Fine-Grained Forgetting for the Description Logic $\mathcal{ALC}$

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

Abstract

Forgetting is an important ontology extraction technique. Two variants of forgetting that have been studied in literature are semantic and deductive forgetting. While deductive forgetting is attractive since it generates the forgetting view in a language with the same complexity as the original ontology, semantic forgetting preserves more information and has higher precision. In this paper, our aim is to find a compromise between both types of forgetting. We present a system that computes a semantic forgetting view in a novel form, and then reduces it to a deductive forgetting view. The system also produces a set Δ of axioms representing the information difference between both views. This enables a new fine-grained ontology extraction process that gives the user the option to enhance the informativeness of the deductive view by appending to it axioms from Δ as desired. The evaluation results show that despite the two stage process, the method is competitive with a state-of-the-art forgetting method, showing a mean performance gain of 50%, 36%, and 55% in different settings.

1 Introduction

Forgetting is the task by which one can eliminate from a given ontology a set of irrelevant vocabulary, and produce a simpler ontology that preserves the semantic content relative to the remaining vocabulary, see e.g. (Eiter and Kern-Isberner 2019). It offers solutions for a wide range of applications such as: *ontology reuse* (Wang et al. 2010), *computing logical difference* (Ludwig and Konev 2014), *information hiding* (Cuenca Grau 2010; Grau and Motik 2010) *abduction* (Del-Pinto and Schmidt 2019), *resolving conflicts* (Lang and Marquis 2010), *relevance* (Subramanian and Genesereth 1987; Lakemeyer 1997; Lang, Liberatore, and Marquis 2003), and *forgetting actions in planning* (Erdem and Ferraris 2007).

Two variants of forgetting have been studied in the literature, namely *semantic* and *deductive* forgetting. Both variants can be characterized model theoretically as follows: In semantic forgetting, the models of the original ontology and the models of the new ontology agree on all interpretations except, possibly, the interpretations of the forgotten symbols (Lin and Reiter 1994a). Whereas, in deductive forgetting, the models of the original ontology and the models of the new ontology are bisimilar (Stirling 1998; French 2006; Divroodi and Nguyen 2015) except over the forgotten symbols (Zhang and Zhou 2009; Lutz and Wolter 2011). It is not hard to see from this characterization that semantic forgetting is stronger than deductive forgetting in terms of the preserved information. In fact, semantic forgetting amounts to an agreement between both ontologies on all consequences, over the remaining vocabulary, that are expressible in any language up to a second order language (Botoeva et al. 2017). In contrast, deductive forgetting amounts to an agreement on only the consequences over the remaining vocabulary that are expressible in the language of the original ontology (Lutz and Wolter 2011; Delgrande 2014).

In practice the following questions arise concerning deductive forgetting: *Will the new ontology capture all model information over the remaining vocabulary? What is the unpreserved information in the new ontology? Can some of this lost information be retrieved in the new ontology?* The aim of this paper is to consider these questions, gain a better understanding of the information difference between semantic and deductive forgetting, and provide a practical tool to compute this information difference.

One might try to address the above questions by computing both variants of forgetting and then compute the logical difference between the results. This approach is however not viable because: (1) There is no complete semantic forgetting method for the description logic $\mathcal{ALC}$, which is the underlying logic considered in this paper. (2) The suggested flow is unpractical since the complexities of forgetting and computing logical difference are, respectively, 3ExpTime, and 2ExpTime (Lutz and Wolter 2011; Konev, Walther, and Wolter 2008).

The contribution of this paper is an optimized solution which integrates semantic and deductive forgetting in one system. The system (1) performs semantic forgetting on a given ontology and forgetting signature, and (2) extracts from the result two sets of axioms. The first is a set $\mathcal{O}^{red}$ equivalent to the result of deductive forgetting. The second is a set Δ that represents the information difference between the results of the semantic and the deductive forgetting. We use a second order language to represent the result of the semantic forgetting, the set Δ, and $\mathcal{O}^{red}$. The language introduces fresh, implicitly existentially quantified concept symbols, and uses only $\mathcal{ALC}$ constructs. Our method is designed to minimise the number of these symbols in the semantic view, and eliminate them from $\mathcal{O}^{red}$, thereby obtaining the final deductive forgetting result. As the result of semantic forgetting of a first order theory is not in general representable in first order logic, but is always representable in second order logic (Ackermann 1935; Lin and Reiter 1994b; Lin and Reiter 1994a; Zhang and Zhou 2010), using some form of second order language is inevitable.

The benefits of this forgetting system are numerous: (1) By computing Δ, we get a novel way of understanding the difference between semantic and deductive forgetting based on the modelled information. (2) Since the semantic forgetting result is represented using $\mathcal{ALC}$ constructs only, and since the newly added symbols are implicitly existentially quantified, this result can be used directly in $\mathcal{ALC}$ applications. (3) The system allows for a fine-grained ontology extraction framework where applications start from $\mathcal{O}^{red}$ and append to it axioms from Δ as needed. (4) In this forgetting framework, the complexity of the final ontology is proportionate to the information modelled in it. This is because the second order symbols are brought into the final ontology on a by-need basis as we append axioms from Δ. (5) It turns out that in the chosen second order language, the set Δ is a syntactically characterized subset of the semantic forgetting result, and the remaining axioms constitute $\mathcal{O}^{red}$, which simplifies the overall system and improves processing performance.

We complement our method with a novel evaluation for which we build our own test ontologies which allows us to customize the hardness of the forgetting problem. We measure the hardness of the forgetting problem by the probability of two or more forgetting symbols appearing together in the same axiom. Accordingly, we evaluate our forgetting method in three different hardness settings, *Low, Moderate*, and *High*. Our evaluation is performed against the state-of-the-art deductive forgetting method of (Koopmann and Schmidt 2013b). Despite the two stage process, the evaluation shows a significant performance gain of 50%, 36%, and 55% in the *Low, Moderate*, and *High* settings respectively when using our method.

2 History and Related Work

Forgetting can be traced back to (Boole 1854) who referred to it as *elimination of the middle terms*. In the context of the propositional logic, it was later studied in relation to *relevance, independence, variable elimination* (Lakemeyer 1997; Lang, Liberatore, and Marquis 2003).

In the context of first order logic, (Lin and Reiter 1994a) viewed (semantic) forgetting as a *second order quantifier elimination problem (SOQE)* (Gabbay and Ohlbach 1992; Gabbay, Schmidt, and Szałas 2008) concluding that the forgetting result of a first order theory is not in general expressible in first order logic, but is always expressible in second order logic (Ackermann 1935). Later, (Zhang and Zhou 2010) proposed, as an alternative to semantic forgetting, the variant of *weak forgetting* which matches our definition of deductive forgetting. The proposal builds on previous work in modal logics which views forgetting as a dual of *uniform interpolation* (Ghilardi 1995; Visser 1996; Herzig and Mengin 2008). In particular, the relation between the original ontology and the new ontology is defined as bisimulation equivalence on all symbols except the forgotten symbol (Zhang and Zhou 2009; Ditmarsch et al. 2008).

In the context of description logics, deductive and semantic forgetting were defined in terms of *inseparability* (Konev, Walther, and Wolter 2009; Konev et al. 2013; Botoeva et al. 2017; Lutz and Wolter 2010). Theoretical complexity results for deductive forgetting are given in (Lutz and Wolter 2011). There, it was shown that deciding the existence of the deductive forgetting result is 2EXPTIME, and its size is, at most, triple exponential in the size of the original ontology. Complexity results for semantic forgetting depend on the language used to represent the result of the forgetting. In our chosen second order language, we show that the size of the result of semantic forgetting is at most single exponential in the size of the input ontology and is double exponential in the number of forgetting symbols.

For general $\mathcal{ALC}$ ontologies, the deductive forgetting result can be captured, possibly infinitely, in $\mathcal{ALC}$. A finite representation is obtainable using fixpoint operators (D'Agostino and Hollenberg 1996; Nonnengart and Szalas 1998; Koopmann and Schmidt 2013b). Recall that complexity of reasoning with respect to general $\mathcal{ALC}$ and $\mathcal{ALC}\mu$ ontologies is EXPTIME (Calvanese, De Giacomo, and Lenzerini 1999). Several deductive forgetting methods were proposed (Ludwig and Konev 2013; Koopmann and Schmidt 2013b; Koopmann and Schmidt 2014; Koopmann and Schmidt 2015). Early work in (Zhao and Schmidt 2015) attempted to capture the the result of semantic forgetting of $\mathcal{ALC}$ ontologies in the more expressive $\mathcal{ALCOI}$ logic. There is however no guarantee that a solution is found where one exists.

In the first stage of our forgetting method, our approach for computing the semantic forgetting is similar to (Gabbay and Ohlbach 1992), and to the deductive forgetting approaches in (Koopmann and Schmidt 2013b; Delgrande 2017). In particular, we iteratively resolve on the forgetting symbols where the original ontology is in *clausal form* (Nonnengart and Weidenbach 2001). Differences appear at this stage to compute the semantic forgetting and capture the result in the special second order language described earlier.

3 Basic Definitions and Ideas

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 general concept expressions. We also allow the following abbreviations:

$$\top \equiv \neg\bot, \quad \forall r.C \equiv \neg\exists r.\neg C, \quad C \sqcup D \equiv \neg(\neg C \sqcap \neg D).$$

An interpretation in $\mathcal{ALC}$ is a model $\mathcal{I} = \langle \Delta^{\mathcal{I}}, .^{\mathcal{I}} \rangle$ where the domain $\Delta^{\mathcal{I}}$ is a nonempty set and $.^{\mathcal{I}}$ is an interpretation function that assigns each concept symbol $A \in N_c$ to a subset of $\Delta^{\mathcal{I}}$ and each $r \in N_r$ to a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The constructs above are interpreted as follows:

$$\bot^{\mathcal{I}} := \emptyset, \quad \top^{\mathcal{I}} := \Delta^{\mathcal{I}}, \quad (\neg C)^{\mathcal{I}} := \Delta^{\mathcal{I}} \backslash C^{\mathcal{I}}$$
$$(C \sqcap D)^{\mathcal{I}} := C^{\mathcal{I}} \cap D^{\mathcal{I}}, \quad (C \sqcup D)^{\mathcal{I}} := C^{\mathcal{I}} \cup D^{\mathcal{I}}$$
$$(\exists r.C)^{\mathcal{I}} := \{x \in \Delta^{\mathcal{I}} | \exists y : (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$$
$$(\forall r.C)^{\mathcal{I}} := \{x \in \Delta^{\mathcal{I}} | \forall y : (x,y) \in r^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$$

A TBox, or an ontology, is a set of axioms of the forms $C \sqsubseteq D$ and $C \equiv D$, where C and D are concepts. The equivalence $C \equiv D$ is a short hand for $C \sqsubseteq D$ and $D \sqsubseteq C$. Without loss of generality we assume only ontologies consisting of subsumption axioms. $\mathcal{I}$ is model of an ontology $\mathcal{O}$ if all axioms 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 α be an $\mathcal{ALC}$ concept, we denote by $sig(\alpha)$ the set of concept and role names appearing in α. For an ontology $\mathcal{O}$, $sig(\mathcal{O})$ is the set of concept and role names appearing in its axioms. That is, $sig(\mathcal{O}) = \bigcup_{C \sqsubseteq D \in \mathcal{O}} sig(C) \cup sig(D)$.

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 name or role name $p \in \Sigma$.*

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 define $\mathcal{O}_1 \equiv^{\mathcal{M}}_{\Sigma} \mathcal{O}_2$ 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.*

Definition 3. *Let $\mathcal{O}$ be an $\mathcal{ALC}$ ontology and $\mathcal{F} \subseteq sig(\mathcal{O}) \cap N_c$ be the forgetting signature. The ontology $\mathcal{V}$ is a* semantic forgetting view *of $\mathcal{O}$ w.r.t. $\mathcal{F}$ iff the following hold,*

1. *$sig(\mathcal{V}) \subseteq sig(\mathcal{O}) \backslash \mathcal{F}$;*
2. *$\mathcal{O} \equiv^{\mathcal{M}}_{sig(\mathcal{O}) \backslash \mathcal{F}} \mathcal{V}$.*

Definition 4. *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 define $\mathcal{O}_1 \equiv^{C}_{\Sigma} \mathcal{O}_2$ iff for every $\mathcal{ALC}$ concept inclusion α, where $sig(\alpha) \subseteq \Sigma$, we have $\mathcal{O}_1 \models \alpha$ iff $\mathcal{O}_2 \models \alpha$.*

Definition 5. *Let $\mathcal{O}$ be an $\mathcal{ALC}$ ontology and $\mathcal{F} \subseteq sig(\mathcal{O}) \cap N_c$ be the forgetting signature. The ontology $\mathcal{V}$ is a* deductive forgetting view *of $\mathcal{O}$ w.r.t. $\mathcal{F}$ iff the following hold,*

1. *$sig(\mathcal{V}) \subseteq sig(\mathcal{O}) \backslash \mathcal{F}$;*
2. *$\mathcal{O} \equiv^{C}_{sig(\mathcal{O}) \backslash \mathcal{F}} \mathcal{V}$.*

The semantic view, as defined in Definition 3, is not in general representable in $\mathcal{ALC}$. Otherwise, the semantic view itself must be a consequence of the deductive view, then semantic and deductive forgetting would always coincide, which is not the case, e.g. (Koopmann and Schmidt 2014). To represent the semantic view using $\mathcal{ALC}$ language constructs, we relax the first condition in Definition 3 and extend the vocabulary of the semantic view with fresh concept symbols, hereafter called *definer symbols* or *definers* for short. We denote the set of introduced definers by N_d where N_d is disjoint with N_c and N_r. The following example shows the use of definers to represent the semantic view.

Example 1. *Let $\mathcal{O} = \{\top \sqsubseteq \exists r.B \sqcap \exists r.\neg B \sqcap \exists s.\neg B\}$, and $\mathcal{F} = \{B\}$. The deductive view of $\mathcal{O}$ with respect to $\mathcal{F}$ is $\mathcal{V}^{ded} = \{\top \sqsubseteq \exists r.\top \sqcap \exists s.\top\}$. However, $\mathcal{V}^{ded}$ does not preserve all information of r and s that is modelled in $\mathcal{O}$. For instance, the information that every domain element has two different r-successors, and that every domain element must have different r and s successors is not captured. With definer symbols, we can represent the semantic view as the ontology $\mathcal{V}^{sem} = \{\top \sqsubseteq \exists r.D_1, \top \sqsubseteq \exists r.D_2, \top \sqsubseteq \exists s.D_3, \top \sqsubseteq \neg D_1 \sqcup \neg D_2, \top \sqsubseteq \neg D_1 \sqcup \neg D_3\}$, where $D_1, D_2, D_3 \in N_d$.*

Example 2. *Let S be the subset of $\mathcal{V}^{sem}$ from Example 1 consisting only of the axioms $\{\top \sqsubseteq \exists r.D_1, \top \sqsubseteq \exists r.D_2, \top \sqsubseteq \exists s.D_3\}$. Comparing S with $\mathcal{V}^{ded}$, it can be shown that $S \equiv^{\mathcal{M}}_{\{r,s\}} \mathcal{V}^{ded}$. The remaining axioms $\Delta = \{\top \sqsubseteq \neg D_1 \sqcup \neg D_2, \top \sqsubseteq \neg D_1 \sqcup \neg D_3\}$ represent the information lost in $\mathcal{V}^{ded}$.*

A major advantage of the proposed method is that it introduces a fine grained forgetting framework where the user can append some of the lost information in Δ to $\mathcal{V}^{ded}$ as desired. In doing this, we envisage that Δ will be converted to useful warnings about information not captured in $\mathcal{V}^{ded}$.

Example 3. *Continuing with Example 2, the set Δ can be converted to the two warnings: $W_1 = \{\top \sqsubseteq \exists r.D_1, \top \sqsubseteq \exists r.D_2, \top \sqsubseteq \neg D_1 \sqcup \neg D_2\}$ and $W_2 = \{\top \sqsubseteq \exists r.D_1, \top \sqsubseteq \exists s.D_3, \top \sqsubseteq \neg D_1 \sqcup \neg D_3\}$. Recall the set $S = \{\top \sqsubseteq \exists r.D_1, \top \sqsubseteq \exists r.D_2, \top \sqsubseteq \exists s.D_3\}$. Suppose that the user is interested in the information related to r but not s, so she accepts W_1 and discards W_2. Accordingly the system appends S with W_1, and after eliminating D_3 from $S \cup W_1$ the final forgetting view would be $\mathcal{V} = \{\top \sqsubseteq \exists r.D_1, \top \sqsubseteq \exists r.D_2, \top \sqsubseteq \neg D_1 \sqcup \neg D_2, \top \sqsubseteq \exists s.\top\}$, which captures more information than the deductive view $\mathcal{V}^{ded}$ but less information than the semantic view $\mathcal{V}^{sem}$.*

Observe that $\mathcal{V}$ in the above example (1) captures more information related to r than $\mathcal{V}^{ded}$; and (2) uses fewer definer symbols than $\mathcal{V}^{sem}$. In this sense, the proposed forgetting model compromises between the simplicity of deductive forgetting and the expressivity of semantic forgetting.

4 Semantic Forgetting

The first stage of the method is to approximate the semantic forgetting view of the input ontology $\mathcal{O}$ w.r.t. some forgetting signature $\mathcal{F}$.

The method applies resolution to the input ontology in clausal form. We use the following steps to transform $\mathcal{O}$ into clausal form, $Clausal(\mathcal{O})$: (1) convert $\mathcal{O}$ to the negation normal form (NNF) such that negation is applied only to concept names, (2) apply structural transformation to extract formulas under role restriction that contain the forgetting symbols, and (3) convert the result to conjunctive normal form.

Example 4. *Consider the axiom $S = A \sqsubseteq \exists r.(B \sqcap C)$ where B is a forgetting symbol. S is first converted to NNF by eliminating the connective $\sqsubseteq$, so $S_1 = \neg A \sqcup \exists r.(B \sqcap C)$. Then, structural transformation is applied to extract $B \sqcap C$ which gives, $S_2 = \{\neg A \sqcup \exists r.D_1, \neg D_1 \sqcup (B \sqcap C)\}$ where $D_1 \in N_d$ is a definer symbol. Finally, S_2 is converted*

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

to conjunctive normal form giving $S_3 = (\neg A \sqcup \exists r.D_1) \sqcap (\neg D_1 \sqcup B) \sqcap (\neg D_1 \sqcup C)$.

In Example 4, D_1 can be seen as existentially quantified concept symbol. $Clausal(\{S\})$ can be viewed as the second order formula $\exists D_1(S_3)$. Observe, however, that S_3 is maintained in $\mathcal{ALC}$ syntax. Intuitively, D_1 represents the elements of $B \sqcap C$ that are reachable by r from A, that is, $\{y \in B^{\mathcal{I}} \cap C^{\mathcal{I}} | (x, y) \in r^{\mathcal{I}} \wedge x \in A^{\mathcal{I}}\}$ for any model $\mathcal{I}$ of S. In the rest of the paper, we will not differentiate between $Clausal(\mathcal{O})$ being a single CNF formula of the form $C_1 \sqcap \cdots \sqcap C_n$, being a set of clauses $\{C_1, \ldots, C_n\}$ where each clause is a disjunction of literals, or being an ontology $\{\top \sqsubseteq C_1, \top \sqsubseteq C_2, \ldots\}$.

Theorem 1. *For any ontology* $\mathcal{O}$, *we have* $\mathcal{O} \equiv^{\mathcal{M}}_{sig(\mathcal{O})} Clausal(\mathcal{O})$.

Forgetting is performed by resolving on the symbols in $\mathcal{F}$ iteratively using the resolution rule in Figure 1. When all possible resolution inferences have been performed on a concept symbol, clauses that contain this concept are removed in a *purity deletion* step. During the forgetting process the following operations are applied eagerly: (1) Tautology deletion: Clauses on the form $A \sqcup \neg A \sqcup C$ are deleted, where A and C are $\mathcal{ALC}$ concepts. (2) Subsumption deletion: Clauses on the form $C \sqcup D$ are deleted if another clause C is present in $\mathcal{O}$, where C, D are general concepts. (3) 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.

Example 5. *Let* $\mathcal{O} = \{A \sqsubseteq \forall r.B \sqcap \forall s.\neg B, G \sqsubseteq \exists r.(\neg B \sqcup C)\}$, *and* $\mathcal{F} = \{B\}$. *The method starts by generating* $\mathcal{O}^{clausal} = \{\neg A \sqcup \forall r.D_1, \neg A \sqcup \forall s.D_2, \neg G \sqcup \exists r.D_3, \neg D_1 \sqcup B, \neg D_2 \sqcup \neg B, \neg D_3 \sqcup \neg B \sqcup C\}$, *where* D_1, D_2, *and* D_3 *are fresh definer symbols. Then, it proceeds by resolving on the concept symbol* B. *This generates, additionally, the clauses* $\{\neg D_1 \sqcup \neg D_2, \neg D_1 \sqcup \neg D_3 \sqcup C\}$. *Finally, the clauses* $\{\neg D_1 \sqcup B, \neg D_2 \sqcup \neg B, \neg D_3 \sqcup \neg B \sqcup C\}$ *are removed by purity deletion. So the semantic forgetting view of* $\mathcal{O}$ *w.r.t. the forgetting signature* $\{B\}$ *is* $\mathcal{O}^{sem} = \{\neg A \sqcup \forall r.D_1, \neg A \sqcup \forall s.D_2, \neg G \sqcup \exists r.D_3, \neg D_1 \sqcup \neg D_2, \neg D_1 \sqcup \neg D_3 \sqcup C\}$.

Theorem 2. *Let* $\mathcal{O}$ *be an ontology, and let* $\mathcal{F}$ *be a forgetting signature. Let* $\mathcal{O}^{sem}$ *be the semantic view obtained by executing the described method. Then,* $\mathcal{O} \equiv^{\mathcal{M}}_{sig(\mathcal{O}) \setminus \mathcal{F}} \mathcal{O}^{sem}$.

Theorem 3. *The size of* $\mathcal{O}^{sem}$ *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 $|Clausal(\mathcal{O})| = m$ where the size of $Clausal(\mathcal{O})$ 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($Clausal(\mathcal{O})$, A)$| = \mathcal{O}(m^2)$. Repeating for k symbols we get $|$Forget($Clausal(\mathcal{O}), \mathcal{F}$)$| = \mathcal{O}(m^{2^k})$. 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 $Clausal(\mathcal{O})$ is dominated by conversion to CNF since the structural transformation is bounded by the number of role restrictions and NNF conversion does not add new axioms. It is known that the size of a CNF formula is, in the worst case, exponential in the size of original formula(Nonnengart and Weidenbach 2001). Therefore $m = \mathcal{O}(2^n)$. Altogether we get that $|$Forget($\mathcal{O}, \mathcal{F}$)$| = \mathcal{O}(2^{n.2^k})$. □

5 Reduction to Deductive Forgetting

The second stage of the method is to reduce the semantic view to a subset that is equivalent to the deductive view. The Red rule in Figure 2 removes from $\mathcal{O}^{sem}$ the clauses that contain two or more negative definers. As we will show, these removed clauses represent the information difference between the semantic and the deductive view.

The RP rule in Figure 2 preserves the $\mathcal{ALC}$ consequences that are otherwise lost by applying the Red rule. Therefore, we require it to be applied exhaustively before the Red rule. The premises of the RP 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, P_j takes the same form of P_0, i.e., a disjunction of negative definers, but also Definers(P_j) $\subseteq$ Definers(P_0) where Definers(P) means the set of definer symbols that are 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 $\bigsqcup_0^n E_i$ or $\mathcal{Q}r.(\bigsqcap_{i=0}^n D_i)$. But the later 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.

The following example shows the application of the RP rule.

Example 6. *Continuing with Example 5, the RP rule applies with the following premises:*

1. $P_0 \sqcup C_0 = \neg D_1 \sqcup \neg D_3 \sqcup C$
2. $\bigcup_{j=1}^{m} \{P_j \sqcup C_j\} = \emptyset$
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 generated conclusion is $\neg A \sqcup \neg G \sqcup \exists r.C$.

Theorem 4. *The conclusion of the* RP *rule in Figure 2, is a direct consequence of the premises.*

Role Propagation (RP)

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

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 symbol.

Reduction (Red)

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

Where C is a general concept expressions that does not contain a negative definer. $D_1, ..., D_n$ are definer symbols. The RP rule applies before this rule if $\neg D_1 \sqcup ... \sqcup \neg D_n$ takes the form of P_0 in the RP rule.

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

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

Following the application of the RP rule, the Red rule is applied. The clauses removed by the Red rule constitute the set Δ. The remaining clauses represent a subset equivalent to the deductive view. In the following, we mean by $\mathcal{O}^{sem}$ the semantic view obtained by forgetting $\mathcal{F}$ from $\mathcal{O}$ and then applying the RP rule exhaustively. By $\mathcal{O}^{red}$ we mean the result of applying the Red rule on $\mathcal{O}^{sem}$. This implies that $\mathcal{O}^{red} \subseteq \mathcal{O}^{sem}$.

Example 7. *Continuing with Example 6. Recall that* $\mathcal{O}^{sem} = \{\neg A \sqcup \forall r.D_1, \neg A \sqcup \forall s.D_2, \neg G \sqcup \exists r.D_3, \neg D_1 \sqcup \neg D_2, \neg D_1 \sqcup \neg D_3 \sqcup C, \neg A \sqcup \neg G \sqcup \exists r.C\}$. *By applying the Red rule we get* $\Delta = \{\neg D_1 \sqcup \neg D_2, \neg D_1 \sqcup \neg D_3 \sqcup C\}$, *and* $\mathcal{O}^{red} = \{\neg A \sqcup \forall r.D_1, \neg A \sqcup \forall s.D_2, \neg G \sqcup \exists r.D_3, \neg A \sqcup \neg G \sqcup \exists r.C\}$.

The following theorem says that $\mathcal{O}^{red}$ and $\mathcal{O}^{sem}$ (consequently by Theorem 2, also $\mathcal{O}$) have the same $\mathcal{ALC}$ consequences.

Theorem 5. $\mathcal{O}^{sem} \equiv^{C}_{sig(\mathcal{O}) \setminus \mathcal{F}} \mathcal{O}^{red}$.

The remainder of this section is dedicated to proving Theorem 5. The proof is presented in two steps. First, a necessary and sufficient condition for Theorem 5 is established in Theorem 7 below. The main benefit of this condition is the transformation of the entailment problem of Theorem 5 into a satisfiability problem. With this transformation, it is sufficient to show that any $\mathcal{ALC}$ concept inclusion α that is satisfiable with respect to $\mathcal{O}^{red}$ is also satisfiable with respect to $\mathcal{O}^{sem}$. Thus, we only need to construct an interpretation of $\mathcal{O}^{sem}$ that satisfies α. This interpretation is constructed in the second step of the proof. In this construction, we first analyse in Theorem 9 the models of $\mathcal{O}^{red}$ that satisfy α, then we construct using the notion of bisimulation (Kurtonina and de Rijke 1997; Stirling 1998) an interpretation of $\mathcal{O}^{sem}$ that satisfies α.

Definition 6. *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 $\mathcal{O}_2$.*

Theorem 6. *mDiff($\mathcal{O}^{sem}$, $\mathcal{O}^{red}$) = $\emptyset$, but in general mDiff($\mathcal{O}^{red}$, $\mathcal{O}^{sem}$) $\neq \emptyset$.*

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

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

Theorem 7. *Let $\mathcal{I}$ be a model in mDiff($\mathcal{O}^{red}$, $\mathcal{O}^{sem}$). That is, $\mathcal{I} \models \mathcal{O}^{red}$ but $\mathcal{I} \not\models \mathcal{O}^{sem}$. A necessary and sufficient condition for Theorem 5 is that, for any $\mathcal{ALC}$ concept inclusion α over $sig(\mathcal{O}) \setminus \mathcal{F}$ we have that $\mathcal{I} \models \alpha$ implies that there is a model $\mathcal{J}$ of $\mathcal{O}^{sem}$ such that $\mathcal{J} \models \alpha$.*

Proof. (1) Necessary: Assume that there is no such model $\mathcal{J}$ of $\mathcal{O}^{sem}$ such that $\mathcal{J} \models \alpha$. Therefore, $\mathcal{O}^{sem} \models \neg\alpha$. Since $\mathcal{O}^{sem} \equiv^{C}_{sig(\mathcal{O}) \setminus \mathcal{F}} \mathcal{O}^{red}$, it follows that $\mathcal{O}^{red} \models \neg\alpha$. This contradicts the assumption that $\mathcal{I} \models \mathcal{O}^{red}$ and $\mathcal{I} \models \alpha$.
(2) Sufficient: Assume $\mathcal{O}^{sem} \not\equiv^{C}_{sig(\mathcal{O}) \setminus \mathcal{F}} \mathcal{O}^{red}$. Therefore, there must be an $\mathcal{ALC}$ concept inclusion α over $sig(\mathcal{O}) \setminus \mathcal{F}$ such that:

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

Consider the first case. There must be a model $\mathcal{I} \in$ mDiff($\mathcal{O}^{red}$, $\mathcal{O}^{sem}$) such that $\mathcal{I} \models \neg\alpha$. But then by hypothesis there is a model $\mathcal{J}$ of $\mathcal{O}^{sem}$ such that $\mathcal{J} \models \neg\alpha$, which contradicts the assumption that $\mathcal{O}^{sem} \models \alpha$. Now consider the second case. Since $\mathcal{O}^{sem} \not\models \alpha$, there must be a model $\mathcal{J}$ of $\mathcal{O}^{sem}$ such that $\mathcal{J} \not\models \alpha$. By Theorem 6, $\mathcal{J}$ is also a model of $\mathcal{O}^{red}$ which contradicts the assumption that $\mathcal{O}^{red} \models \alpha$. □

We now move to the second step of proving Theorem 5. Let $\mathcal{I}$ be any model in mDiff($\mathcal{O}^{red}, \mathcal{O}^{sem}$) and assume there is an $\mathcal{ALC}$ concept inclusion α over $sig(\mathcal{O})\backslash\mathcal{F}$, such that $\mathcal{I} \models \alpha$ and α is not satisfiable with respect to $\mathcal{O}^{sem}$. Recall that our aim is to contradict this assumption by constructing a model $\mathcal{J}$ of $\mathcal{O}^{sem}$ such that $\mathcal{J} \models \alpha$.

Definition 7. *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. There is a transition, or an edge, from d to e iff $(d, e) \in r^{\mathcal{I}}$ where $r \in N_r$.*

Definition 8. *Let $(\mathcal{I}, d_1)$ and $(\mathcal{J}, d_2)$ be two pointed interpretations, and Σ be some signature. $(\mathcal{I}, d_1), (\mathcal{J}, d_2)$ are Σ-bisimilar, $(\mathcal{I}, d_1) \sim_\Sigma (\mathcal{J}, d_2)$, 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 $(e, e') \in R$.*

Theorem 8. *(Lutz and Wolter 2011) 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 $(\mathcal{I}, d_1) \equiv_\Sigma (\mathcal{J}, d_2)$.*

Let α be of the form $C \sqsubseteq E$ where C and E are arbitrary. Without loss of generality, assume that $\mathcal{I}$ is a pointed interpretation generated by an arbitrary $d \in C^{\mathcal{I}}$.

Theorem 9. *Suppose $\mathcal{O}^{sem} = \mathcal{O}^{red} \cup \{\neg D_1 \sqcup ... \sqcup \neg D_n \sqcup F\}$ where $n > 1$. Let $(\mathcal{I}, d)$ be as above. Then, $\mathcal{I} \in$ mDiff($\mathcal{O}^{red}, \mathcal{O}^{sem}$) iff there is $e \in \Delta^{\mathcal{I}}$ that is reachable from d where $e \in D_1^{\mathcal{I}} \cap ... \cap D_n^{\mathcal{I}}$ and $e \notin F^{\mathcal{I}}$.*

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

Theorem 9 sets our aim to construct a pointed interpretation $(\mathcal{J}, d)$ in which no such e exists. As said earlier, our target is to construct $(\mathcal{J}, d)$ such that $(\mathcal{I}, d) \sim_{sig(\mathcal{O}\backslash\mathcal{F}} (\mathcal{J}, d)$. Observe that the theorem assumes that the difference between $\mathcal{O}^{sem}$ and $\mathcal{O}^{red}$ is the single clause $\neg D_1 \sqcup ... \sqcup \neg D_n \sqcup F$. If both ontologies differ on several clauses $\neg D_1^i \sqcup ... \sqcup \neg D_n^i \sqcup F^i$ where $1 \leq i \leq n$, we construct a series of ontologies $\mathcal{O}^i$ where $\mathcal{O}^{red} = \mathcal{O}^0$, $\mathcal{O}^{sem} = \mathcal{O}^n$, and $\mathcal{O}^i = \mathcal{O}^{i-1} \cup \{\neg D_1^i \sqcup ... \sqcup \neg D_n^i \sqcup F^i\}$. Then we construct a series of models $\mathcal{J}^i$ of $\mathcal{O}^i$ where $\mathcal{J}^n = \mathcal{J}$ and have:

$$(\mathcal{I}, d) \sim_\Sigma (\mathcal{J}^1, d) \sim_\Sigma \cdots \sim_\Sigma (\mathcal{J}^n, d) \quad (1)$$

It follows, by the transitivity of $\sim_\Sigma$, that $(\mathcal{I}, d) \sim_\Sigma (\mathcal{J}, d)$ and $\mathcal{J} \models \mathcal{O}^{sem}$.

Definition 9. *Let $\mathcal{I}$ be a model and e be a domain element 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 concept names and definer symbols in $N_c \cup N_d$ that contain e in their interpretation.*

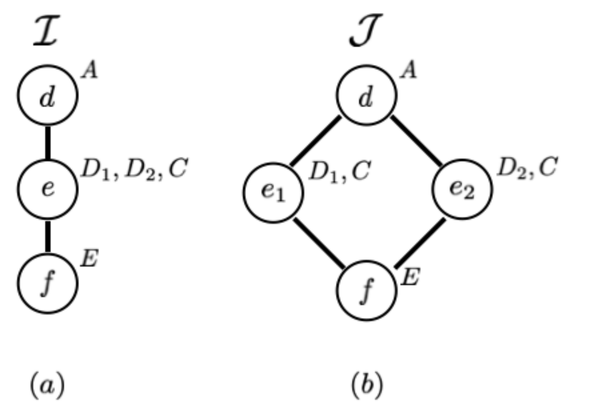

Figure 3: Construction of $\mathcal{J}$ from $\mathcal{I}$. Each node contains a domain element. An edge between elements a and b means that $r(a, b) \in r^{\mathcal{I}}$. A list of concept names whose interpretation includes the element appears at the top right of the element. For readability, negated concept names are omitted from this list. For instance, in the graph of $\mathcal{J}$, the element e_1 is understood as being a member in the interpretation of $D_1, C, \neg A, \neg E, \neg D_2$

Theorem 10. *Let $\mathcal{O}^{sem}$, $\mathcal{O}^{red}$, and $\mathcal{I}$ be defined as in Theorem 9. There is a model $\mathcal{J}$ generated by d such that:*

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

Proof. We start with an example to give the intuition of the construction method of $\mathcal{J}$. Suppose $\mathcal{O}^{sem} = \{\neg A \sqcup \exists r.D_1, \neg A \sqcup \exists r.D_2, \neg D_1 \sqcup C, \neg D_2 \sqcup C, \neg D_1 \sqcup \neg D_2, \neg C \sqcup \exists r.E\}$, and $\mathcal{O}^{red} = \mathcal{O}^{sem} \backslash \{\neg D_1 \sqcup \neg D_2\}$. Let $\Sigma = \{A, C, E, r\}$. $\mathcal{O}^{red}$ tells that every domain element in A is related via r to elements of C, and every element in C is related via r to an element of E. An example of a model $\mathcal{I}$ of $\mathcal{O}^{red}$ is shown in Figure 3(a). Clearly $\mathcal{I} \not\models \mathcal{O}^{sem}$ since $\mathcal{O}^{sem}$ requires that every domain element of A is related via r to two disjoint subsets of C. A way to transform $\mathcal{I}$ into a model $\mathcal{J}$, such that the conditions of the theorem are satisfied, is to replace e with two fresh domain elements e_1 and e_2. The key requirements are that both e_1 and e_2 are elements of C, and neither of them is an element of D_1 and D_2 at the same time. The model $\mathcal{J}$ is shown in Figure 3(b). It is straight forward to see that $(\mathcal{I}, d) \sim_\Sigma (\mathcal{J}, d)$, and $\mathcal{J} \models \neg D_1 \sqcup \neg D_2$.

We now give a formal proof as follows: By Theorem 9, there is $e \in \Delta^{\mathcal{I}}$ that is reachable from d where $e \in D_1^{\mathcal{I}} \cap ... \cap D_n^{\mathcal{I}}$ and $e \notin F^{\mathcal{I}}$. Transform $\mathcal{I}$ into $\mathcal{J}$ by replacing e with fresh domain elements e_i where $1 \leq i \leq n$, such that $e_i \notin D_i, e_i \in D_j^{\mathcal{J}}$ where $1 \leq j \leq n$ and $j \neq i$, and:

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

where $r \in N_r$, and $\{e', e''\} \subset \Delta^{\mathcal{J}}$. Assume that e, and e_i, are reachable from d in k transitions. We prove the theorem by induction from bottom to top. By construction, for any node $u \in (\mathcal{I}, d)$ or $(\mathcal{J}, d)$ such that u is reachable from d in at least $k+1$ transitions we have $(\mathcal{I}, u) \sim_\Sigma (\mathcal{J}, u)$. That is, nothing below e and e_i changed. Second, for every $u \in (\mathcal{I}, d)$ at depth k we have: (1) If $u = e$, then $\mathcal{C}_\mathcal{I}(u) \backslash \{D_i\} = \mathcal{C}_\mathcal{J}(e_i)$. Also, since the third condition of the transformation guarantees that every thing below u and e_i is the same, it follows that $(\mathcal{I}, u) \sim_\Sigma (\mathcal{J}, e_i)$. (2) If $u \neq e$, then u is also a node in $(\mathcal{J}, d)$ and $(\mathcal{I}, u) \sim_\Sigma (\mathcal{J}, u)$. Finally, for every $u \in (\mathcal{I}, d)$ such that u is reachable in at most $k-1$ transitions from d we have that $u \in (\mathcal{J}, d)$, and we have the following cases: (1) $e \notin (\mathcal{I}, u)$. Then, as before $(\mathcal{I}, u) \sim_\Sigma (\mathcal{J}, u)$. (2) $e \in (\mathcal{I}, u)$. Condition 1 in the transformation guarantees that $\mathcal{C}_\mathcal{I}(u) = \mathcal{C}_\mathcal{J}(u)$ on all $C \in (N_c \cup N_d)$. Suppose that $u \in (\exists \vec{r}.C)^\mathcal{I}$ where $\exists \vec{r} = \exists r_1.\exists.r_2....\exists.r_m$, $r_i \in N_r$, and C is any $\mathcal{ALC}$ concept over Σ. Then there must be a path $ur_1u_1r_2u_2r_3...r_mu_m$ in $(\mathcal{I}, d)$ such that $u_m \in C^\mathcal{I}$. If $u_i \neq e$ for every $i \in [1..m]$ then by construction this path also exists in $(\mathcal{J}, d)$ and thus $u \in (\exists \vec{r}.C)^\mathcal{J}$. If $u_i = e$ for any $i \in [1..m]$, then again by construction there is a path $ur_1u_1r_2...r_ivr_{i+1}...r_mu_m$ and we have the choice to set v to any of $\{e_1, ..., e_n\}$. In particular, we have $u_m \in C^\mathcal{J}$. Thus, $u \in (\exists \vec{r}.C)^\mathcal{J}$. We get that $(\mathcal{I}, u) \sim_\Sigma (\mathcal{J}, u)$ for every u reachable in at most $k-1$ transitions from d. The same argument can be used to show that every node in $(\mathcal{J}, d)$ has a bisimilar node in $(\mathcal{I}, d)$. Altogether, we get that $(\mathcal{I}, d) \sim_\Sigma (\mathcal{J}, d)$, and $\mathcal{J} \not\models D_1 \sqcap D_2 \sqcap ... \sqcap D_n \sqcap \neg F$. □

We conclude this section with a discussion of the Red rule. The clauses excluded by the Red rule represent the information difference between $\mathcal{O}^{sem}$ and $\mathcal{O}^{red}$. As illustrated by Example 3, we envisage that these axioms are presented to the user as warnings of information loss. However, to be meaningful, *context* axioms must also be included. For instance, in Example 3, the axioms extracted by the Red rule are $\{\top \sqsubseteq \neg D_1 \sqcup \neg D_2, \top \sqsubseteq \neg D_1 \sqcup \neg D_3\}$. When presented as warnings W_1 and W_2, they are accompanied with $\{\top \sqsubseteq \exists r.D_1, \top \sqsubseteq \exists r.D_2\}$ and $\{\top \sqsubseteq \exists r.D_1, \top \sqsubseteq \exists r.D_3\}$ respectively. This can be realized easily by collecting from the semantic view the clauses that contain the definer symbols positively. For instance, to present $\top \sqsubseteq \neg D_1 \sqcup \neg D_2$ as a warning, we collect from the semantic view the clauses that contain D_1 and D_2 and present them as additional axioms in the warning.

6 Eliminating the Definer Symbols

We identified Δ as the axioms that contain two or more negative definers. The semantic and the deductive views may contain other definer symbols that appear negatively in clauses, possibly together with positive definers, where no other negative definer is present. These definers can be eliminated safely while preserving the interpretations of the keep signature. For this, we use the DE rule in Figure 4. The side conditions of the DE rule also restrict eliminating definers that may appear both positively and negatively in a clause. This case signifies cycles in the original ontology that can only be expressed using fixpoint operators (Nonnengart and Szalas 1998). As an alternative, (Koopmann and Schmidt 2013b) proposed leaving the definer symbols in the deductive view as witnesses of these cycle. We find this the best option because it defers the decision of the correct representation to a later stage. For clauses that contain only one negative definer symbol, possibly with other positive definers, the DE rule in Figure 4 is applied exhaustively. The rule replaces the definer symbol D with its super-concept $C_1 \sqcap ... \sqcap C_n$. Note that, in the DE rule, C may be the $\bot$ concept.

Non-cyclic Definer Elimination (DE)

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

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

Figure 4: Definer elimination rule

Theorem 11. *Let $\mathcal{O}$ be an ontology, $\mathcal{F}$ a forgetting signature. Also let $\mathcal{O}^{red}$ be generated as described before, and $\mathcal{V}$ be generated from $\mathcal{O}^{red}$ by applying the DE rule from Figure 4 exhaustively. Then:*

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

Proof. First we prove (1). The DE rule in Figure 4 can be seen as a two steps 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 is clearly equivalence preserving. The second replaces every D in $\mathcal{O}$ with the concept C. This step is the inverse of the structural transformation. Therefore, by Theorem 1 we get that $\mathcal{O}^{red} \equiv^\mathcal{M}_{sig(\mathcal{V})} \mathcal{V}$.

Second we prove (2). By Theorem 2, we get that $\mathcal{O}$ and $\mathcal{O}^{sem}$ coincide on all interpretations of the $sig(\mathcal{O})\backslash\mathcal{F}$. It follows that both $\mathcal{O}$ and $\mathcal{O}^{sem}$ agree on all $\mathcal{ALC}$ concept inclusions over $sig(\mathcal{O})\backslash\mathcal{F}$. By Theorem 5, both $\mathcal{O}^{red}$ and $\mathcal{O}^{sem}$, consequently also $\mathcal{O}$, 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. $\mathcal{F}$. □

7 Evaluation

We implemented a prototype of our method based on *Java 12* and *OWL 5.1.11*, and evaluated it against *Lethe* (version 2.11-0.026)[1] (Koopmann and Schmidt 2013a). Lethe is a Scala based implementation of the deductive forgetting method in (Koopmann and Schmidt 2013b). Similar to our method, Lethe performs forgetting on a transformation of the ontology in a second order language. However, it uses a direct resolution calculus to compute the deductive view.

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

It is traditional to benchmark different forgetting methods by assigning to them one or more forgetting tasks on a popular ontology, and compare their performance in each task. However, this does not guarantee challenging forgetting problems. Forgetting is more complex if forgetting symbols appear together frequently in the ontology's axioms. This criterion is not satisfied in most key ontologies. For instance, the most frequently occurring concept symbol in the axioms of the *National Cancer Institute Thesaurus (NCIT)* ontology (Fragoso et al. 2004), occurs in just 1.5% of the axioms. So, the probability of two or more forgetting symbols appearing in the same axiom is low.

Instead, we built our own ontologies. The idea is simple, different biomedical ontologies model different information related to a given biomedical term. However, these subsets of information are likely formalized using the same biomedical terms. For instance, if two ontologies use the medical term *spinal-cord* in their axioms, they are also likely to use the term *spine* in some of these axioms. So if we group in a new ontology, the axioms of each ontology that contain the term spinal-cord, then we can use the forgetting signature {spinal-cord, spine} and be more confident that both will appear together in many axioms. To implement the described idea, we construct each ontology in our dataset in the following way:

1. Collect a set of random concept names Σ from the *Interlinking Ontology for Biological Concepts (IOBC)* ontology (Kushida et al. 2019).
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 where the symbols in Σ appear.

In total, 22 ontologies were downloaded from the Bioportal, and then used as described above to construct 90 ontologies. In each construction, at least 25% of the input ontologies were expressed in a more complex logic than $\mathcal{ALC}$.

The constructed ontologies were divided into three groups: *Low*, *Moderate*, and *High*, with 30 ontologies in each group. Each group was used in a different experimental setting. In the Low setting, the forgetting symbols appear in at least half of the axioms, and the probability of two forgetting symbols appearing in the same axiom is low. In the Moderate setting, the forgetting symbols appear in all of the axioms, and at least one axiom contains two forgetting symbols. In the High setting, the forgetting symbols appear in all of the axioms, and at least half of the axioms contain two forgetting symbols.

The experiments were performed on a x64-based processor Intel(R) Core(TM) i5-8350U CPU @ 1.70GH, running a 64-bit operating system (Windows 10). Each experiment was allocated 128MB of memory to run. The experiment computes two versions of the deductive forgetting view, one by our prototype and one by Lethe. We verified the correctness of our method by comparing both versions using a reasoner. Manual inspection was also required when the reasoner detected differences due to definer symbols present as witnesses to cycles but named differently in each view.

Group	Prototype T.O.	Lethe T.O.	Outliers
Low	0	0	0
Moderate	0	1	3
High	3	0	3

Table 1: Timeouts and outliers

Measure	Low	Moderate	High
Mean	0.50	0.36	0.55
Std. Dev.	0.32	0.70	0.40

Table 2: Mean and Standard Deviation of time gain over Lethe.

The timeout for each experiment was thirty minutes. Table 1 shows the number of timeout runs and *outlier* experiments. The latter are experiments where the difference between the time consumed by Lethe and by our prototype was very high. We calculated the outliers based on the 90th percentile after excluding the timeouts. That is, we discarded as outliers the highest 10% according to the calculated time difference between our prototype and Lethe. Table 1 shows an uptrend in the discarded experiments, due to timeout and outliers, as the forgetting problems gets harder. To understand the reason behind the uptrend, we investigated the time ratio between the computation of the semantic view, and the reduction to the deductive view. Figure 6(a) shows a significant increase in this ratio as we move towards the High setting. Therefore, the uptrend of the timeouts and outliers is mainly due to the time consumed by the semantic forgetting stage. This is noticeable with our prototype and not Lethe because unlike Lethe, optimization techniques, such as subsumption deletion, are not implemented in our prototype. These optimization techniques eagerly reduce the size of the semantic view and consequently the number of required resolutions. The effect of their absence becomes more apparent as the size of the input ontology and the hardness of the problem increase. We observed that the sizes of the ontologies in the discarded experiments, were 303 axioms in the Moderate setting and 330 axioms in the High setting. Whereas the average sizes of all ontologies participating in the experiments of both settings were 160 and 75 respectively.

Figure 5 shows the normal distribution of the time performance gain over Lethe, which is calculated using:

$$\text{Time gain} = (\text{Lethe} - \text{Proposed method})/\text{Lethe}. \quad (2)$$

The graphs in Figure 5 are based on the mean and standard deviation values in Table 2. The mean values show $50\%, 36\%$, and 55% improvements in processing times compared to Lethe. However, the standard deviation of the Moderate setting is relatively high. This means a high variance in the time gain across the experiments in the Moderate setting compared to the other settings. The reasons of this observation are: (1) The average size of the ontologies in the Moderate setting is 160 compared to 116 and 75 in the Low and High settings. Thus, the time gain in the Moderate setting is more variable because of the lack in optimization

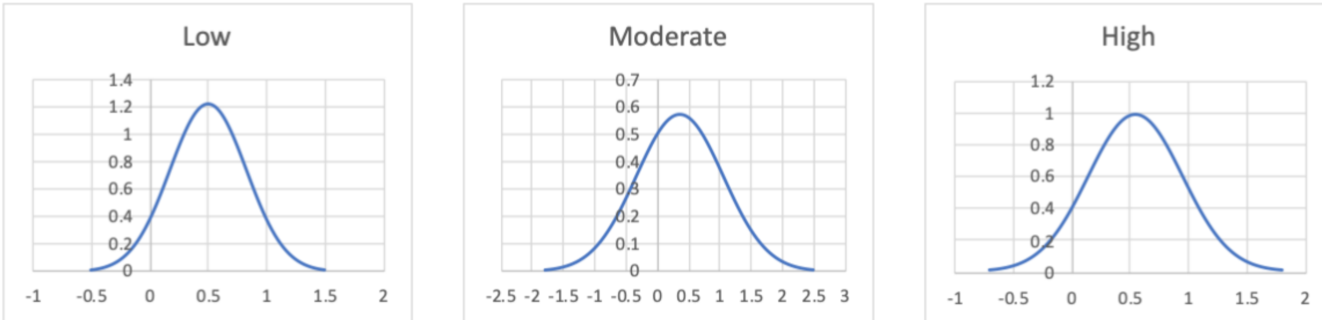

Figure 5: Normal distribution of time gain in the Low, Moderate, and High settings. The horizontal axis represents the time performance gain calculated as in (2). The vertical axis represents normal distribution values according to data in Table 2.

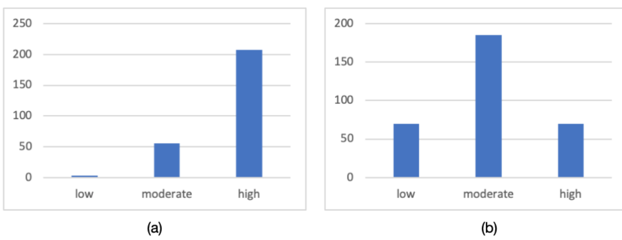

Figure 6: Diagram (a) on the left shows the ratio between semantic forgetting and reduction time. Diagram (b) on the right shows the average size of Δ across the three settings.

	Average	Median	75 Percentile
Definers	16	0.0	9.3

Table 3: Average, median, and 75 Percentile values of the number of non-cyclic definers in the semantic forgetting view.

techniques. (2) As Figure 6(b) shows, the average size of Δ is 185 axioms in the Moderate setting, compared to 70 and 69 in the Low and High settings. So more resolution steps are performed in the Moderate setting to compute this Δ. To determine the relative weight of both reasons, we inspected the experiments that contributed most to the standard deviation. The standard deviation was distorted mainly due to a single experiment. The size of the ontology in these experiment is 526 which is very high compared to the average size in the Moderate setting. Also, the size of Δ in this experiment is 2132 axioms, which is very high. By excluding this experiment, the mean and the standard deviation were adjusted to 0.48 and 0.43, which is inline with the measurements in other settings.

Table 3 shows statistical values of the number of the non-cyclic definer symbols in the semantic view. That is, the definers that are not present to witness cycles as explained in Section 6. The values are computed over all experiments because the number of non-cyclic definers is controlled by the semantics and the modelling of the input ontology rather than the number of forgetting symbols appearing in the same axioms. Thus, the Low, Moderate, and High settings, which are designed mainly for performance testing, are not relevant to the analysis of these definers. To understand these values, we compare them to the average size of 16.6 of the forgetting symbols over all experiments. While this number is very close to the average number of introduced definers in Table 3; the median value of 0.0 implies that in at least half of the experiments the semantic view did not have non-cyclic definers in its vocabulary. Similarly, the 75 Percentile value of 9.3 implies that in 75% of the experiments, approximately 9 non-cyclic definer symbols were introduced in the semantic view compared to approximately 17 forgotten symbols.

8 Conclusions and Future Work

We presented a new forgetting method that performs semantic and deductive forgetting in one system. The method also extracts a set Δ of information differences between both views. While the deductive view can be represented in $\mathcal{ALC}$, the semantic view and the set Δ are represented using additional existentially quantified concept symbols. Appending the deductive view with information from Δ is a compromise between the simplicity of the deductive view and the expressivity of the semantic view since the final view contains only the second order symbols required to represent the incremented information. This offers a more fine-grained forgetting method that gives more control over the information modelled in the forgetting view, and reveals, in terms of the modelled information, the differences between both views. The evaluation showed improvement in processing time over the current state-of-the-art despite the two stage process.

One novelty of our forgetting method is the new representation language of the semantic view. Although the aim is to minimize the number of definers, in general, definers cannot be avoided in the semantic view. Further minimization is expected after integrating optimization techniques such as subsumption deletion with our method. It is interesting that our fine-grained forgetting method shows the existence of a spectrum of forgetting variants inbetween the deductive and the semantic forgetting variants. More understanding of this spectrum is important. We intend to address both these points in future work.

References

Ackermann, W. 1935. Untersuchungen über das Eliminationsproblem der mathematischen logik. *Mathematische Annalen* 110:390–413.

Boole, G. 1854. *An Investigation of the Laws of Thought: On Which Are Founded the Mathematical Theories of Logic and Probabilities*. Cambridge University Press.

Botoeva, E.; Konev, B.; Lutz, C.; Ryzhikov, V.; Wolter, F.; and Zakharyaschev, M. 2017. 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 2016*. Springer. 27–89.

Calvanese, D.; De Giacomo, G.; and Lenzerini, M. 1999. Reasoning in Expressive Description Logics with Fixpoints Based on Automata on Infinite Trees. In *Proceedings of the 16th International Joint Conference on Artifical Intelligence*, IJCAI 1999, 84–89. Morgan Kaufmann.

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

D'Agostino, G., and Hollenberg, M. 1996. Uniform interpolation, automata and the modal μ-calculus. *Logic Group Preprint Series* 165.

Del-Pinto, W., and Schmidt, R. A. 2019. ABox abduction via forgetting in ALC. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, AAAI 2019, 2768–2775. AAAI Press.

Delgrande, J. P. 2014. Towards a Knowledge Level Analysis of Forgetting. In *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning*, KR 2014, 606–609. AAAI Press.

Delgrande, J. 2017. A Knowledge Level Account of Forgetting. *Journal of Artificial Intelligence Research* 60:1165–1213.

Ditmarsch, H.; Herzig, A.; Lang, J.; and Marquis, P. 2008. Introspective forgetting. In *Proceedings of the 21st Australasian Joint Conference on Artificial Intelligence*, AI 2008, 18–29. Springer.

Divroodi, A. R., and Nguyen, L. A. 2015. On bisimulations for description logics. *Information Sciences* 295:465–493.

Eiter, T., and Kern-Isberner, G. 2019. A Brief Survey on Forgetting from a Knowledge Representation and Reasoning Perspective. *KI - Künstliche Intelligenz* 33(1):9–33.

Erdem, E., and Ferraris, P. 2007. Forgetting Actions in Domain Descriptions. In *Proceedings of the 22nd National Conference on Artificial Intelligence*, AAAI 2007, 409–414. AAAI Press.

Fragoso, G.; Coronado, S.; Haber, M.; Hartel, F.; and Wright, L. 2004. Overview and utilization of the nci thesaurus. *Comparative and functional genomics* 5:648–54.

French, T. 2006. *Bisimulation quantifiers for modal logics*. Ph.D. Dissertation, The University of Western Australia.

Gabbay, D. M., and Ohlbach, H. J. 1992. Quantifier elimination in second-order predicate logic. In *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning*, KR 1992, 425–435. Morgan Kaufmann.

Gabbay, D. M.; Schmidt, R. A.; and Szałas, A. 2008. *Second-Order Quantifier Elimination: Foundations, Computational Aspects and Applications*. College Publications.

Ghilardi, S. 1995. An algebraic theory of normal forms. *Annals of Pure and Applied Logic* 71(3):189 – 245.

Grau, B., and Motik, B. 2010. Pushing the limits of reasoning over ontologies with hidden content. In *Proceedings of the 12th International Conference on the Principles of Knowledge Representation and Reasoning*, KR 2010, 214–224. AAAI Press.

Herzig, A., and Mengin, J. 2008. Uniform interpolation by resolution in modal logic. In *Proceedings of the 11th European Conference on Logics in Artificial Intelligence*, volume 5293 LNAI, 219–231. Springer.

Konev, B.; Lutz, C.; Walther, D.; and Wolter, F. 2013. Model-theoretic inseparability and modularity of description logic ontologies. *Artificial Intelligence* 203:66–103.

Konev, B.; Walther, D.; and Wolter, F. 2008. The Logical Difference Problem for Description Logic Terminologies. In *Proceedings of the 4th International Joint Conference on Automated Reasoning*, IJCAR 2008, 259–274. Springer.

Konev, B.; Walther, D.; and Wolter, F. 2009. Forgetting and uniform interpolation in large-scale description logic terminologies. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, IJCAI 2009, 830–835. Morgan Kaufmann.

Koopmann, P., and Schmidt, R. A. 2013a. Implementation and evaluation of forgetting in ALC-ontologies. In *Proceedings of the 7th International Workshop on Modular Ontologies (WoMo 2013)*, volume 1081. CEUR-WS.org.

Koopmann, P., and Schmidt, R. A. 2013b. Uniform interpolation of $\mathcal{ALC}$-ontologies using fixpoints. In *Proceedings of the 9th International Symposium on Frontiers of Combining Systems (FroCoS 2013)*, volume 8152 of *Lecture Notes in Artificial Intelligence*, 87–102. Springer.

Koopmann, P., and Schmidt, R. A. 2014. Count and forget: Uniform interpolation of $\mathcal{SHQ}$-ontologies. In *Automated Reasoning (IJCAR 2014)*, volume 8562 of *Lecture Notes in Artificial Intelligence*, 434–448. Springer.

Koopmann, P., and Schmidt, R. A. 2015. Saturation-based forgetting in the description logic $\mathcal{SIF}$. In *Proceedings of the 28th International Workshop on Description Logics (DL-2015)*, volume 1350 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Kurtonina, N., and de Rijke, M. 1997. Classifying description logics. In *Proceedings of the 1997 Description Logic Workshop*, DL 1997, 49–53. Université Paris-Sud.

Kushida, T.; Kozaki, K.; Kawamura, T.; Tateisi, Y.; Yamamoto, Y.; and Takagi, T. 2019. Interconnection of biological knowledge using NikkajiRDF and interlinking ontology for biological concepts. *New Generation Computing* 37:1–25.

Lakemeyer, G. 1997. Relevance from an Epistemic Perspective. *Journal of Artificial Intelligence* 97(1–2):137–167.

Lang, J., and Marquis, P. 2010. Reasoning under inconsistency: A forgetting-based approach. *Artificial Intelligence* 174(12):799–823.

Lang, J.; Liberatore, P.; and Marquis, P. 2003. Propositional independence: Formula-variable independence and forgetting. *Journal of Artificial Intelligence Research* 18:391–443.

Lin, F., and Reiter, R. 1994a. Forget it. In *Working Notes of AAAI Fall Symposium on Relevance*, 154–159.

Lin, F., and Reiter, R. 1994b. How to progress a database (and why) i. logical foundations. In *Proceedings of the Fourth International Conference on Principles of Knowledge Representation and Reasoning*, KR 1994. Morgan Kaufmann. 425 – 436.

Ludwig, M., and Konev, B. 2013. Towards practical uniform interpolation and forgetting for $\mathcal{ALC}$ TBoxes. In *Proceedings of the 26th International Workshop on Description Logics*, DL 2013. AAAI Press.

Ludwig, M., and Konev, B. 2014. Practical uniform interpolation and forgetting for $\mathcal{ALC}$ TBoxes with applications to logical difference. In *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning*, KR 2014. AAAI Press.

Lutz, C., and Wolter, F. 2010. Deciding inseparability and conservative extensions in the description logic $\mathcal{EL}$. *J. Symb. Comput.* 45:194–228.

Lutz, C., and Wolter, F. 2011. Foundations for uniform interpolation and forgetting in expressive description logics. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, IJCAI 2011, 989–995.

Nonnengart, A., and Szalas, A. 1998. A fixpoint approach to second-order quantifier elimination with applications to correspondence theory. In *Logic at Work: Essays Dedicated to the Memory of Helena Rasiowa (1999)*. Springer Physica-Verlag.

Nonnengart, A., and Weidenbach, C. 2001. Computing small clause normal forms. In *Handbook of Automated Reasoning*. North-Holland. 335 – 367.

Stirling, C. 1998. The joys of bisimulation. In *Mathematical Foundations of Computer Science 1998*, 142–151. Springer.

Subramanian, D., and Genesereth, M. R. 1987. The relevance of irrelevance. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, IJCAI 1987, 416–422.

Visser, A. 1996. *Uniform interpolation and layered bisimulation*, volume 6 of *Lecture Notes in Logic*. Springer-Verlag. 139–164.

Wang, Z.; Wang, K.; Topor, R.; and Pan, J. Z. 2010. Forgetting for knowledge bases in DL-Lite. *Annals of Mathematics and Artificial Intelligence* 58(1-2):117–151.

Zhang, Y., and Zhou, Y. 2009. Knowledge forgetting: Properties and applications. *Artificial Intelligence* 173:1525–1537.

Zhang, Y., and Zhou, Y. 2010. Forgetting revisited. In *Proceedings of the Twelfth International Conference on the Principles of Knowledge Representation and Reasoning*, KR 2010. AAAI Press.

Zhao, Y., and Schmidt, R. A. 2015. Concept forgetting in $\mathcal{ALCOI}$-ontologies using an Ackermann approach. In *Proceedings of the 14th International Semantic Web Conference, ISWC 2015*, volume 9366 of *Lecture Notes in Computer Science*, 587–602. Springer.