \sqcap H)\}$ *where the axiom on the right of the union operator is the conclusion of the* Role Propagation *rule obtained in Example 3. We also have* $\Delta = \{\neg D_1 \sqcup \neg D_2, \neg D_1 \sqcup \neg D_3 \sqcup C\}$*, and* $\mathcal{O}^{red} = \{\neg A \sqcup \forall r.D_1, \neg A \sqcup \forall s.D_2, \neg G \sqcup \exists r.D_3, \neg D_1 \sqcup H, \neg A \sqcup \neg G \sqcup \exists r.(C \sqcap H)\}$.

Theorem 3. $\mathcal{O}^{rp} \equiv^{C}_{sig(\mathcal{O})\setminus\mathcal{F}} \mathcal{O}^{red}$*.*

Theorem 3 proves a main contribution of the paper. First, observe that $\mathcal{O}^{rp} \equiv^{\mathcal{M}}_{sig(\mathcal{O})\setminus\mathcal{F}} \mathcal{O}^{int}$. It follows from this observation, and Theorems 1 and 3 that $\mathcal{O} \equiv^{C}_{sig(\mathcal{O})\setminus\mathcal{F}} \mathcal{O}^{red}$. Therefore, if $\mathcal{O}^{ui}$ is a deductive view of $\mathcal{O}^{red}$ with respect to $sig(\mathcal{O})\setminus\mathcal{F}$, then $\mathcal{O}^{red}$ is deductively equivalent to $\mathcal{O}^{ui}$ with respect to $sig(\mathcal{O})\setminus\mathcal{F}$. We shall strengthen this in the next section and show that if no cycles occur in $\mathcal{O}^{red}$ then it is semantically equivalent to the deductive view with respect to $sig(\mathcal{O})\setminus\mathcal{F}$.

Second, observe that $\mathcal{O}^{rp} = \mathcal{O}^{red} \cup \Delta$. Additionally, the clauses in Δ have been generated in $\mathcal{O}^{int}$ by resolution inferences over the forgetting symbols, and the premises of these inferences were removed by purity deletion. Therefore, we find in general that $\mathcal{O}^{red} \not\models \Delta$. This implies that Δ can be viewed as representing the information difference between $\mathcal{O}^{rp}$ and $\mathcal{O}^{red}$. Altogether we therefore conclude that Δ can be viewed as the information difference between $\mathcal{O}$ and the deductive view with respect to $sig(\mathcal{O})\backslash\mathcal{F}$.

We end this section with a discussion on the extracted set Δ which consists of clauses of the format $\neg D_1 \sqcup \cdots \sqcup \neg D_n \sqcup F$ where $n \geq 2$, or in axiom form, $D_1 \sqcap \cdots \sqcap D_n \sqsubseteq F$. Since we introduced the definer symbols to represent subsets of role successors, these clauses can be understood as information on the conjunctions of different subsets of role successors. For instance, in Example 2, the clause $\neg D_1 \sqcup \neg D_2 \in \Delta$ specifies the constraint that the subset of r-successors and the subset of s-successors of domain elements in the interpretation of A are disjoint. It was not a coincidence that we introduced definer symbols to represent subsets of role successors. Using them in this way and introducing them via structural transformation forces the clauses in Δ to be explicit members of $\mathcal{O}^{int}$ which simplifies their extraction, giving us a representation of the difference between $\mathcal{O}$ and the deductive view.

6. Eliminating the Definer Symbols

We identified Δ as the axioms that contain two or more negative definers. $\mathcal{O}^{int}$ and $\mathcal{O}^{red}$ may contain definer symbols that appear negatively in clauses where no other negative definer is present. These definers can be eliminated safely while preserving the interpretations of the non-forgotten vocabulary. For this, we use the *Definer Elimination* rule in Figure 3.

The side conditions of the *Definer Elimination* rule exclude the elimination of definers that may appear both positively and negatively in a clause. We call such definer symbols *cyclic definers*. The existence of cyclic definers signifies cycles in the original ontology over some forgetting symbols. In this case the deductive view may not exist as it requires an infinite representation.

An approach to eliminate cyclic definers and obtain a finite approximation of the deductive view is using fixpoint operators [12]. As an alternative, cyclic definers can be left in the deductive view as witnesses of these cycles [13]. We find this the best option because it defers the decision of a suitable representation to a later stage.

For clauses that contain only one negative definer symbol, possibly with other positive definers, the *Definer Elimination* rule in Figure 3 is applied exhaustively. The rule replaces the definer symbol D with its super-concept $C_1 \sqcap \ldots \sqcap C_n$. Note that, in the *Definer Elimination* rule, C may be $\bot$. Besides the *Definer Elimination* rule, we also eagerly apply the *Tautology Deletion* and the *Purification* rules (see Section 4).

Example 5. *Continuing with Example 4, $\mathcal{V}$ is extracted from $\mathcal{O}^{red}$ as follows:*

1. *The definers D_2 and D_3 are eliminated using* Purification. *Since D_2 and D_3 appear only positively in $\mathcal{O}^{red}$, they are purified by replacing them with $\top$ which gives $\{\neg A \sqcup \forall r.D_1, \neg A \sqcup \forall s.\top, \neg G \sqcup \exists r.\top, \neg D_1 \sqcup H, \neg A \sqcup \neg G \sqcup \exists r.(C \sqcap H)\}$. As $\forall s.\top$ evaluates to $\top$, the result can be simplified further to $\{\neg A \sqcup \forall r.D_1, \neg G \sqcup \exists r.\top, \neg D_1 \sqcup H, \neg A \sqcup \neg G \sqcup \exists r.(C \sqcap H)\}$.*

Definer Elimination

$$\frac{\mathcal{O} \cup \{\neg D \sqcup C_1, ..., \neg D \sqcup C_n\}}{\mathcal{O}[D/C]}$$

where $C = \sqcap_{i=1}^{n} C_i$ and $D \notin sig(C)$, C does not contain any negative definers, and $\mathcal{O}$ does not contain D negatively.

Figure 3: Definer elimination rule

2. *The definer D_1 is eliminated by the* Definer Elimination *rule in Figure 3 giving $\{\neg A \sqcup \forall r.H, \neg G \sqcup \exists r.\top, \neg H \sqcup H, \neg A \sqcup \neg G \sqcup \exists r.(C \sqcap H)\}$. The clause $\neg H \sqcup H$ is then eliminated by* Tautology Deletion *giving $\mathcal{V} = \{\neg A \sqcup \forall r.H, \neg G \sqcup \exists r.\top \neg A \sqcup \neg G \sqcup \exists r.(C \sqcap H)\}$*

Theorem 4. *Let $\mathcal{V}$ be generated from $\mathcal{O}^{red}$ by applying the Definer Elimination rule from Figure 3 exhaustively. Then, (1) $\mathcal{O}^{red} \equiv^{\mathcal{M}}_{sig(\mathcal{O})\setminus\mathcal{F}} \mathcal{V}$; and (2) if $sig(\mathcal{V}) \cap N_d = \emptyset$ then $\mathcal{V}$ is a deductive forgetting view of $\mathcal{O}$ w.r.t. $\mathcal{F}$.*

7. Computing More Informative Forgetting Views

Our forgetting method allows customizing the informativeness of the final forgetting view according to user requirements, which reveals a spectrum of forgetting views that are more informative than the deductive view and at most as informative as the intermediate ontology. This can be done by overriding the *Reduction* rule as illustrated in the following example.

Example 6. *Consider $\mathcal{O}^{int}$ from Example 2. Applying the rules in Figure 2 gives $\mathcal{O}^{red}$ and Δ from Example 4. We may increase the informativeness of the final forgetting view by overriding the* Reduction *rule. We describe three different forgetting views that can be generated in this way.*

1. *If we want to preserve all the information about the r-successors of A, then we override the* Reduction *rule to retain the clauses where D_1 occurs. In this case, $\Delta_1 = \emptyset$ and the final forgetting view will be $\mathcal{O}_1^{red} = \mathcal{V}_1 = \{\neg A \sqcup \forall r.D_1, \neg A \sqcup \forall s.D_2, \neg G \sqcup \exists r.D_3, \neg D_1 \sqcup \neg D_2, \neg D_1 \sqcup \neg D_3 \sqcup C, \neg A \sqcup \neg G \sqcup \exists r.C\}$.*
2. *If we want to preserve the information about the r-successors of A in relation to the r-successors of G, we override the* Reduction *rule to retain the clauses where both D_1 and D_3 occur. That is, $\Delta_2 = \{\neg D_1 \sqcup \neg D_2\}$ and $\neg D_1 \sqcup \neg D_3 \sqcup C \notin \Delta$. Then, $\mathcal{O}_2^{red} = \{\neg A \sqcup \forall r.D_1, \neg A \sqcup \forall s.D_2, \neg G \sqcup \exists r.D_3, \neg D_1 \sqcup \neg D_3 \sqcup C, \neg A \sqcup \neg G \sqcup \exists r.C\}$, and $\mathcal{V}_2 = \{\neg A \sqcup \forall r.D_1, \neg G \sqcup \exists r.D_3, \neg D_1 \sqcup \neg D_3 \sqcup C, \neg A \sqcup \neg G \sqcup \exists r.C\}$ with D_2 purified away.*
3. *If we are interested in the relation between the r and s successors of A, then we override the* Reduction *rule to remove $\neg D_1 \sqcup \neg D_3 \sqcup C$ but not $\neg D_1 \sqcup \neg D_2$. That is $\Delta_3 = \{\neg D_1 \sqcup \neg D_3 \sqcup C\}$. Consequently, we get $\mathcal{O}_3^{red} = \{\neg A \sqcup \forall r.D_1, \neg A \sqcup \forall s.D_2, \neg G \sqcup \exists r.D_3, \neg A \sqcup \neg G \sqcup \exists r.C, \neg D_1 \sqcup \neg D_2\}$. The final forgetting view then becomes $\mathcal{V}_3 = \{\neg A \sqcup \forall r.D_1, \neg A \sqcup \forall s.D_2, \neg G \sqcup \exists r.\top, \neg A \sqcup \neg G \sqcup \exists r.C, \neg D_1 \sqcup \neg D_2\}$ with D_3 purified away.*

Observe that in $\mathcal{V}_1$ and $\mathcal{V}_2$ the clause $\neg A \sqcup \neg G \sqcup \exists r.C$ is redundant. We can eliminate this redundancy by applying the *Role Propagation* rule only when a premise of the rule occurs in Δ.

Since the final forgetting views may use definers, the following question may be asked: *Are definers in forgetting views limiting?* We argue that since definers are existentially quantified and the forgetting view is expressed using $\mathcal{ALC}$ syntax, standard reasoning tasks such as *satisfiability checking*, and *query answering*, can be performed with respect to the non-forgotten vocabulary using the existing $\mathcal{ALC}$ methods. The following example explains the idea.

Example 7. *Let ontology $\mathcal{O} = \{A_1 \sqsubseteq \forall r.B, A_2 \sqsubseteq \forall r.\neg B\}$ be an ontology, and $\mathcal{O}^{int} = \{\neg A_1 \sqcup \forall r.D_1, \neg A_2 \sqcup \forall r.D_2, \neg D_1 \sqcup \neg D_2 \sqsubseteq \bot\}$ the intermediate ontology of $\mathcal{O}$ with respect to $\mathcal{F} = \{B\}$ where D_1 and D_2 are definers. Assume $\Delta = \emptyset$, then $\mathcal{O}^{int}$ is the final forgetting view. Both $\mathcal{O}$ and $\mathcal{O}^{int}$ model 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 we additionally have a database $\mathcal{A} = \{A_1(a_1), A_2(a_2), r(a_1, b)\}$, and we want to prove the unsatisfiability of $r(a_2, b)$ with respect to the knowledge base consisting of $\mathcal{O}$ and $\mathcal{A}$. This can be done by using a standard $\mathcal{ALC}$ reasoner to show that $\mathcal{O}, \mathcal{A}, r(a_2, b) \models \bot$. Replacing $\mathcal{O}$ with $\mathcal{O}^{int}$, would still prove the unsatisfiability of $r(a_2, b)$. Moreover, the reasoner does not require the full interpretation of D_1 and D_2 to prove that $\mathcal{O}^{int}, \mathcal{A}, r(a_2, b) \models \bot$.*

8. Evaluation

We implemented a prototype of our method based on *Java 12* and the *OWL API 5.1.11*. We refer to our prototype as *SeD*. We used a random corpus of 50 ontologies from the NCBO Bioportal repository to perform the evaluation. Details of the corpus are given in the long version.

We performed two evaluations, each corresponding to a different selection of the forgetting signature $\mathcal{F}$. *Evaluation 1* selected $\mathcal{F}$ as a segment of the N_c sorted by name (recall that N_c is the set of concept names of the input ontology), so $\mathcal{F}$ contained related concept names. E.g., 'Abdomen' and 'Abdomen-pain' were likely to be together in $\mathcal{F}$ as they would be adjacent in the sorted N_c. The intuition was to simulate the use case of extracting the knowledge of a single topic, e.g., the *digestive system* from a large biomedical ontology.

Evaluation 2 selected the concept names that occurred most frequently under role restrictions, aiming for more definers being introduced in $\mathcal{O}^{int}$ and bigger Δ sets. The intuition was to simulate the worst case when the information difference between the intermediate ontology and the deductive views would be large, and the performance of our two-stage forgetting method would be expected to degrade.

In every evaluation and for each ontology in the corpus, three forgetting experiments were performed to forget 10%, 30%, and 50% of concept symbols in N_c giving a total of 150 experiments in each evaluation. Each experiment compared SeD and *Lethe* (version 2.11-0.026)[1] [15]. Lethe is an implementation of the deductive forgetting method in [13]. Each tool was allocated 2GB of memory and five hours time out. All experiments were run on a x64-based processor Intel(R) Core(TM) i5 CPU @ 2.7GHz with a 64-bit operating system (macOS Catalina 10.15.7).

[1] http://www.cs.man.ac.uk/~koopmanp/lethe/index.html

Table 1
Timeouts of SeD and Lethe

	Evaluation 1			Evaluation 2		
	10%	30%	50%	10%	30%	50%
SeD	2	3	5	2	3	3
Lethe	2	7	8	1	6	7

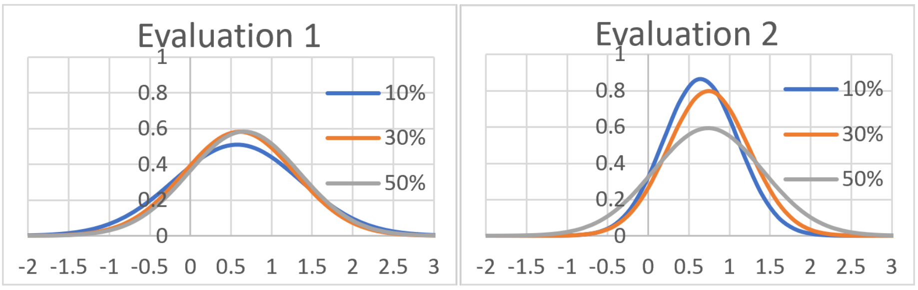

Figure 4: Normal Distribution of the *Gain* values. Range of X-axes is Average $\pm$ 3 Standard Deviations

Table 1 shows the timeouts of SeD and Lethe. SeD appeared to be more reliable than Lethe in both evaluations. Also, SeD was less affected by increasing the size of $\mathcal{F}$ from 10% to 30% to 50% of N_c, suggesting that SeD is more scalable to harder problems than Lethe.

Next we compared the execution times of SeD and Lethe to compute the deductive view. We first computed the time gained by using SeD over Lethe with the formula *Gain* = $(T_L - T_S)/T_L$, where T_L and T_S are the times consumed by Lethe and SeD respectively. Second, we computed the averages and standard deviations of the *Gain* values. In line with standard data analysis methods, *outliers* were excluded. These were experiments with extreme *Gain* values compared to the rest of the experiments. In *Evaluation 1* we excluded two experiments in the 10% setting and one in the 50% setting, whereas in *Evaluation 2* we excluded one experiment in the 10% setting and one from the 50% setting.

The averages and standard deviations in *Evaluation 1* were: (0.58, 0.78), (0.60, 0.68), and (0.66, 0.68) in the 10%, 30%, and 50% settings respectively. In *Evaluation 2* they were: (0.65, 0.46), (0.74, 0.50), and (0.74, 0.67) in the 10%, 30%, and 50% settings respectively. Figure 4 shows the normal distributions of the *Gain* values in the three settings in the two evaluations. The graphs reflect the attained positive averages stated above, and compare the gain values across the three settings in each evaluation, also allowing the two evaluations to be compared. Surprisingly, a better and more consistent performance was found in *Evaluation 2* over *Evaluation 1*, against our expectation that *Evaluation 2* would represent the worst case scenario for SeD. This is indicated by the higher peaks indicating higher probability of achieving the average *Gain*, and the narrower curves indicating less variation in the results.

The performance improvement happens due to the forgetting method itself not the corpus. While Lethe translates the input ontology to a clausal form that is similar to ours, it disallows clauses with two or more negative definers. To compensate for this restriction, Lethe introduces definer symbols as part of the forgetting calculus, and builds a subsumption hierarchy between the definers. This hierarchy forces extra resolution inferences to be performed. The following example illustrates dynamic introduction of definers in Lethe.

Example 8. *Let* $\mathcal{O} = \{A_1 \sqsubseteq \exists r.\neg B, A_2 \sqsubseteq \forall r.B\}$*, and* $\mathcal{F} = \{B\}$*. Lethe generates* $\mathcal{O}^{clausal}$ *as* $\{\neg A_1 \sqcup \exists r.D_1, \neg A_2 \sqcup \forall r.D_2, \neg D_1 \sqcup \neg B, \neg D_2 \sqcup B\}$*. Instead of resolving* B *directly to compute the clause* $\neg D_1 \sqcup \neg D_2$*, Lethe introduces a new definer* D_3 *and generates an intermediate ontology* $\mathcal{O}_1 = \mathcal{O}^{clausal} \cup \{\neg A_1 \sqcup \neg A_2 \sqcup \exists r.D_3, \neg D_3 \sqcup D_1, \neg D_3 \sqcup D_2\}$*. The last two clauses on* $\mathcal{O}_1$ *are resolved with* $\neg D_1 \sqcup \neg B$ *and* $\neg D_2 \sqcup B$ *to give* $\neg D_3 \sqcup \neg B$ *and* $\neg D_3 \sqcup B$ *which in turn are resolved together to give* $\neg D_3$*. All clauses where* B *occurs are then removed to give* $\mathcal{O}_2 = \{\neg A_1 \sqcup \exists r.D_1, \neg A_2 \sqcup \forall r.D_2, \neg A_1 \sqcup \neg A_2 \sqcup \exists r.D_3, \neg D_3\}$*. The definers* $D_1, D_2,$ *and* D_3 *are eliminated in a similar way to the method described in Section 6 giving* $\mathcal{V} = \{\neg A_1 \sqcup \exists r.\top, \neg A_1 \sqcup \neg A_2\}$*.*

Our normal form is more flexible as it allows several negative definers to appear in a clause, thus avoids the extra resolution inferences performed by Lethe.

We measured the size of the extracted Δ set. In *Evaluation 1*, Δ was on average 0.01%, 0.66%, and 18.3% of the size of the original ontology, in the 10%, 30%, and 50% settings respectively. In *Evaluation 2*, Δ was on average 0.23%, 0.88%, and 7.13% of the size of the original ontology, in the 10%, 30%, and 50% settings respectively.

We also measured the number of definers in Δ as a ratio to the forgetting signature. In *Evaluation 1*, they were on average 0.03%, 0.63%, and 1.5% in the 10%, 30%, and 50% settings respectively. In *Evaluation 2*, they were on average 0.04%, 0%, and 1.4% in the 10%, 30%, and 50% settings respectively. These ratios indicate that our fine-grained method was feasible since appending $\mathcal{O}^{red}$ with axioms from Δ introduced few definers relative to the size of $\mathcal{F}$.

Finally, we measured the size of the deductive view relative to the original ontology. In *Evaluation 1* it was on average 114%, 103%, and 117% of the size of the original ontology in the 10%, 30%, and 50% settings respectively. In *Evaluation 2*, it was on average 113%, 95%, and 103% of the size of the original ontology in the 10%, 30%, and 50% settings respectively.

9. Conclusions and Future Work

We presented a new forgetting method that performs deductive forgetting, and extracts a set Δ of axioms representing the information difference between the original ontology and the deductive view. Not only does this give a clearer understanding, in terms of the modelled information, on the difference between the input ontology and the deductive view, but also it allows a fine-grained forgetting system that gives control over the information modelled in the forgetting view. Empirical evaluation suggests that our forgetting method is faster than the state-of-the-art forgetting tool Lethe despite computing more information. Nevertheless, our evaluation suggested that appending the deductive forgetting view with information from Δ introduces few foreign symbols compared to the forgotten symbols. The final forgetting view therefore remains a compact extract of the original ontology for the use in applications.

Future work will study in greater depth the newly revealed spectrum of forgetting variants, and their intersections with other forgetting variants in the literature.

References

[1] M. Ludwig, B. Konev, Practical uniform interpolation and forgetting for $\mathcal{ALC}$ TBoxes with applications to logical difference, in: Proc. KR 2014, AAAI Press, 2014.

[2] B. C. Grau, B. Motik, Pushing the limits of reasoning over ontologies with hidden content, in: Proc. KR 2010, AAAI Press, 2010, pp. 214–224. URL: http://www.comlab.ox.ac.uk/people/Boris.Motik/pubs/gm10pushing-ibq.pdf.

[3] W. Del-Pinto, R. A. Schmidt, ABox abduction via forgetting in ALC, in: Proc. AAAI 2019, AAAI Press, 2019, pp. 2768–2775.

[4] J. Lang, P. Marquis, Reasoning under inconsistency: A forgetting-based approach, Artificial Intelligence 174 (2010) 799–823. URL: https://www.sciencedirect.com/science/article/pii/S0004370210000676. doi:https://doi.org/10.1016/j.artint.2010.04.023.

[5] D. Subramanian, M. R. Genesereth, The relevance of irrelevance., in: Proc. IJCAI, 1987.

[6] G. Lakemeyer, Relevance from an Epistemic Perspective, Artif. Intell. 97 (1997) 137–167. doi:10.1016/S0004-3702(97)00038-6.

[7] J. Lang, P. Liberatore, P. Marquis, Propositional independence: Formula-variable independence and forgetting, Artificial Intelligence Research 18 (2003) 391–443.

[8] E. Erdem, P. Ferraris, Forgetting Actions in Domain Descriptions, in: Proc. AAAI 2007, AAAI Press, 2007, pp. 409–414.

[9] Y. Zhang, Y. Zhou, Knowledge forgetting: Properties and applications, Artificial Intelligence 173 (2009) 1525–1537. doi:10.1016/j.artint.2009.07.005.

[10] C. Lutz, F. Wolter, Foundations for uniform interpolation and forgetting in expressive description logics, in: Proc. IJCAI 2011, 2011, p. 989–995.

[11] J. Delgrande, A Knowledge Level Account of Forgetting, Artificial Intelligence Research 60 (2017) 1165–1213. doi:10.1613/jair.5530.

[12] A. Nonnengart, A. Szalas, A fixpoint approach to second-order quantifier elimination with applications to correspondence theory, in: Logic at Work, Springer, 1998.

[13] P. Koopmann, R. A. Schmidt, Uniform interpolation of $\mathcal{ALC}$-ontologies using fixpoints, in: Proc. FroCoS 2013, volume 8152 of *LNAI*, Springer, 2013, pp. 87–102.

[14] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, P. F. Patel-Schneider, The Description Logic Handbook, Cambridge Univ. Press, 2003.

[15] P. Koopmann, R. A. Schmidt, Implementation and evaluation of forgetting in ALC-ontologies, in: Proc. WoMo 2013, volume 1081, CEUR-WS.org, 2013.

[16] F. Lin, R. Reiter, Forget it, in: AAAI Fall Symp. on Relevance, 1994, pp. 154–159.

[17] D. M. Gabbay, H. J. Ohlbach, Quantifier elimination in second-order predicate logic, in: Proc. KR 1992, Morgan Kaufmann, 1992, pp. 425–435.

[18] Y. Zhang, Y. Zhou, Forgetting revisited, in: Proc. KR, AAAI Press, 2010.

[19] S. Ghilardi, An algebraic theory of normal forms, Ann. Pure Appl. Logic 71 (1995) 189 – 245.

[20] A. Herzig, J. Mengin, Uniform interpolation by resolution in modal logic, in: Proc. JELIA 2008, volume 5293 LNAI, Springer, 2008, pp. 219–231.

[21] H. Ditmarsch, A. Herzig, J. Lang, P. Marquis, Introspective forgetting, in: Proc. AI 2008, Springer, 2008, pp. 18–29.

[22] B. Konev, D. Walther, F. Wolter, Forgetting and uniform interpolation in large-scale

description logic terminologies, in: Proc. IJCAI 2009, Morgan Kaufmann, 2009, p. 830–835.
[23] B. Konev, C. Lutz, D. Walther, F. Wolter, Model-theoretic inseparability and modularity of description logic ontologies, Artificial Intelligence 203 (2013) 66–103. doi:10.1016/j.artint.2013.07.004.
[24] E. Botoeva, B. Konev, C. Lutz, V. Ryzhikov, F. Wolter, M. Zakharyaschev, Inseparability and conservative extensions of description logic ontologies: A survey, in: Reasoning Web 2016, Springer, 2017, pp. 27–89. URL: https://doi.org/10.1007/978-3-319-49493-7_2. doi:10.1007/978-3-319-49493-7_2.
[25] C. Lutz, F. Wolter, Deciding inseparability and conservative extensions in the description logic $\mathcal{EL}$, Symbolic Computation 45 (2010) 194–228. doi:10.1016/j.jsc.2008.10.007.
[26] G. D'Agostino, M. Hollenberg, Uniform interpolation, automata and the modal μ-calculus, Logic Group Preprint Series 165 (1996).
[27] D. Calvanese, G. De Giacomo, M. Lenzerini, Reasoning in Expressive Description Logics with Fixpoints Based on Automata on Infinite Trees, in: Proc. IJCAI 1999, Morgan Kaufmann, 1999, pp. 84–89.
[28] A. Nonnengart, C. Weidenbach, Computing small clause normal forms, in: Handbook of Automated Reasoning, North-Holland, 2001, pp. 335 – 367.
[29] M. A. E. Dummett, E. J. Lemmon, Modal Logics Between S 4 and S 5, Mathematical Logic Quarterly 5 (1959) 250–264. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/malq.19590051405.
[30] P. Blackburn, J. van Benthem, F. Wolter, Handbook of modal logic, volume 3 of *Studies in logic and practical reasoning*, 1st ed., Elsevier, 2007.
[31] R. Piro, Model Theoretic Characterisations of Description Logics, Ph.D. thesis, Department of Computer Science, 2013.

A. Proofs of Section 4

Lemma 1. *Let $\mathcal{O}$ be an $\mathcal{ALC}$ ontology, then $\mathcal{O} \equiv^{\mathcal{M}}_{sig(\mathcal{O})} \mathcal{O}^{clausal}$.*

Proof. Standard NNF and CNF transformations preserve logical equivalence. It remains to show $\equiv^{\mathcal{M}}_{sig(\mathcal{O})}$ equivalence is preserved by structural transformation. Let $\mathcal{O}_1$ be an ontology and $\mathcal{C} = \mathcal{Q}r.E$ be a concept in $\mathcal{O}_1$ where $\mathcal{Q} \in \{\exists, \forall\}$. After structural transformation we move to a new ontology $\mathcal{O}_2 = \mathcal{O}'_1 \cup \{\neg D \sqcup E\}$ where $\mathcal{O}'_1$ is equal to $\mathcal{O}_1$ but replaces $\mathcal{C}$ with $\mathcal{Q}r.D$ and $D \notin sig(\mathcal{O}_1)$. Assume that $\mathcal{I}$ is a model of $\mathcal{O}_2$. It is straight forward to see that $\mathcal{I}$ is also a model of $\mathcal{O}_1$. For the reverse direction, assume that $\mathcal{I}$ is a model of $\mathcal{O}_1$. We move to a new model $\mathcal{J}$ which extends $\mathcal{I}$ with D and interprets it as $D^{\mathcal{J}} = \{y \in E^{\mathcal{I}} | (x, y) \in r^{\mathcal{I}} \wedge x \in \mathcal{C}^{\mathcal{I}}\}$. Then, we have that $\mathcal{J} \models \mathcal{O}_2$, and $\mathcal{I}$ and $\mathcal{J}$ $sig(\mathcal{O}_0)$-coincide. Altogether, we get that $\mathcal{O}_1 \equiv^{\mathcal{M}}_{sig(\mathcal{O}_1)} \mathcal{O}_2$. Given an ontology $\mathcal{O}$ in NNF and forgetting signature $\mathcal{F}$, we apply the above transformation exhaustively until no forgetting symbol is present under role restriction in $\mathcal{O}$. This gives a finite series of ontologies $\mathcal{O}_i$ where $0 \leq i \leq n$, $\mathcal{O}_0 = \mathcal{O}$, $\mathcal{O}_n = \mathcal{O}^{clausal}$ and $\mathcal{O}_i$ is generated by applying a single structural transformation step on $\mathcal{O}_{i-1}$. This series is finite and is bounded by the number of role restrictions in $\mathcal{O}$. By the above argument we have $\mathcal{O}_0 \equiv^{\mathcal{M}}_{sig(\mathcal{O}_0)} \ldots \equiv^{\mathcal{M}}_{sig(\mathcal{O}_{n-1})} \mathcal{O}_n$. But $sig(\mathcal{O}_i) \subseteq sig(\mathcal{O}_{i+1})$ because the transformation only introduces new symbols. So $\mathcal{O}_0 \equiv^{\mathcal{M}}_{sig(\mathcal{O}_0)} \dots \equiv^{\mathcal{M}}_{sig(\mathcal{O}_0)} \mathcal{O}_n$. By transitivity it follows that $\mathcal{O}_0 \equiv^{\mathcal{M}}_{sig(\mathcal{O}_0)} \mathcal{O}_n$. □

Theorem 1. $\mathcal{O} \equiv^{\mathcal{M}}_{sig(\mathcal{O})\setminus\mathcal{F}} \mathcal{O}^{int}$.

Proof. Since by Lemma 1 $\mathcal{O} \equiv^{\mathcal{M}}_{sig(\mathcal{O})} \mathcal{O}^{clausal}$, it suffices to prove that $\mathcal{O}^{clausal} \equiv^{\mathcal{M}}_{sig(\mathcal{O})\setminus\mathcal{F}} \mathcal{O}^{int}$. Recall that $\mathcal{O}^{int}$ is generated from $\mathcal{O}^{clausal}$ by resolving on the forgetting symbols exhaustively followed by performing *purity deletion* with *purification* and *tautology deletion* operations applied eagerly throughout the process. Tautology deletion is a standard equivalence preserving method. Purification preserves the interpretation of the non-purified symbols. Therefore, it suffices to prove the correctness of the purity deletion step since resolution only introduces consequences of $\mathcal{O}^{clausal}$. The proof is adapted from [17]. Let A be a forgetting symbol, then in the normal form of $\mathcal{O}^{clausal}$ A does not occur under role restriction. Let $\mathcal{S}_1 = \{E, C \sqcup A, D \sqcup \neg A, C \sqcup D\}$ be the state before purity deletion where $C \sqcup A$ is a representative of all clauses that contain A positively. This is viable since A appears positively in $\mathcal{O}^{clausal}$ in the clauses $C_i \sqcup A$ for $1 \leq i \leq n$, and these clauses can be rewritten equivalently using the single formula $C \sqcup A$ where $C = \bigsqcap C_i$. In the same way, assume E is a representative of all clauses that do not contain A, and $D \sqcup \neg A$ is a representative of all clauses that contain A negatively. $C \sqcup D$ is the resolvent of $C \sqcup A$ and $D \sqcup \neg A$. Let $\mathcal{S}_2 = \{E, C \sqcup D\}$ be the state after purity deletion, omitting $\{C \sqcup A, D \sqcup \neg A\}$ from $\mathcal{S}_1$. Let $\mathcal{I}$ be a model of $\mathcal{S}_1$, then $\mathcal{I}$ is a model of $\mathcal{S}_2$ because $\mathcal{S}_2$ is a subset of $\mathcal{S}_1$. For the reverse direction, suppose that $\mathcal{I}$ is a model of $\mathcal{S}_2$. First, observe that for every $x \in \Delta^{\mathcal{I}}$ we have that $x \in (C \sqcup D)^{\mathcal{I}}$. We extend $\mathcal{I}$ with new concept symbol A interpreted as follows. For every domain element $x \in \Delta^{\mathcal{I}}$:

1. if $x \in C^{\mathcal{I}}$ then $x \notin A^{\mathcal{I}}$, else
2. if $x \in D^{\mathcal{I}}$ then $x \in A^{\mathcal{I}}$

Call the extended model, $\mathcal{J}$. Evidently, $\mathcal{J} \models \mathcal{S}_1$, at the same time, $\mathcal{J}$ interprets every other symbol exactly the same as $\mathcal{I}$. Therefore, $\mathcal{O} \equiv^{\mathcal{M}}_{sig(\mathcal{O})\setminus\mathcal{F}} \mathcal{O}^{int}$. □

Theorem 2. *The size of $\mathcal{O}^{int}$ is in the worst case exponential in the size of the given ontology $\mathcal{O}$ and double exponential in the number of forgetting symbols.*

Proof. Suppose $|\mathcal{F}| = k$, and $|\mathcal{O}^{clausal}| = m$ where the size of $\mathcal{O}^{clausal}$ is taken to be the number of its clauses. Consider a single iteration of resolution, and let $A \in \mathcal{F}$ be the forgetting symbol. Since we use binary resolution, we get $|$Forget($\mathcal{O}^{clausal}$, A)$| = \mathcal{O}(m^2)$. Repeating for k symbols we get $|$Forget($\mathcal{O}^{clausal}, \mathcal{F}$)$| = \mathcal{O}(m^{2^k})$. We now calculate m. Suppose $|\mathcal{O}| = n$ where the size of $\mathcal{O}$ is taken to be the number of axioms in $\mathcal{O}$. Observe that the size of $\mathcal{O}^{clausal}$ is dominated by conversion to CNF because structural transformation is linearly bounded by the number of role restrictions and conversion to NNF does not add new axioms. The size of a CNF formula is, in the worst case, exponential in the size of original formula[28]. Therefore $m = \mathcal{O}(2^n)$. Altogether we get that $|$Forget($\mathcal{O}, \mathcal{F}$)$| = \mathcal{O}(2^{n.2^k})$. □

B. Proofs of Section 5

Lemma 2. *The conclusion of the* Role Propagation *rule in Figure 2 is entailed by the premises.*

Proof. Let $\mathcal{I}^{int}$ be an arbitrary model of $\mathcal{O}^{int}$ and d be a domain element in $\Delta^{\mathcal{I}^{int}}$. If $d \notin (E_0 \sqcup ... \sqcup E_n)^{\mathcal{I}^{int}}$, then it must be the case that $d \in (\mathcal{Q}r.D_0 \sqcap \forall r.D_1 \sqcap ... \sqcap \forall r.D_n)^{\mathcal{I}^{int}}$. This is equivalent to saying $d \in (\mathcal{Q}r.D_0 \sqcap \forall r.(D_1 \sqcap ... \sqcap D_n))^{\mathcal{I}^{int}}$. Let $\mathcal{Q} = \exists$, then there is $e \in D_0^{\mathcal{I}^{int}}$ such that $(d, e) \in r^{\mathcal{I}^{int}}$. It must also be that $e \in (D_0 \sqcap ... \sqcap D_n)^{\mathcal{I}^{int}}$. Observe that $P_0 \equiv \neg(D_0 \sqcap ... \sqcap D_n)$, so $e \notin P_0^{\mathcal{I}^{int}}$. But since $\mathcal{I}^{int} \models P_0 \sqcup C_0$ we get that $e \in C_0^{\mathcal{I}^{int}}$. Similarly, since $P_j \sqsubseteq P_0$, we have $e \in C_j^{\mathcal{I}^{int}}$. Altogether, $d \in (E_0 \sqcup ... \sqcup E_n \sqcup \exists r.(C_0 \sqcap ... \sqcap C_m))^{\mathcal{I}^{int}}$ for any domain element d. If $\mathcal{Q} = \forall$, then $d \in \forall r.(D_0 \sqcap ... \sqcap D_n))^{\mathcal{I}^{int}}$ which is equivalent to saying that $d \in (\forall r.\neg P_0)^{\mathcal{I}^{int}}$. It follows that $d \in (\forall r.C_0)^{\mathcal{I}^{int}}$. Additionally, it must be the case that $d \in (\forall r.\neg P_j)^{\mathcal{I}^{int}}$ because $\forall r.\neg P_j$ subsumes $\forall r.\neg P_0$. So $d \in (\forall r.C_j)^{\mathcal{I}^{int}}$ Altogether, $d \in (E_0 \sqcup ... \sqcup E_n \sqcup \forall r.(C_0 \sqcap ... \sqcap C_m))^{\mathcal{I}^{int}}$ for any domain element d. □

Theorem 3. $\mathcal{O}^{rp} \equiv^{C}_{sig(\mathcal{O})\setminus\mathcal{F}} \mathcal{O}^{red}$.

Proof. We incrementally build the proof through the remaining definitions and lemmas in this section.

Definition 5. *Let $\mathcal{O}_1$ and $\mathcal{O}_2$ be any two ontologies. By* mDiff$(\mathcal{O}_1, \mathcal{O}_2)$ *we mean the set of models of $\mathcal{O}_1$ that are not models of $\mathcal{O}_2$.*

Lemma 3. *mDiff($\mathcal{O}^{rp}, \mathcal{O}^{red}$) = $\emptyset$, but in general mDiff($\mathcal{O}^{red}, \mathcal{O}^{rp}$) $\neq \emptyset$.*

Proof. (1) mDiff($\mathcal{O}^{rp}$, $\mathcal{O}^{red}$) = $\emptyset$: Let $\mathcal{I}$ be any model of $\mathcal{O}^{rp}$. Since $\mathcal{O}^{red} \subseteq \mathcal{O}^{rp}$, it must be that $\mathcal{I} \models \mathcal{O}^{red}$.

(2) mDiff($\mathcal{O}^{red}$, $\mathcal{O}^{rp}$) $\neq$ $\emptyset$: We prove this by giving an example. Let $\mathcal{O}^{rp} = \{\exists r.D_1, \exists r.D_2, \neg D_1 \sqcup \neg D_2\}$ where D_1 and D_2 are definer symbols. Then, $\mathcal{O}^{red} = \{\exists r.D_1, \exists r.D_2\}$. Let $\mathcal{I}$ be the interpretation with $\Delta^{\mathcal{I}} = \{a, b\}$, and $D_1^{\mathcal{I}} = D_2^{\mathcal{I}} = \{b\}, r^{\mathcal{I}} = \{(a, b)\}$. Clearly $\mathcal{I}$ is a model of $\mathcal{O}^{red}$ but is not a model of $\mathcal{O}^{rp}$. □

Lemma 4. $\mathcal{O}^{rp} \not\equiv^{C}_{sig(\mathcal{O})\setminus\mathcal{F}} \mathcal{O}^{red}$ *iff there exists a concept inclusion α over $sig(\mathcal{O})\setminus\mathcal{F}$ such that $\mathcal{O}^{rp} \models \alpha$ and $\mathcal{O}^{red} \not\models \alpha$.*

Proof. The right to left direction is obvious. Left to right: Suppose $\mathcal{O}^{rp} \not\equiv^{C}_{sig(\mathcal{O})\setminus\mathcal{F}} \mathcal{O}^{red}$. By definition there must be a concept inclusion $\alpha = C \sqsubseteq D$ over $sig(\mathcal{O})\setminus\mathcal{F}$ such that:

1. $\mathcal{O}^{rp} \models \alpha$ and $\mathcal{O}^{red} \not\models \alpha$, or
2. $\mathcal{O}^{rp} \not\models \alpha$ and $\mathcal{O}^{red} \models \alpha$.

Consider the second case, there must be a model $\mathcal{I}$ of $\mathcal{O}^{rp}$ that satisfies $C \sqcap \neg D$. By Lemma 3 $\mathcal{I}$ is a model of $\mathcal{O}^{red}$ which contradicts the assumption that $\mathcal{O}^{red} \models \alpha$. □

To proceed, we need to define the notions of *bisimulation* and *tree unravelling*.

Definition 6. *A* pointed interpretation $(\mathcal{I}, d)$ *is an interpretation $\mathcal{I}$ generated by $d \in \Delta^{\mathcal{I}}$. $(\mathcal{I}, d)$ is a directed graph with the root d, and for any $e_1, e_2 \in \Delta^{\mathcal{I}}$, there is a transition, or an edge, from e_1 to e_2 iff $(e_1, e_2) \in r^{\mathcal{I}}$ where $r \in N_r$.*

Definition 7. *Let $(\mathcal{I}, d_1)$ and $(\mathcal{J}, d_2)$ be two pointed interpretations, and Σ a signature. $(\mathcal{I}, d_1), (\mathcal{J}, d_2)$ are* Σ-bisimilar, *in symbols $(\mathcal{I}, d_1) \sim_\Sigma (\mathcal{J}, d_2)$, iff there is a relation $R \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{J}$ where $(d_1, d_2) \in R$ and for every $(d, d') \in R$ the following hold:*

1. *$d \in A^\mathcal{I}$ iff $d' \in A^\mathcal{J}$ for all concept names $A \in \Sigma$.*
2. *if $(d, e) \in r^\mathcal{I}$ then there is $e' \in \Delta^\mathcal{J}$ such that $(d', e') \in r^\mathcal{J}$ for every role name $r \in \Sigma$ and $(e, e') \in R$.*
3. *if $(d', e') \in r^\mathcal{J}$ then there is $e \in \Delta^\mathcal{I}$ such that $(d, e) \in r^\mathcal{I}$ for every role name $r \in \Sigma$ and $(e, e') \in R$.*

It is well known that bisimilar interpretations coincide on all $\mathcal{ALC}$-consequences. This is stated in the following lemma borrowed from [10].

Lemma 5. *[10] Let $(\mathcal{I}, d_1), (\mathcal{J}, d_2)$ be two pointed interpretations, and let Σ be some signature. If $(\mathcal{I}, d_1), (\mathcal{J}, d_2)$ are Σ-bisimilar then for every $\mathcal{ALC}$ concept C where $sig(C) \subseteq \Sigma$ we have that $d_1 \in C^\mathcal{I}$ iff $d_2 \in C^\mathcal{J}$, in symbols $(\mathcal{I}, d_1) \equiv_\Sigma (\mathcal{J}, d_2)$.*

Every $\mathcal{ALC}$ pointed interpretation $(\mathcal{I}, d)$ can be *unravelled* into a *tree interpretation* $(\mathcal{I}', d)$ [29, 30, 31] with $\Delta^{\mathcal{I}'}$ defined as follows:

1. $d \in \Delta^{\mathcal{I}'}$; and
2. The word $w = d.r_1.d_1.r_2.d_2. \dots .r_n.d_n$ is in $\Delta^{\mathcal{I}'}$ if and only if there is a path in $(\mathcal{I}, d)$ from d to d_n along the edges $r_i \in N_r$ and the nodes $d_i \in \Delta^\mathcal{I}$ where $1 \leq i \leq n$.

For every concept name $A \in N_c \cup N_d$ and role name $r \in N_r$:

1. $A^{\mathcal{I}'} = \{d.r_1 \dots r_n.d_n \in \Delta^{\mathcal{I}'} \mid d_n \in A^\mathcal{I}\}$, and
2. $r^{\mathcal{I}'} = \{(w_1, w_1.r.d') \mid w_1, w_1.r.d' \in \Delta^{\mathcal{I}'} \wedge d' \in \Delta^\mathcal{I}\}$.

$(\mathcal{I}', d)$ can be seen as a tree whose nodes are the elements of $\Delta^{\mathcal{I}'}$ and edges are the role names in N_r. By construction, $(\mathcal{I}', d)$ has the following properties:

1. Every node in $(\mathcal{I}', d)$ has exactly one predecessor, except the root node d which does not have a predecessor.
2. $r^{\mathcal{I}'} \cap s^{\mathcal{I}'} \neq \emptyset$ if and only if $r = s$ for all $r, s \in N_r$.
3. For every $e \in \Delta^\mathcal{I}$ there exists $e' \in \Delta^{\mathcal{I}'}$ and vice versa such that $e \in C^\mathcal{I}$ if and only if $e' \in C^{\mathcal{I}'}$ where C is an $\mathcal{ALC}$ concept.
4. If (I, d) is a cyclic graph, then $(\mathcal{I}', d)$ is an acyclic tree with infinite depth.

In proving Theorem 3, suppose for the sake of contradiction that $\mathcal{O}^{rp} \not\equiv^C_{sig(\mathcal{O})\setminus\mathcal{F}} \mathcal{O}^{red}$, or by Lemma 4 there is a concept inclusion $\alpha = C \sqsubseteq E$ with $\mathcal{O}^{rp} \models \alpha$ and $\mathcal{O}^{red} \not\models \alpha$. Let $\mathcal{I}$ be a model of $\mathcal{O}^{red}$ generated by an arbitrary $d \in C^\mathcal{I} \cap \neg E^\mathcal{I}$, then $\mathcal{I} \in$ mDiff($\mathcal{O}^{red}, \mathcal{O}^{rp}$). The following Lemma sets a condition on $\mathcal{I}$.

Lemma 6. *Suppose $\mathcal{O}^{rp} = \mathcal{O}^{red} \cup \{\neg D_1 \sqcup \dots \sqcup \neg D_n \sqcup F\}$ where $n > 1$, and let $(\mathcal{I}, d)$ be as above. Then, $\mathcal{I} \in$ mDiff($\mathcal{O}^{red}, \mathcal{O}^{rp}$) iff there is $e \in \Delta^\mathcal{I}$ that is reachable from d where $e \in D_1^\mathcal{I} \cap \dots \cap D_n^\mathcal{I}$ and $e \notin F^\mathcal{I}$.*

Proof. Right to left is obvious. Left to right: Suppose there is no such e, then $\mathcal{I} \models \neg D_1 \sqcup \ldots \sqcup \neg D_n \sqcup F$. But also $\mathcal{I} \models \mathcal{O}^{red}$, so we get that $\mathcal{I} \models \mathcal{O}^{rp}$ which contradicts that $\mathcal{I} \in$ mDiff$(\mathcal{O}^{red}, \mathcal{O}^{rp})$. □

Definition 8. *Let $\mathcal{I}$ be a model and $e \in \Delta^{\mathcal{I}}$. Recall that N_d is the set of definer symbols, and define $\mathcal{C}_{\mathcal{I}}(e)$ to be the closure under single negation of the symbols in $N_c \cup N_d$ that contain e in their interpretation.*

Lemma 7. *Let $\mathcal{O}^{rp}$, $\mathcal{O}^{red}$, and $\mathcal{I}$ be defined as in Lemma 6. There is a model $\mathcal{J}$ generated by d such that:*

1. $\mathcal{J} \models \mathcal{O}^{red}$;
2. $(\mathcal{I}, d) \sim_{sig(\mathcal{O}^{red})\setminus N_d} (\mathcal{J}, d)$; *and*
3. *There is no $e \in \Delta^{\mathcal{J}}$ such that $e \in D_1^{\mathcal{J}} \cap \ldots \cap D_n^{\mathcal{J}} \cap (\neg F)^{\mathcal{J}}$.*

Proof. Let $\mathcal{I}_0$ be a model that coincides with $\mathcal{I}$ on everything but reinterprets the definer symbols as follows:

$$D^{\mathcal{I}_0} := D^{\mathcal{I}} \cap \{y \in \Delta^{\mathcal{I}_0} \mid (x,y) \in r^{\mathcal{I}_0} \text{ and } x \notin G^{\mathcal{I}_0} \text{ and } (\mathcal{O}^{red} \models G \sqcup \exists r.D, \text{ or } \mathcal{O}^{red} \models G \sqcup \forall r.D)\} \quad (1)$$

where $D \in N_d, r \in N_r$, and G is an $\mathcal{ALC}$ concept.

The idea of (1) is to restrict the elements in the extension of D to the minimum set that is required for $\mathcal{O}^{red}$. For example, suppose $\mathcal{O}^{red} = \{\neg A \sqcup \forall r.D_1, \neg B \sqcup \forall r.D_2, \neg D_1 \sqcup \neg D_2, \neg A \sqcup \neg B \sqcup \forall r.\bot\}$. The model $\mathcal{I}$ with $\Delta^{\mathcal{I}} = \{a,b\}, A^{\mathcal{I}} = \{a\}, B^{\mathcal{I}} = \{\}, D_1^{\mathcal{I}} = D_2^{\mathcal{I}} = \{b\}, r^{\mathcal{I}} = \{(a,b)\}$ is a tree model of $\mathcal{O}^{red}$. But $D_2^{\mathcal{I}} = \{b\}$, which violates the intuition that D_2 represents the r-successors of elements in $B^{\mathcal{I}}$. Re-interpreting D_2 using (1) removes b from $D_2^{\mathcal{I}}$. We shall find this necessary when constructing $\mathcal{J}$.

Recall that in $\mathcal{O}^{rp}$ a definer symbol D exists positively only in clauses of the form $C \sqcup \exists r.D$ or $C \sqcup \forall r.D$, and negatively only in clauses of the form $\neg D \sqcup C$. Then, it can be seen that $\mathcal{I}_0$ is a model of $\mathcal{O}^{red}$ since all clauses of $\mathcal{O}^{red}$ on the forms of $G \sqcup \exists r.D$ and $G \sqcup \forall r.D$ are true in $\mathcal{I}_0$ by the side conditions of (1). Also, since (1) only removes elements from the extension of D, clauses on the form of $\neg D \sqcup H$ remain satisfied in $\mathcal{I}_0$. As (1) only modifies the interpretations of definer symbols, it also follows that $(\mathcal{I}, d) \sim_{sig(\mathcal{O}^{red})\setminus N_d} (\mathcal{I}_0, d)$.

We shall now construct a sequence of interpretations $\mathcal{I}_1, \mathcal{I}_2, \ldots$. The interpretations are constructed such that $\mathcal{I}_{k+1}$ is a transformation of $\mathcal{I}_k$ that eliminates one domain element $e \in D_1^{\mathcal{I}_k} \cap \ldots \cap D_n^{\mathcal{I}_k} \cap (\neg F)^{\mathcal{I}_k}$ where $k \geq 0$, and $(\mathcal{I}_k, d) \sim_{sig(\mathcal{O}^{red})\setminus N_d} (\mathcal{I}_{k+1}, d)$. The limit of this sequence is $\mathcal{J}$. It then follows by transitivity of $\sim$ that $(\mathcal{I}, d) \sim_{sig(\mathcal{O}^{red})\setminus N_d} (\mathcal{J}, d)$, and there is no $e \in \Delta^{\mathcal{J}}$ such that $e \in D_1^{\mathcal{J}} \cap \ldots \cap D_n^{\mathcal{J}} \cap (\neg F)^{\mathcal{J}}$.

Suppose there is a domain element $e \in \Delta^{I_k}$ such that $e \in D_1^{\mathcal{I}_k} \cap \ldots \cap D_n^{\mathcal{I}_k} \cap (\neg F)^{\mathcal{I}_k}$. It is guaranteed by (1) that e has a predecessor, call it e_{pre}, such that for every i with $1 \leq i \leq n$ we have $\mathcal{O}^{red} \models G_i \sqcup \exists r.D_i$ or $\mathcal{O}^{red} \models G_i \sqcup \forall r.D_i$, and $e_{pre} \notin G_i^{\mathcal{I}_k}$. Note that the case when $\mathcal{O}^{red} \models G_i \sqcup \forall r.D_i$ for all i with $1 \leq i \leq n$ is not possible since by the construction of $\mathcal{O}^{red}$ the clause $G_1 \sqcup \cdots \sqcup G_n \sqcup \forall r.F$ must have been introduced in $\mathcal{O}^{red}$ by the *Role Propagation* rule in Figure 2. Since $e_{pre} \notin G_i^{\mathcal{I}_k}$ with $1 \leq i \leq n$, it must be that $e_{pre} \in (\forall r.F)^{\mathcal{I}_k}$, hence $e \in F^{\mathcal{I}_k}$

which contradicts the hypothesis that $e \notin F^{\mathcal{I}_k}$. For the other cases we transform $\mathcal{I}_k$ to $\mathcal{I}_{k+1}$ as follows:

(I) Suppose that there is an l such that $1 \leq l \leq n$ with $\mathcal{O}^{red} \models G_l \sqcup \exists r.D_l$ and $\mathcal{O}^{red} \models G_i \sqcup \forall r.D_i$ where $1 \leq i \leq n$ and $i \neq l$. Again from the *Role Propagation* rule we must have $\mathcal{O}^{red} \models G_1 \sqcup \dots G_n \sqcup \exists r.(F \sqcap \bigsqcap_{j=1}^{m} F_j)$, where $\mathcal{O}^{rp} \models P_j \sqcup F_j$ and P_j is any sub-concept of $\neg D_1 \sqcup \dots \sqcup \neg D_n$ (these are the second premise of the *Role Propagation* rule). Since $e_{pre} \notin G_i^{\mathcal{I}_k}$ with $1 \leq i \leq n$, it must be that $e_{pre} \in (\exists r.(F \sqcap \bigsqcap_{j=1}^{m} F_j))^{\mathcal{I}_k}$. That is, there is a child e' of e_{pre} via r, such that $e' \in D_i^{\mathcal{I}_k}$ where $1 \leq i \leq n$ and $i \neq l$, and $e' \in F^{\mathcal{I}_k}$. We construct a model $\mathcal{I}_{k+1}$ which is equivalent to $\mathcal{I}_k$ in everything except D_l which it interprets differently: If $e' \in D_l^{\mathcal{I}_k}$, then $D_l^{\mathcal{I}_{k+1}} := D_l^{\mathcal{I}_k} \backslash \{e\}$. If $e' \notin D_l^{\mathcal{I}_k}$, then $D_l^{\mathcal{I}_{k+1}} := (D_l^{\mathcal{I}_k} \cup \{e'\}) \backslash \{e\}$. We note that $\mathcal{I}_{k+1}$ is a model of $\mathcal{O}^{red}$: (1) By removing e from the interpretation of D_l, we need only to worry about clauses of $\mathcal{O}^{red}$ where D_l appears positively, that is, we show that clauses of the form $G \sqcup \exists r.D_l$ are satisfied at e_{pre}. Since by the transformation above $e' \in D_l^{\mathcal{I}_{k+1}}$ and $(e_{pre}, e') \in r^{\mathcal{I}_{k+1}}$, we have $e_{pre} \in (\exists r.D_l)^{\mathcal{I}_{k+1}}$. Therefore $e_{pre} \in (G \sqcup \exists r.D_l)^{\mathcal{J}_d}$ for any $\mathcal{ALC}$ concept G. (2) By adding e' to the interpretation of D_l, we need only to worry about clauses of $\mathcal{O}^{red}$ where D_l appears negatively, that is, we show that $e' \in (\neg D_l \sqcup G)^{\mathcal{I}_{k+1}}$ for any $\mathcal{ALC}$ concept G. Since $\neg D_l \sqcup G \in \mathcal{O}^{red}$, it must have been a premise for the *Role Propagation* rule, i.e., there must be a j such that $1 \leq j \leq m$ such that $G = F_j$. Since $e' \in (F \sqcap \bigsqcap_{j=1}^{m} F_j)^{\mathcal{I}_{k+1}}$, then $e' \in F_j$, consequently $e' \in (\neg D \sqcup G)^{\mathcal{I}_{k+1}}$.

As the transformation only changes the interpretation of D_l, it can be seen that $\mathcal{I}_k \sim_{sig(\mathcal{O}^{red})\backslash\{D_l\}} \mathcal{I}_{k+1}$.

(II) Suppose that $\mathcal{O}^{red} \models G_i \sqcup \exists r.D_i$ with $1 \leq i \leq p$ and $2 \leq p \leq n$, and $\mathcal{O}^{red} \models G_j \sqcup \forall r.D_j$ with $p < j \leq n$. That is, two or more definers in $D_1, \dots, D_n$ occur under existential role restriction. In this case the *Role Propagation* rule in Figure 2 does not apply. We construct a model $\mathcal{I}_{k+1}$ which is equivalent to $\mathcal{I}_k$ in everything except that it replaces e with fresh domain elements e_i where $1 \leq i \leq p$ such that $e_i \notin D_i^{\mathcal{I}_{k+1}}$, $e_i \in D_j^{\mathcal{I}_{k+1}}$ with $1 \leq j \leq p$ and $j \neq i$, and:

1. $e_i \in A^{\mathcal{I}_{k+1}}$ iff $A \in \mathcal{C}_{\mathcal{I}_k}(e) \backslash \{D_i\}$ for every $A \in N_c \cup N_d$;
2. $(e_{pre}, e_i) \in r^{\mathcal{I}_{k+1}}$;
3. $(e_i, e') \in r^{\mathcal{I}_{k+1}}$ iff $(e, e') \in r^{\mathcal{I}_k}$.

where $r \in N_r$.

First we show that $\mathcal{I}_{k+1} \models \mathcal{O}^{red}$: (1) As before, by removing e from $\Delta^{\mathcal{I}_{k+1}}$ we need only to show that clauses of $\mathcal{O}^{red}$ of the form $G_i \sqcup \exists r.D_i$ where $1 \leq i \leq p$ are satisfied at e_{pre}. By construction, e is replaced with e_i where $1 \leq i \leq p$ such that $e_i \in D_j^{\mathcal{I}_{k+1}}$ with $1 \leq j \leq p$ and $j \neq i$. So for every D_i with $1 \leq i \leq p$ there are $p - 1$ new domain elements e_j such that $e_j \in D_i^{\mathcal{I}_{k+1}}$ and $(e_{pre}, e_j) \in r^{\mathcal{I}_{k+1}}$. Therefore, $e_{pre} \in (\exists r.D_i)^{\mathcal{I}_{k+1}}$ consequently $(G_i \sqcup \exists r.D_i)^{\mathcal{I}_{k+1}}$ for $1 \leq i \leq p$. (2) By introducing new elements e_i such that $e_i \notin D_i^{\mathcal{I}_{k+1}}$ with $1 \leq i \leq p$, we need to show that if $\neg D_i \sqcup G_i \in \mathcal{O}^{red}$ then $e_i \in G_i^{\mathcal{I}_{k+1}}$. Since $e \in G_i^{\mathcal{I}_k}$, it follows from the first condition of the transformation that $e_i \in G_i^{\mathcal{I}_{k+1}}$. Altogether, $\mathcal{I}_{k+1} \models \mathcal{O}^{red}$.

Second we prove that $(\mathcal{I}_k, d) \sim_{sig(\mathcal{O}^{red})\backslash N_d} (\mathcal{I}_{k+1}, d)$ by induction from bottom to top. Assume that e, (hence e_i) is reachable from d in m transitions. Let $\Sigma = sig(\mathcal{O}^{red}) \backslash \{D_1, \dots, D_n\}$.

By construction, for any node $u \in (\mathcal{I}_k, d)$ or $(\mathcal{I}_{k+1}, d)$ such that u is reachable from d in at least $m+1$ transitions, we have $(\mathcal{I}_k, u) \sim_\Sigma (\mathcal{I}_{k+1}, u)$. That is, nothing below e and e_i has changed. Second, for every $u \in \mathcal{I}_k$ at depth m we have: (1) If $u = e$, then by construction $\mathcal{C}_{\mathcal{I}_k}(u) \backslash \{D_i\} = \mathcal{C}_{\mathcal{I}_{k+1}}(e_i)$. Also, since the third condition of the transformation guarantees that everything below u and e_i is the same, it follows that $(\mathcal{I}_k, u) \sim_\Sigma (\mathcal{I}_{k+1}, e_i)$. (2) If $u \neq e$, then u is also a node in $(\mathcal{I}_{k+1}, d)$, hence $(\mathcal{I}_k, u) \sim_\Sigma (\mathcal{I}_{k+1}, u)$. Third, for every $u \in (\mathcal{I}_k, d)$ such that u is reachable in at most $m-1$ transitions from d we have that $u \in (\mathcal{I}_{k+1}, d)$, and we have the following cases: (1) $e \notin (\mathcal{I}_k, u)$. Then, as before $(\mathcal{I}_k, u) \sim_\Sigma (\mathcal{I}_{k+1}, u)$. (2) $e \in (\mathcal{I}_k, u)$. Condition 1 in the transformation guarantees that $\mathcal{C}_{\mathcal{I}_k}(u) = \mathcal{C}_{\mathcal{I}_{k+1}}(u)$ on all $C \in (N_c \cup N_d)$. Suppose that $u \in (\exists \vec{r}.C)^{\mathcal{I}_k}$ where $\exists \vec{r} = \exists r_1.\exists.r_2....\exists.r_l$, $r_i \in N_r$, and C is any $\mathcal{ALC}$ concept over Σ. Then there must be a path $ur_1u_1r_2u_2r_3...r_lu_l$ in $\mathcal{I}_k$ such that $u_l \in C^{\mathcal{I}_k}$. If $u_i \neq e$ for every $i \in [1..l]$ then by construction this path also exists in $(\mathcal{I}_{k+1}, d)$ and $u \in (\exists \vec{r}.C)^{\mathcal{I}_{k+1}}$. If $u_i = e$ for any $i \in [1..l]$, then again by construction there is a path $ur_1u_1r_2...r_ivr_{i+1}...r_lu_l$ and we have the choice to set v to any of $\{e_1, ..., e_n\}$. In particular, we have $u_l \in C^{\mathcal{I}_{k+1}}$ and $u \in (\exists \vec{r}.C)^{\mathcal{I}_{k+1}}$. We get that $(\mathcal{I}_k, u) \sim_\Sigma (\mathcal{I}_{k+1}, u)$ for every u reachable in at most $m-1$ transitions from d. The same argument can be used to show that every node in $(\mathcal{I}_{k+1}, d)$ has a bisimilar node in $(\mathcal{I}_k, d)$. Altogether, we get that $\mathcal{I}_k \sim_\Sigma (\mathcal{I}_{k+1}, d)$.

In the sequence of constructed models, it follows that:

1. $\mathcal{J} \models \mathcal{O}^{red}$;
2. $\mathcal{I} \sim_{sig(\mathcal{O}^{red})\backslash N_d} \mathcal{J}$; and
3. $\mathcal{J} \not\models D_1 \sqcap D_2 \sqcap ... \sqcap D_n \sqcap \neg F$.

□

Since $\mathcal{J} \models \mathcal{O}^{red}$ and there is no $e \in \Delta^{\mathcal{J}}$ such that $e \in D_1^{\mathcal{J}} \cap ... \cap D_n^{\mathcal{J}} \cap (\neg F)^{\mathcal{J}}$, it follows by Lemma 6 that $\mathcal{J} \models \mathcal{O}^{rp}$. Also, since $\mathcal{I} \sim_{sig(\mathcal{O}^{red})\backslash N_d} \mathcal{J}$, and $sig(\mathcal{O}^{red})\backslash N_d = sig(\mathcal{O})\backslash \mathcal{F}$, it follows by Lemma 5 that $\mathcal{J} \models C \sqcap \neg E$. But this contradicts the assumption that $\mathcal{O}^{rp} \models C \sqsubseteq E$ implying that the assumption that there is $\alpha = C \sqsubseteq E$ with $\mathcal{O}^{rp} \models \alpha$ and $\mathcal{O}^{red} \not\models \alpha$ is incorrect, and $\mathcal{O}^{rp} \equiv^C_{sig(\mathcal{O}^{red})\backslash N_d} \mathcal{O}^{red}$.

Finally assume $\mathcal{O}^{rp} = \mathcal{O}^{red} \cup \Delta$ with Δ being a set of n clauses of the form $\neg D_1 \sqcup \cdots \sqcup D_n \sqcup F$. We construct n ontologies $\mathcal{O}^{rp}_i$ where $\mathcal{O}^{rp}_n = \mathcal{O}^{rp}$, $\mathcal{O}^{rp}_k = \mathcal{O}^{rp}_{k-1} \cup \{\mathcal{S}\}$ where $\mathcal{S}$ is a clause in Δ and $\mathcal{O}^{rp}_0 = \mathcal{O}^{red}$. By the above proof we have $\mathcal{O}^{rp}_k \equiv^C_{sig(\mathcal{O}^{red})\backslash N_d} \mathcal{O}^{rp}_{k-1}$ for $1 \leq k \leq n$. By transitivity of $\equiv^C$ and because $sig(\mathcal{O}^{red})\backslash N_d = sig(\mathcal{O})\backslash \mathcal{F}$ we conclude $\mathcal{O}^{rp} \equiv^C_{sig(\mathcal{O})\backslash \mathcal{F}} \mathcal{O}^{red}$. □

C. Proofs of Section 6

Theorem 4. *Let $\mathcal{V}$ be generated from $\mathcal{O}^{red}$ by applying the Definer Elimination rule from Figure 3 exhaustively. Then:*

1. $\mathcal{O}^{red} \equiv^{\mathcal{M}}_{sig(\mathcal{O})\backslash \mathcal{F}} \mathcal{V}$;
2. *if $sig(\mathcal{V}) \cap N_d = \emptyset$ then $\mathcal{V}$ is a deductive forgetting view of $\mathcal{O}$ w.r.t. $\mathcal{F}$.*

Size of Ontology			#Concept Names		
Avg.	Median	Max	Avg.	Median	Max
22663	4351	133290	10152	3390	73390

Table 2
Average, median, and maximum number of axioms and concept names in our ontology corpus.

Proof. First we prove 1. The Definer Elimination rule in Figure 3 can be seen as a two step operation. The first replaces all clauses of the form $\neg D \sqcup C_i$ with a single clause $\neg D \sqcup C$ where $C = \sqcap C_i$. This step clearly preserves equivalence. The second replaces every D in $\mathcal{O}$ with the concept C. This step is the inverse of structural transformation. Therefore, by Lemma 1 we get that $\mathcal{O}^{red} \equiv^{\mathcal{M}}_{sig(\mathcal{V})} \mathcal{V}$.

Second we prove 2. By Theorem 1, we get that $\mathcal{O}$ and $\mathcal{O}^{int}$ coincide on all interpretations of the $sig(\mathcal{O})\backslash\mathcal{F}$ symbols, consequently on all $\mathcal{ALC}$ concept inclusions over $sig(\mathcal{O})\backslash\mathcal{F}$. By Theorem 3, both $\mathcal{O}^{red}$ and $\mathcal{O}^{rp}$, consequently also $\mathcal{O}^{int}$ and $\mathcal{O}$, agree on all $\mathcal{ALC}$ concept inclusions over $sig(\mathcal{O})\backslash\mathcal{F}$. Finally, since $\mathcal{O}^{red} \equiv^{\mathcal{M}}_{sig(\mathcal{O})\backslash\mathcal{F}} \mathcal{V}$, we get that $\mathcal{O} \equiv^{C}_{sig(\mathcal{O})\backslash\mathcal{F}} \mathcal{V}$, since $\mathcal{V}$ is generated by eliminating the definer symbols in $\mathcal{O}^{red}$. It follows that $\mathcal{V}$ contains only the definer symbols that are witnesses to cycles in $\mathcal{O}$. Altogether we get that $\mathcal{V}$ is a deductive view of $\mathcal{O}$ w.r.t. $\mathcal{F}$. □

D. Experiments

In the experiments, we used a corpus of 50 random ontologies from the NCBO BioPortal. Ontologies expressed in languages more expressive than $\mathcal{ALC}$ were prepared by excluding the axioms that contained non-$\mathcal{ALC}$ constructs.

Table 2 summarizes key statistics of the corpus. The median size of the ontologies was 4351 axioms. That is, at least 50% of the ontologies in the corpus were medium size. The higher average of 22663 indicates that the corpus also included many large-scale ontologies with a maximum of 133290 axioms.

The table also shows a diversity in the number of concept names that occurred in the ontologies, which can be used as an indicator on the complexity of the modeled information in the ontologies. The median number of concept names that occurred in the ontology was 3390, indicating a medium complexity for most ontologies. The high average of 10152 concept names in the ontologies indicated that many ontologies in the corpus used a large number of concept names up to a maximum of 73390. This means that our corpus had many complex ontologies that modeled large amount of information. As our evaluations ran in three settings that correspond to forgetting 10%, 30%, and 50% of the concept names, the information in Table 2 also meant that the average sizes of the forgetting signature were 1015, 3046, and 5076 concept names in the 10%, 30%, and 50% settings respectively.

Code and data are published at the following URL: github.com/SemanticToDeductiveForgetting/sed

E. Structural Transformation

In this section we define *Structural Transformation* adapted to DLs A *position* is a word over natural numbers. Let α be a clause in negation normal form. A concept that occurs in α can be referenced by its *position*. In symbols we write $\alpha|i$ to denote the concept at position i in α.

Consider a general $\mathcal{ALC}$ concept C. Let $pos(C)$ be the set of positions of the sub-concepts occurring in C. A sub-concept D of C is referred to by $\alpha|i.\pi$ where i is position of C relative to α and $\pi \in pos(C)$ is the position of D relative to C. The definition of $pos(C)$ is given inductively as follows:

1. $i.\pi \in pos(C)$ if $C = D_1 \odot D_2 \odot \cdots \odot D_n$ where $\odot \in \{\sqcup, \sqcap\}$, $\pi \in pos(D_i)$ and $1 \leq i \leq n$.
2. $1.\pi \in pos(C)$ if C takes any of the forms: $\neg D$, $\exists r.D$, or $\forall r.D$ and $\pi \in pos(D)$.

Example 9. *Consider the axiom* $\neg A \sqcup \exists r.B \sqsubseteq D \sqcap (C_1 \sqcup \forall r.\forall r.C_2)$. *In negation normal form it becomes* $\alpha = (A \sqcap \forall r.\neg B) \sqcup (D \sqcap (C_1 \sqcup \forall r.\forall r.C_2))$ *The following table lists the subconcepts of* α *and their positions relative to* α.

Concept	*Position*	*Concept*	*Position*
$A \sqcap \forall r.\neg B$	$\alpha|1$	$D \sqcap (C_1 \sqcup \forall r.\forall r.C_2)$	$\alpha|2$
A	$\alpha|1.1$	$\forall r.\neg B$	$\alpha|1.2$
D	$\alpha|2.1$	$C_1 \sqcup \forall r.\forall r.C_2$	$\alpha|2.2$
$\neg B$	$\alpha|1.2.1$	C_1	$\alpha|2.2.1$
$\forall r.\forall r.C_2$	$\alpha|2.2.2$	B	$\alpha|1.2.1.1$
$\forall r.C_2$	$\alpha|2.2.2.1$	C_2	$\alpha|2.2.2.1.1$

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

Definition 9. *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 of the form of* $B \odot E$ *or* $\neg B \odot E$, *where* E *is an* $\mathcal{ALC}$ *concept and* $\odot \in \{\sqcup, \sqcap\}$.

Definition 10. *Let* $\mathcal{O}$ *be an ontology,* $\mathcal{F}$ *a forgetting signature,* $B \in \mathcal{F}$ *a forgetting symbol,* α *a clause in negation normal form and* $\alpha|i$ *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* $i \in pos(\alpha)$. *We say:*

- α *is* structurally transformed at position i *when replacing it with* $\alpha[i/\mathcal{Q}r.D]$, *and adding the clause* $\neg D \sqcup E$ *to the ontology, where* $D \in N_d$ *is a fresh definer symbol, i.e.,* $D \notin sig(\mathcal{O})$.
- α *is* structurally B-transformed *if it structurally transform at* i *for every* i *where* $\alpha|i$ *is a* B-*concept.*
- α *is* structurally transformed *if it is structurally* B-*transformed for every* $B \in \mathcal{F}$
- $\mathcal{O}$ *is* structurally transformed *if every* $\alpha \in \mathcal{O}$ *is structurally transformed.*