Role Forgetting for $\mathcal{ALCOQH}(\triangledown)$-Ontologies Using an Ackermann-Based Approach

Yizheng Zhao and **Renate A. Schmidt**
School of Computer Science, The University of Manchester, UK
{yizheng.zhao, renate.schmidt}@manchester.ac.uk

Abstract

Forgetting refers to a non-standard reasoning problem concerned with eliminating concept and role symbols from description logic-based ontologies while preserving all logical consequences up to the remaining symbols. Whereas previous research has primarily focused on forgetting concept symbols, in this paper, we turn our attention to role symbol forgetting. In particular, we present a practical method for semantic role forgetting for ontologies expressible in the description logic $\mathcal{ALCOQH}(\triangledown)$, i.e., the basic description logic $\mathcal{ALC}$ extended with nominals, qualified number restrictions, role inclusions and the universal role. Being based on an Ackermann approach, the method is the only approach so far for forgetting role symbols in description logics with qualified number restrictions. The method is goal-oriented and incremental. It always terminates and is sound in the sense that the forgetting solution is equivalent to the original ontology up to the forgotten symbols, possibly with new concept definer symbols. Despite our method not being complete, performance results of an evaluation with a prototypical implementation have shown very good success rates on real-world ontologies.

1 Introduction

The origins of interest in forgetting can be traced back to the work of Boole on propositional variable elimination and the seminal work of Ackermann [Ackermann, 1935] who recognized that the problem amounts to the elimination of existential second-order quantifiers. In logic the problem has been studied as the (dual) uniform interpolation problem [Visser, 1996; D'Agostino and Hollenberg, 2000; Herzig and Mengin, 2008], a notion related to the Craig interpolation problem, but stronger. In computer science the importance of forgetting can be found in the knowledge representation literature [Lin and Reiter, 1994; Lang *et al.*, 2003; Delgrande and Wang, 2015], specification refinement literature [Bicarregui *et al.*, 2001] and the area of description logic-based ontology engineering [Botoeva *et al.*, 2016; Wang *et al.*, 2008; 2010; 2014; Konev *et al.*, 2009a; 2009b; 2013; Lutz and Wolter, 2011; Lutz *et al.*, 2012; Grau and Motik, 2012; Ludwig and Konev, 2014; Nikitina and Rudolph, 2014]. In ontology-based information processing, forgetting allows users to focus on specific parts of ontologies in order to create decompositions and restricted views for in depth analysis or sharing with other users. Forgetting is also useful for information hiding, explanation generation, and ontology debugging and repair.

Because forgetting is an inherently difficult problem — it is much harder than standard reasoning (satisfiability testing) — and very few logics are known to be complete for forgetting (or have the uniform interpolation property),[1] there has been insufficient research on the topic (in particular on the topic of role forgetting), and few forgetting tools are available. Recent work has developed practical methods for computing uniform interpolants for ontologies defined in expressive OWL language dialects [Koopmann and Schmidt, 2013a; 2013b; 2015b]. These methods, which are saturation approaches based on resolution, can eliminate both concept and role symbols and can handle ontologies specified in description logics from $\mathcal{ALC}$ to $\mathcal{ALCH}$ and $\mathcal{SIF}$. The methods have been extended to $\mathcal{SHQ}$ for concept forgetting in [Koopmann and Schmidt, 2014]. While most of this work is focused on TBox and RBox uniform interpolation, practical methods for uniform interpolation for description logics $\mathcal{ALC}$ and $\mathcal{SHI}$ with ABoxes are described in [Koopmann and Schmidt, 2015a; Koopmann, 2015].

An alternative approach that can perform both concept and role forgetting is described, automated and evaluated in [Zhao and Schmidt, 2016]. This approach is a semantic approach which accommodates ontologies expressible in description logics with nominals, role inverse, role inclusions, role conjunction and the universal role. The foundation for this approach is an adaptation of Ackermann's Lemma [Ackermann, 1935], which also provides the foundation for approaches to second-order quantifier elimination [Doherty *et al.*, 1997; Nonnengart and Szałas, 1999; Gabbay *et al.*, 2008] and modal correspondence theory [Szałas, 1993; Conradie *et al.*, 2006; Schmidt, 2012].

In this paper, we follow an Ackermann approach to forgetting and present a practical method for semantic role forgetting in expressive description logics not considered so far. In

[1] [Konev *et al.*, 2013] have shown that the solution of forgetting does not always exist for $\mathcal{ALC}$ and $\mathcal{EL}$, and the existence of a solution for forgetting a concept (role) symbol is undecidable for $\mathcal{ALC}$.

particular, the method accommodates ontologies expressible in the description logic $\mathcal{ALCOQH}$ and the extension with the universal role $\triangledown$. The extended expressivity enriches the target language, making it expressive enough to represent the forgetting solution which otherwise would have been lost. For example, the solution of forgetting the role symbol $\{r\}$ from the ontology $\{A_1 \sqsubseteq {\geq}2r.B_1, A_2 \sqsubseteq {\leq}1r.B_2\}$ is the set $\{A_1 \sqsubseteq {\geq}2\triangledown.B_1, A_1 \sqcap A_2 \sqsubseteq {\geq}1\triangledown.(B_1 \sqcap \neg B_2)\}$, whereas in a description logic without the universal role, the uniform interpolant is $\{\top\}$, which is weaker. Being based on non-trivial generalizations of Ackermann's Lemma, the method is the only approach so far for forgetting role symbols in description logics with qualified number restrictions. The method is goal-oriented and incremental. It always terminates and is sound in the sense that the forgetting solution is equivalent to the original ontology up to the forgotten symbols, possibly with new concept definer symbols. Our method is nearly role forgetting complete for $\mathcal{ALCOQH}(\triangledown)$-ontologies, and we characterize cases where the method is complete. Only problematic are cases where forgetting a role symbol would require the combinations of certain cardinality constraints and role inclusions. Despite the inherent difficulty of forgetting for this level of expressivity, performance results of an evaluation with a prototypical implementation have shown very good success rates on real-world ontologies.

2 $\mathcal{ALCOQH}(\triangledown)$ and Other Basic Notions

Let $\mathsf{N_C}$, $\mathsf{N_R}$ and $\mathsf{N_O}$ be countably infinite and pairwise disjoint sets of *concept symbols*, *role symbols* and *individual symbols* (aka *nominals*), respectively. *Roles* in $\mathcal{ALCOQH}(\triangledown)$ can be any role symbol $r \in \mathsf{N_R}$ or the universal role $\triangledown$. *Concepts* in $\mathcal{ALCOQH}(\triangledown)$ have one of the following forms:

$$a \mid \top \mid A \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid {\geq}mR.C \mid {\leq}nR.C,$$

where $a \in \mathsf{N_O}$, $A \in \mathsf{N_C}$, C and D are any concepts, R is any role, and $m \geq 1$ and $n \geq 0$ are natural numbers. Additional concepts and roles are defined as abbreviations: $\bot = \neg\top$, $\vartriangle = \neg\triangledown$, $\exists R.C = {\geq}1R.C$, $\forall R.C = {\leq}0R.\neg C$, $\neg{\geq}mR.C = {\leq}nR.C$ and $\neg{\leq}nR.C = {\geq}mR.C$ with $n = m-1$. Concepts of the form ${\geq}mR.C$ and ${\leq}nR.C$ are referred to as *qualified number restrictions* (or *number restrictions* for short), which allow one to specify cardinality constraints on roles. We assume w.l.o.g. that concepts and roles are equivalent relative to associativity and commutativity of $\sqcap$ and $\sqcup$, $\top$ and $\triangledown$ are units w.r.t. $\sqcap$, and $\neg$ is an involution.

An $\mathcal{ALCOQH}(\triangledown)$-ontology is mostly assumed to be composed of a TBox, an RBox and an ABox. A *TBox* $\mathcal{T}$ is a finite set of *concept axioms* of the form $C \sqsubseteq D$ (*concept inclusion*), where C and D are concepts. An *RBox* $\mathcal{R}$ is a finite set of *role axioms* of the form $r \sqsubseteq s$ (*role inclusion*), where $r, s \in \mathsf{N_R}$. We define $C \equiv D$ and $r \equiv s$ as abbreviations for the pair of $C \sqsubseteq D$ and $D \sqsubseteq C$ and the pair of $r \sqsubseteq s$ and $s \sqsubseteq r$, respectively. An *ABox* $\mathcal{A}$ is a finite set of *concept assertions* of the form $C(a)$ and *role assertions* of the form $R(a, b)$, where $a, b \in \mathsf{N_O}$, C is a concept, and R is a role. In a description logic with nominals, ABox assertions can be equivalently expressed as TBox axioms, namely, $C(a)$ as $a \sqsubseteq C$ and $R(a, b)$ as $a \sqsubseteq \exists R.b$. Hence, in this paper, we assume w.l.o.g. that an ontology contains only TBox and RBox axioms.

The semantics of $\mathcal{ALCOQH}(\triangledown)$ is defined in terms of an *interpretation* $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, where $\Delta^{\mathcal{I}}$ is a non-empty set (the *domain of the interpretation*), and $\cdot^{\mathcal{I}}$ is the *interpretation function*, which assigns to every nominal $a \in \mathsf{N_O}$ a singleton $a^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, to every concept symbol $A \in \mathsf{N_C}$ a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and to every role symbol $r \in \mathsf{N_R}$ a relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The interpretation function $\cdot^{\mathcal{I}}$ is inductively extended to concepts and roles as follows:

$$\top^{\mathcal{I}} = \Delta^{\mathcal{I}} \qquad \triangledown^{\mathcal{I}} = \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \qquad (\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \backslash C^{\mathcal{I}}$$

$$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}} \qquad (C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$$

$$({\geq}mR.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \#\{(x, y) \in R^{\mathcal{I}} \mid y \in C^{\mathcal{I}}\} \geq m\}$$

$$({\leq}nR.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \#\{(x, y) \in R^{\mathcal{I}} \mid y \in C^{\mathcal{I}}\} \leq n\},$$

where $\#\{\cdot\}$ denotes the cardinality of the set $\{\cdot\}$.

A concept axiom $C \sqsubseteq D$ is *true* in an interpretation $\mathcal{I}$, and we write $\mathcal{I} \models C \sqsubseteq D$, iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. A role axiom $r \sqsubseteq s$ is *true* in an interpretation $\mathcal{I}$, and we write $\mathcal{I} \models r \sqsubseteq s$, iff $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$. $\mathcal{I}$ is a *model* of an ontology $\mathcal{O}$ iff every axiom in $\mathcal{O}$ is *true* in $\mathcal{I}$. In this case we write $\mathcal{I} \models \mathcal{O}$.

Our method works with TBox and RBox axioms in clausal normal form. We assume w.l.o.g. that a *TBox literal* is a concept of the form a, $\neg a$, A, $\neg A$, ${\geq}mR.C$ or ${\leq}nR.C$, where $a \in \mathsf{N_O}$, $A \in \mathsf{N_C}$, $m \geq 1$ and $n \geq 0$ are natural numbers, C is any concept, and R is any role. A *TBox clause* is a disjunction of a finite number of TBox literals. An *RBox clause* is a disjunction of a role symbol and a negated role symbol. TBox and RBox clauses are obtained by clausification of TBox and RBox axioms, where in the latter case role negation is introduced. This is done for consistency in presentation, replacing role inclusion by disjunction as the main operator. Nominals are treated as regular concept symbols in our method, because we are only concerned with role forgetting in this paper.

An axiom (clause) that contains a designated (concept or role) symbol $\mathcal{S}$ is called an *$\mathcal{S}$-axiom* (*$\mathcal{S}$-clause*). An occurrence of $\mathcal{S}$ is assumed to be *positive* (*negative*) in an $\mathcal{S}$-axiom ($\mathcal{S}$-clause) if it is under an *even* (*odd*) number of explicit and implicit negations. For instance, r is assumed to be positive in ${\geq}mr.A$ and $s \sqsubseteq r$, and negative in ${\leq}nr.A$ and $r \sqsubseteq s$. A set $\mathcal{N}$ of axioms (clauses) is assumed to be *positive* (*negative*) w.r.t. $\mathcal{S}$ if every occurrence of $\mathcal{S}$ in $\mathcal{N}$ is positive (negative).

3 Definition of Forgetting, Ackermann's Lemma and Obstacles to Role Forgetting

Forgetting can be defined in two ways that are closely related: one is analogous to *model inseparability* (i.e., a semantic notion based on model-conservative extensions; see e.g. [Konev *et al.*, 2013]), which preserves *equivalence* up to certain signatures, and the other is via *uniform interpolation* (i.e., a syntactic notion based on deductive-conservative extensions; see e.g. [Visser, 1996]), which preserves *logical consequences*; see [Botoeva *et al.*, 2016] a survey for their interrelation.

Our notion of forgetting is a semantic notion. By $\mathrm{sig}_{\mathsf{C}}(X)$ and $\mathrm{sig}_{\mathsf{R}}(X)$ we denote the sets of respectively the concept and role symbols occurring in X (*excluding nominals*), where X ranges over axioms, clauses, sets of axioms, and sets of clauses. Let $r \in \mathsf{N_R}$ be any role symbol, and let $\mathcal{I}$ and $\mathcal{I}'$ be any interpretations. We say $\mathcal{I}$ and $\mathcal{I}'$ are *equivalent up to* r, or r-*equivalent*, if $\mathcal{I}$ and $\mathcal{I}'$ coincide but differ possibly in the interpretations of r. More generally, $\mathcal{I}$ and $\mathcal{I}'$ are *equivalent up to a set* Σ *of role symbols*, or Σ-*equivalent*, if $\mathcal{I}$ and $\mathcal{I}'$ coincide but differ possibly in the interpretations of the symbols in Σ. This can be understood as follows: (i) $\mathcal{I}$ and $\mathcal{I}'$ have the same domain, i.e., $\Delta^{\mathcal{I}} = \Delta^{\mathcal{I}'}$, and interpret every concept symbol and every individual symbol identically, i.e., $A^{\mathcal{I}} = A^{\mathcal{I}'}$ for every $A \in \mathsf{N_C}$ and $a^{\mathcal{I}} = a^{\mathcal{I}'}$ for every $a \in \mathsf{N_O}$; (ii) for every role symbol $r \in \mathsf{N_R}$ not in Σ, $r^{\mathcal{I}} = r^{\mathcal{I}'}$.

Definition 1 (Role Forgetting for $\mathcal{ALCOQH}(\triangledown)$**)** *Let $\mathcal{O}$ be an $\mathcal{ALCOQH}(\triangledown)$-ontology and let Σ be a subset of $\mathrm{sig}_R(\mathcal{O})$. An ontology $\mathcal{O}'$ is a* solution of forgetting Σ from $\mathcal{O}$*, iff the following conditions hold: (i) $\mathrm{sig}_R(\mathcal{O}') \subseteq \mathrm{sig}_R(\mathcal{O})\backslash\Sigma$, and (ii) for any interpretation $\mathcal{I}$: $\mathcal{I} \models \mathcal{O}'$ iff $\mathcal{I}' \models \mathcal{O}$, for some interpretation $\mathcal{I}'$ Σ-equivalent to $\mathcal{I}$.*

It follows from this that: (i) the original ontology $\mathcal{O}$ and the forgetting solution $\mathcal{O}'$ are equivalent up to (the interpretations of) the symbols in Σ. Also (ii) forgetting solutions are unique up to equivalence, that is, if both $\mathcal{O}'$ and $\mathcal{O}''$ are solutions of forgetting Σ from $\mathcal{O}$, then they are logically equivalent.

In this paper, Σ is always assumed to be a set of symbols to be forgotten. The symbol in Σ under current consideration for forgetting is referred to as the *pivot* in our method. An axiom (clause) that contains an occurrence of the pivot is referred to as a *pivot-axiom* (*pivot-clause*).

Given an ontology $\mathcal{O}$ and a set Σ of concept and role symbols, computing a solution of forgetting Σ from $\mathcal{O}$ can be reduced to the problem of eliminating single symbols in Σ. This can be based on the use of a monotonicity property found in [Ackermann, 1935], referred to as *Ackermann's Lemma*. For ontologies, Ackermann's Lemma can be formulated as the following theorem. The proof is an easy adaptation of Ackermann's original result [Gabbay *et al.*, 2008].

Theorem 1 (Ackermann's Lemma) *Let $\mathcal{O}$ be an ontology that contains axioms $\alpha_1 \sqsubseteq \mathcal{S}, ..., \alpha_n \sqsubseteq \mathcal{S}$, where $\mathcal{S} \in N_C$ (or $\mathcal{S} \in N_R$), and the α_i ($1 \leq i \leq n$) are concepts (or roles) that do not contain $\mathcal{S}$. If $\mathcal{O}\backslash\{\alpha_1 \sqsubseteq \mathcal{S}, ..., \alpha_n \sqsubseteq \mathcal{S}\}$ is negative w.r.t. $\mathcal{S}$, then $\mathcal{O}^{\mathcal{S}}_{\alpha_1\sqcup...\sqcup\alpha_n}$ is a* solution of forgetting $\{\mathcal{S}\}$ from $\mathcal{O}$ *(i.e., Conditions (i) and (ii) of Definition 1 hold), where $\mathcal{O}^{\mathcal{S}}_{\alpha_1\sqcup...\sqcup\alpha_n}$ denotes the ontology obtained from $\mathcal{O}$ by substituting $\alpha_1 \sqcup ... \sqcup \alpha_n$ for every occurrence of $\mathcal{S}$ in $\mathcal{O}$.*

In Ackermann-based approaches, e.g., [Szałas, 2006; Conradie *et al.*, 2006; Schmidt, 2012; Zhao and Schmidt, 2015; 2016], the lemma is often used as the following rule:

$$\frac{\mathcal{O}\backslash\{\alpha_1 \sqsubseteq \mathcal{S}, ..., \alpha_n \sqsubseteq \mathcal{S}\}, \alpha_1 \sqsubseteq \mathcal{S}, ..., \alpha_n \sqsubseteq \mathcal{S}}{\mathcal{O}^{\mathcal{S}}_{\alpha_1\sqcup...\sqcup\alpha_n}} \qquad (1)$$

The idea of Ackermann's Lemma is based on a notion of '*substitution*', which can informally yet intuitively be understood as follows: given an ontology $\mathcal{O}$ with $\mathcal{S} \in \mathrm{sig}_C(\mathcal{O})$ (or $\mathcal{S} \in \mathrm{sig}_R(\mathcal{O})$) being the pivot, if there exists a concept (or a role) α such that $\mathcal{S} \notin \mathrm{sig}(\alpha)$ and α *defines* $\mathcal{S}$ w.r.t. $\mathcal{O}$, then we can substitute this *definition* for every occurrence of $\mathcal{S}$ in $\mathcal{O}$ ($\mathcal{S}$ is thus eliminated from $\mathcal{O}$). This lemma also holds, when the inclusions are reversed, i.e., $\mathcal{S} \sqsubseteq \alpha_1, ..., \mathcal{S} \sqsubseteq \alpha_n$, and the polarity of $\mathcal{S}$ in the rest part of $\mathcal{O}$ is switched, i.e., $\mathcal{O}\backslash\{\mathcal{S} \sqsubseteq \alpha_1, ..., \mathcal{S} \sqsubseteq \alpha_n\}$ is positive w.r.t. $\mathcal{S}$.

A crucial task in Ackermann-based approaches, therefore, is to find a *definition* of the pivot w.r.t. the present ontology, that is, to reformulate all pivot-axioms with positive occurrences of the pivot being in the form $\alpha \sqsubseteq \mathcal{S}$ (or dually, with negative occurrences of the pivot being in the form $\mathcal{S} \sqsubseteq \alpha$), where $\mathcal{S} \notin \mathrm{sig}(\alpha)$. In the context of this paper where axioms are represented in clausal form, this means reformulating all pivot-clauses with positive occurrences of the pivot being in the form $\neg\alpha \sqcup \mathcal{S}$ (or dually, with negative occurrences of the pivot being in the form $\neg\mathcal{S} \sqcup \alpha$), where $\mathcal{S} \notin \mathrm{sig}(\alpha)$.

In the case of concept forgetting, a concept symbol (or a negated concept symbol) deep inside a clause could be moved outward by using *Galois connections* between $\forall r$ and $\forall r^-$ (e.g., a TBox clause $\neg A \sqcup \forall r.\mathcal{S}$ can be equivalently rewritten as $(\forall r^-.\neg A) \sqcup \mathcal{S}$, where r^- denotes the inverse of r), or by exploiting the idea of *Skolemization* (e.g., an ABox clause $\neg a \sqcup \exists r.\neg\mathcal{S}$ can be equivalently rewritten as $\neg a \sqcup \exists r.b$ and $\neg b \sqcup \neg\mathcal{S}$, where b is a fresh nominal). This is explained in detail in the work of [Conradie *et al.*, 2006; Schmidt, 2012; Zhao and Schmidt, 2015; 2016].

In the case of role forgetting, since every role symbol that occurs in a TBox clause is always preceded by a role restriction operator (i.e., a number restriction in this work), it is not obvious how to reformulate the TBox pivot-clauses. Thus a direct approach based on Ackermann's Lemma does not seem feasible for role forgetting in ontologies with TBoxes.

How then to do role forgetting? For the translation of ontologies in first-order logic, there are no such obstacles. We could apply Ackermann's Lemma for first-order logic (e.g., as in the DLS algorithm of [Doherty *et al.*, 1997]) to eliminate a single role symbol. Such an indirect approach requires suitable back-translation however, which is absent at present for expressive description logics. Translating first-order formulas into equivalent description logic expressions is not straightforward, in particular when number restrictions are present in the target language. For example, the solution of forgetting the role symbol $\{r\}$ from $\{A_1 \sqcup {\geq}2r.B_1, A_2 \sqcup {\leq}1r.B_2\}$ in quantifier-free first-order logic is the set:

$$\begin{aligned}&\{\forall x(A_1(x) \vee B_1(f_1(x))), \forall x(A_1(x) \vee B_1(f_2(x))),\\ &\forall x(A_1(x) \vee f_1(x) \not\approx f_2(x)),\\ &\forall x(A_1(x) \vee A_2(x) \vee \neg B_2(f_1(x)) \vee \neg B_2(f_2(x)))\},\end{aligned}$$

where $f_1(x)$ and $f_2(x)$ are Skolem terms, and $f_1(x) \not\approx f_2(x)$ is an inequality. Because of the presence of the Skolem terms and the inequality, it is not clear whether this solution can be expressed equivalently in a description logic.

4 Our Ackermann-Based Approach to Eliminating A Single Role Symbol

In this section we present our approach to eliminating a single role symbol from a set of TBox and RBox clauses expressible in $\mathcal{ALCOQH}(\triangledown)$. It is a direct approach based on non-trivial generalizations of Ackermann's Lemma. The approach has two key ingredients: (i) transformation of the present clause set into *reduced form*, and (ii) a set of *Ackermann rules*. The Ackermann rules reflect the generalizations of Ackermann's Lemma and allow a single role symbol to be eliminated from a set of clauses in reduced form.

Ackermann I

$$\frac{\mathcal{N}, \overbrace{C_1 \sqcup {\geq}x_1 r.D_1, \ldots, C_m \sqcup {\geq}x_m r.D_m}^{\mathcal{P}^+_{\mathcal{T}}(r)}, \overbrace{\neg s_1 \sqcup r, \ldots, \neg s_v \sqcup r}^{\mathcal{P}^+_{\mathcal{R}}(r)}}{\mathcal{N}, \textsc{Block}(\mathcal{P}^+_{\mathcal{T}}(r), \emptyset), \textsc{Block}(\mathcal{P}^+_{\mathcal{R}}(r), \emptyset)}$$

Ackermann II

$$\frac{\mathcal{N}, \overbrace{E_1 \sqcup {\leq}y_1 r.F_1, \ldots, E_n \sqcup {\leq}y_n r.F_n}^{\mathcal{P}^-_{\mathcal{T}}(r)}, \overbrace{t_1 \sqcup \neg r, \ldots, t_w \sqcup \neg r}^{\mathcal{P}^-_{\mathcal{R}}(r)}}{\mathcal{N}, \textsc{Block}(\mathcal{P}^-_{\mathcal{T}}(r), \emptyset), \textsc{Block}(\mathcal{P}^-_{\mathcal{R}}(r), \emptyset)}$$

Ackermann III

$$\frac{\mathcal{N}, \overbrace{C_1 \sqcup {\geq}x_1 r.D_1, \ldots, C_m \sqcup {\geq}x_m r.D_m}^{\mathcal{P}^+_{\mathcal{T}}(r)}, \overbrace{E_1 \sqcup {\leq}y_1 r.F_1, \ldots, E_n \sqcup {\leq}y_n r.F_n}^{\mathcal{P}^-_{\mathcal{T}}(r)}, \overbrace{t_1 \sqcup \neg r, \ldots, t_w \sqcup \neg r}^{\mathcal{P}^-_{\mathcal{R}}(r)}}{\mathcal{N}, \textsc{Block}(\mathcal{P}^+_{\mathcal{T}}(r), E_1 \sqcup {\leq}y_1 r.F_1), ..., \textsc{Block}(\mathcal{P}^+_{\mathcal{T}}(r), E_n \sqcup {\leq}y_n r.F_n), \textsc{Block}(\mathcal{P}^+_{\mathcal{T}}(r), t_1 \sqcup \neg r), ..., \textsc{Block}(\mathcal{P}^+_{\mathcal{T}}(r), t_w \sqcup \neg r)}$$

Ackermann IV

$$\frac{\mathcal{N}, \overbrace{\neg s_1 \sqcup r, \ldots, \neg s_v \sqcup r}^{\mathcal{P}^+_{\mathcal{R}}(r)}, \overbrace{E_1 \sqcup {\leq}y_1 r.F_1, \ldots, E_n \sqcup {\leq}y_n r.F_n}^{\mathcal{P}^-_{\mathcal{T}}(r)}, \overbrace{t_1 \sqcup \neg r, \ldots, t_w \sqcup \neg r}^{\mathcal{P}^-_{\mathcal{R}}(r)}}{\mathcal{N}, \textsc{Block}(\mathcal{P}^+_{\mathcal{R}}(r), E_1 \sqcup {\leq}y_1 r.F_1), ..., \textsc{Block}(\mathcal{P}^+_{\mathcal{R}}(r), E_n \sqcup {\leq}y_n r.F_n), \textsc{Block}(\mathcal{P}^+_{\mathcal{R}}(r), t_1 \sqcup \neg r), ..., \textsc{Block}(\mathcal{P}^+_{\mathcal{R}}(r), t_w \sqcup \neg r)}$$

Ackermann V

$$\frac{\mathcal{N}, \overbrace{C_1 \sqcup {\geq}x_1 r.D_1, \ldots, C_m \sqcup {\geq}x_m r.D_m}^{\mathcal{P}^+_{\mathcal{T}}(r)}, \overbrace{\neg s_1 \sqcup r, \ldots, \neg s_v \sqcup r}^{\mathcal{P}^+_{\mathcal{R}}(r)}, \overbrace{E_1 \sqcup {\leq}0 r.F_1, \ldots, E_n \sqcup {\leq}0 r.F_n}^{\mathcal{P}^{-,0}_{\mathcal{T}}(r)}, \overbrace{t_1 \sqcup \neg r, \ldots, t_w \sqcup \neg r}^{\mathcal{P}^-_{\mathcal{R}}(r)}}{\begin{array}{c}\mathcal{N}, \textsc{Block}(\mathcal{P}^{-,0}_{\mathcal{T}}(r), C_1 \sqcup {\geq}x_1 r.D_1), ..., \textsc{Block}(\mathcal{P}^{-,0}_{\mathcal{T}}(r), C_m \sqcup {\geq}x_m r.D_m), \\ \textsc{Block}(\mathcal{P}^{-,0}_{\mathcal{T}}(r), \neg s_1 \sqcup r), ..., \textsc{Block}(\mathcal{P}^{-,0}_{\mathcal{T}}(r)), \neg s_v \sqcup r), \\ \textsc{Block}(\mathcal{P}^-_{\mathcal{R}}(r), C_1 \sqcup {\geq}x_1 r.D_1), ..., \textsc{Block}(\mathcal{P}^-_{\mathcal{R}}(r)), C_m \sqcup {\geq}x_m r.D_m), \\ \textsc{Block}(\mathcal{P}^-_{\mathcal{R}}(r), \neg s_1 \sqcup r), ..., \textsc{Block}(\mathcal{P}^-_{\mathcal{R}}(r)), \neg s_v \sqcup r)\end{array}}$$

Notation in Ackermann rules ($1 \leq i \leq m$, $1 \leq j \leq n$, $1 \leq k \leq v$, $1 \leq l \leq w$):

1. $\textsc{Block}(\mathcal{P}^+_{\mathcal{T}}(r), \emptyset)$ denotes the set $\{C_1 \sqcup {\geq}x_1 \nabla.D_1, \ldots, C_m \sqcup {\geq}x_m \nabla.D_m\}$. **2.** $\textsc{Block}(\mathcal{P}^+_{\mathcal{R}}(r), \emptyset)$ denotes $\emptyset$.

3. $\textsc{Block}(\mathcal{P}^-_{\mathcal{T}}(r), \emptyset)$ denotes the set $\{E_1 \sqcup {\leq}y_1 \neg\nabla.F_1, \ldots, E_n \sqcup {\leq}y_n \neg\nabla.F_n\}$. **4.** $\textsc{Block}(\mathcal{P}^-_{\mathcal{R}}(r), \emptyset)$ denotes $\emptyset$.

5. $\textsc{Block}(\mathcal{P}^+_{\mathcal{T}}(r), E_j \sqcup {\leq}y_j r.F_j)$ denotes the union of following sets, where $m = |\mathcal{P}^+_{\mathcal{T}}(r)|$:

Ground Block: $\{C_1 \sqcup {\geq}x_1 \nabla.D_1, \ldots, C_m \sqcup {\geq}x_m \nabla.D_m\}$

1st-tier Block: $\bigcup_{1 \leq i \leq m} \{E_j \sqcup C_i \sqcup {\geq}(x_i - y_j)\nabla.(D_i \sqcap \neg F_j)\}$ for any i such that $x_i > y_j$

2nd-tier Block: $\bigcup_{1 \leq i_1 < i_2 \leq m} \{E_j \sqcup C_{i_1} \sqcup C_{i_2} \sqcup {\geq}x\nabla.(D_{i_1} \sqcap D_{i_2}) \sqcup {\geq}(x_{i_1} + x_{i_2} - y_j - (x-1))\nabla.((D_{i_1} \sqcup D_{i_2}) \sqcap \neg F_j) \sqcup {\geq}1\nabla.((D_{i_1} \sqcup D_{i_2}) \sqcap \neg F_j) \mid x \in \{1, \ldots, x_{\mathsf{min}}\}\}$ for any i_1 and i_2 such that $x_{i_1} + x_{i_2} > y_j$, where x_{min} denotes the minimum of x_{i_1}, x_{i_2} and $x_{i_1} + x_{i_2} - y_j$.

. . .

mth-tier Block: $\{E_j \sqcup C_1 \sqcup \ldots \sqcup C_m \sqcup {\geq}x\nabla.(D_1 \sqcap \ldots \sqcap D_m) \sqcup {\geq}(x_1 + \ldots + x_m - y_j - (x-1))\nabla.((D_1 \sqcup \ldots \sqcup D_m) \sqcap \neg F_j) \sqcup \ldots \sqcup {\geq}1\nabla.((D_1 \sqcup \ldots \sqcup D_m) \sqcap \neg F_j) \mid x \in \{1, \ldots, x_{\mathsf{min}}\}\}$ if $x_1 + \ldots + x_m \geq y_j$, where x_{min} denotes the minimum of $x_1, \ldots, x_m$ and $x_1 + ... + x_m - y_j$.

6. $\textsc{Block}(\mathcal{P}^+_{\mathcal{T}}(r), t_l \sqcup \neg r)$ denotes the set: $\{C_1 \sqcup {\geq}x_1 t_l.D_1, \ldots, C_m \sqcup {\geq}x_m t_l.D_m\}$.

7. $\textsc{Block}(\mathcal{P}^+_{\mathcal{R}}(r), E_j \sqcup {\leq}y_j r.F_j)$ denotes the set: $\{E_j \sqcup {\leq}y_j s_1.F_j, \ldots, E_j \sqcup {\leq}y_j s_v.F_j\}$.

8. $\textsc{Block}(\mathcal{P}^+_{\mathcal{R}}(r), t_l \sqcup \neg r)$ denotes the set: $\{\neg s_1 \sqcup t_l, \ldots, \neg s_v \sqcup t_l\}$.

9. $\textsc{Block}(\mathcal{P}^{-,0}_{\mathcal{T}}(r), C_i \sqcup {\geq}x_i r.D_i)$ denotes the union of following sets, where $n = |\mathcal{P}^{-,0}_{\mathcal{T}}(r)|$:

Ground Block: $\{C_i \sqcup {\geq}x_i \nabla.D_i\}$

1st-tier Block: $\bigcup_{1 \leq j \leq n} \{C_i \sqcup E_j \sqcup {\geq}x_i \nabla.(D_i \sqcap \neg F_j)\}$

2nd-tier Block: $\bigcup_{1 \leq j_1 < j_2 \leq n} \{C_i \sqcup E_{j_1} \sqcup E_{j_2} \sqcup {\geq}x_i \nabla.(D_i \sqcap \neg F_{j_1} \sqcap \neg F_{j_2})\}$

. . .

nth-tier Block: $\{C_i \sqcup E_1 \sqcup \ldots \sqcup E_n \sqcup {\geq}x_i \nabla.(D_i \sqcap \neg F_1 \sqcap \ldots \sqcap \neg F_n)\}$

10. $\textsc{Block}(\mathcal{P}^{-,0}_{\mathcal{T}}(r), \neg s_k \sqcup r)$ denotes the set: $\{E_1 \sqcup {\leq}0 s_k.F_1, \ldots, E_n \sqcup {\leq}0 s_k.F_n\}$.

11. $\textsc{Block}(\mathcal{P}^-_{\mathcal{R}}(r), C_i \sqcup {\geq}x_i r.D_i)$ denotes the set: $\{C_i \sqcup {\geq}x_i t_1.D_i, \ldots, C_i \sqcup {\geq}x_i t_w.D_i\}$.

12. $\textsc{Block}(\mathcal{P}^-_{\mathcal{R}}(r), \neg s_k \sqcup r)$ denotes the set: $\{t_1 \sqcup \neg s_k, \ldots, t_w \sqcup \neg s_k\}$.

Figure 1: Ackermann rules for eliminating $r \in \mathsf{N_R}$ from a set of clauses in reduced form. In the rules we assume that $r \notin \mathrm{sig}_{\mathsf{R}}(\mathcal{N})$.

4.1 Transformation into Reduced Form

Definition 2 (Reduced Form) *For $r \in \mathsf{N_R}$ the pivot, a TBox pivot-clause is in* reduced form *if it has the form $E \sqcup {\geq}mr.F$ or $E \sqcup {\leq}nr.F$, where E and F are concepts that do not contain r, and $m \geq 1$ and $n \geq 0$ are natural numbers. An RBox pivot-clause is in* reduced form *if it has the form $\neg\mathcal{S} \sqcup r$ or $\mathcal{S} \sqcup \neg r$, where $\mathcal{S} \in \mathsf{N_R}$ and $\mathcal{S} \neq r$. A set $\mathcal{N}$ of clauses is in* reduced form *if all pivot-clauses in $\mathcal{N}$ are in reduced form.*

The reduced forms incorporate all basic forms of TBox and RBox clauses in which a role symbol could occur. While an RBox pivot-clause is always in reduced form, this is not true for a TBox pivot-clause. A TBox pivot-clause not in reduced form has the form $E \sqcup {\geq}m\mathcal{S}.F$ or $E \sqcup {\leq}n\mathcal{S}.F$, where $\mathcal{S}$ can be any role (including the pivot role symbol), and E and F are concepts with at least one of them containing the pivot; $({\leq}1r.A) \sqcup ({\leq}3s.{\geq}2r.B)$ is such an example. Transforming a TBox pivot-clause into reduced form is not always possible unless definer symbols are introduced. *Definer symbols* are auxiliary concept symbols that do not occur in the present ontology [Koopmann and Schmidt, 2013b], and are introduced as follows: let $\mathcal{N}$ be a set of clauses not in reduced form, and let $\mathsf{N_D} \subset \mathsf{N_C}$ be a set of definer symbols disjoint from $\mathrm{sig}_\mathsf{C}(\mathcal{N})$. Definer symbols are used as substitutes, incrementally replacing 'E' and 'F' for every TBox pivot-clause not in reduced form until neither 'E' nor 'F' contain the pivot. A new clause $\neg\mathsf{D}_1 \sqcup E$ is added to $\mathcal{N}$ for each replaced subconcept E, a new clause $\neg\mathsf{D}_2 \sqcup F$ is added to $\mathcal{N}$ for each replaced subconcept F immediately under a $\geq$-restriction, and a new clause $\mathsf{D}_3 \sqcup F$ is added to $\mathcal{N}$ for each replaced subconcept F immediately under a $\leq$-restriction, where $\mathsf{D}_1, \mathsf{D}_2, \mathsf{D}_3 \in \mathsf{N_D}$ are fresh definer symbols. $\mathcal{N}$ is thus transformed into a set of clauses in reduced form. For the example mentioned above, this means that D_1 and D_2 are introduced to replace ${\leq}1r.A$ and ${\geq}2r.B$, respectively, which yields $\mathsf{D}_1 \sqcup {\leq}3s.\mathsf{D}_2$ and two additional clauses $\neg\mathsf{D}_1 \sqcup {\leq}1r.A$ and $\mathsf{D}_2 \sqcup {\geq}2r.B$. The original clause is thus represented by these three clauses in reduced form (to which an Ackermann rule can be applied).

Theorem 2 *Using definer introduction as described above, any $\mathcal{ALCOQH}(\triangledown)$-ontology can be transformed into a set of clauses in reduced form. The transformation preserves equivalence up to the introduced definer symbols.*

4.2 Ackermann Rules

Let $\mathcal{N}$ be a set of TBox and RBox clauses exhibiting all different reduced forms, for $r \in \mathrm{sig}_\mathsf{R}(\mathcal{N})$ the pivot. We refer to the clauses of the form $C \sqcup {\geq}mr.D$ and the form $C \sqcup {\leq}nr.D$ as *positive TBox premises* and *negative TBox premises* of the Ackermann rules, respectively. We refer to the clauses of the form $\neg\mathcal{S} \sqcup r$ and the form $\mathcal{S} \sqcup \neg r$ as *positive RBox premises* and *negative RBox premises* of the Ackermann rules, respectively. By $\mathcal{P}^+_\mathcal{T}(r)$ and $\mathcal{P}^-_\mathcal{T}(r)$ we denote respectively the sets of positive TBox premises and negative TBox premises. By $\mathcal{P}^+_\mathcal{R}(r)$ and $\mathcal{P}^-_\mathcal{R}(r)$ we denote respectively the sets of positive RBox premises and negative RBox premises. By $\mathcal{P}^+(r)$ and $\mathcal{P}^-(r)$ we denote respectively the union of $\mathcal{P}^+_\mathcal{T}(r)$ and $\mathcal{P}^+_\mathcal{R}(r)$, and the union of $\mathcal{P}^-_\mathcal{T}(r)$ and $\mathcal{P}^-_\mathcal{R}(r)$.

The Ackermann rules, shown in Figure 1, are based on an idea of '*combination*'. Specifically, the idea is to combine all positive premises $\mathcal{P}^+(r)$ with every negative premise $\alpha(r)$ in $\mathcal{P}^-(r)$ (or dually, to combine all negative premises $\mathcal{P}^-(r)$ with every positive premise $\alpha(r)$ in $\mathcal{P}^+(r)$). The result is a finite set of clauses, denoted by $\textbf{B}\textsc{lock}(\mathcal{P}^{+(-)}(r), \alpha(r))$. It is observed that the result obtained from combining $\mathcal{P}^+(r)$ with a negative premise is always identical to the union of the results obtained from combining respectively $\mathcal{P}^+_\mathcal{T}(r)$ and $\mathcal{P}^+_\mathcal{R}(r)$ with that premise (and the dual also holds). We therefore treat every combination of $\mathcal{P}^+(r)$ with a negative premise as two separate combinations in our Ackermann rules (same for the dual), so that it can be understood better from which premises a resulting $\textbf{B}\textsc{lock}$ of clauses is obtained.

For different $\mathcal{P}^+_\mathcal{T}(r)$, $\mathcal{P}^+_\mathcal{R}(r)$, $\mathcal{P}^-_\mathcal{T}(r)$, $\mathcal{P}^-_\mathcal{R}(r)$, and $\alpha(r)$, the combination is performed as 12 distinct cases (see Figure 1). For most cases, the idea is analogous to that of Ackermann's Lemma (and its dual), where the pivot is eliminated by substituting its *definition* found w.r.t. the present premises for every occurrence of the pivot in these premises. Only for Cases 5 and 9, the combination has a different flavor; their idea is illustrated with two concrete examples.

Case 5: Combining $\mathcal{P}^+_\mathcal{T}(r)$ with a negative TBox premise in $\mathcal{P}^-_\mathcal{T}(r)$, e.g., $E_j \sqcup {\leq}y_jr.F_j$ $(1 \leq j \leq n)$, yields a set of TBox clauses, denoted by $\textbf{B}\textsc{lock}(\mathcal{P}^+_\mathcal{T}(r), E_j \sqcup {\leq}y_jr.F_j)$.

Example 1 Combining $\mathcal{P}^+_\mathcal{T}(r) = \{A_1 \sqcup {\geq}2r.B_1, A_2 \sqcup {\geq}1r.B_2\}$ with $\{A \sqcup {\leq}1r.B\}$ yields a set $\textbf{B}\textsc{lock}(\mathcal{P}^+_\mathcal{T}(r), A \sqcup {\leq}1r.B)$ that contains the following subsets of clauses:

Ground Block: $\{A_1 \sqcup {\geq}2\triangledown.B_1, A_2 \sqcup {\geq}1\triangledown.B_2\}$

1st-tier Block: $\{A \sqcup A_1 \sqcup {\geq}1\triangledown.(B_1 \sqcap \neg B)\}$

2nd-tier Block: $\{A \sqcup A_1 \sqcup A_2 \sqcup {\geq}1\triangledown.(B_1 \sqcap B_2) \sqcup {\geq}2\triangledown.((B_1 \sqcup B_2) \sqcap \neg B) \sqcup {\geq}1\triangledown.((B_1 \sqcup B_2) \sqcap \neg B)$

Case 9: Combining $\mathcal{P}^{-,0}_\mathcal{T}(r)$ with a positive TBox premise in $\mathcal{P}^+_\mathcal{T}(r)$, e.g., $C_i \sqcup {\geq}x_ir.D_i$ $(1 \leq i \leq m)$, yields a set of TBox clauses, denoted by $\textbf{B}\textsc{lock}(\mathcal{P}^{-,0}_\mathcal{T}(r), C_i \sqcup {\geq}x_ir.D_i)$. In this case, $\mathcal{P}^{-,0}_\mathcal{T}(r)$ denotes the set of negative TBox premises of the form $E \sqcup {\leq}0r.F$ (i.e., cardinality constraints are 0).

Example 2 Combining $\mathcal{P}^{-,0}_\mathcal{T}(r) = \{A_1 \sqcup {\leq}0r.B_1, A_2 \sqcup {\leq}0r.B_2\}$ with $\{A \sqcup {\geq}2r.B\}$ yields a set $\textbf{B}\textsc{lock}(\mathcal{P}^{-,0}_\mathcal{T}(r), A \sqcup {\geq}2r.B)$ that contains the following subsets of clauses:

Ground Block: $\{A \sqcup {\geq}2\triangledown.B\}$

1st-tier Block: $\{A \sqcup A_1 \sqcup {\geq}2\triangledown.(B \sqcap \neg B_1)\}$

2nd-tier Block: $\{A \sqcup A_1 \sqcup A_2 \sqcup {\geq}2\triangledown.(B \sqcap \neg B_1 \sqcap \neg B_2)\}$

How are the Ackermann rules used? For a set $\mathcal{N}$ of clauses in reduced form, depending on which kinds of premises the set $\mathcal{N}$ contains, we apply different Ackermann rules (to the premises in $\mathcal{N}$ to eliminate the pivot). Specifically, if $\mathcal{N}$ contains only positive premises, we apply the Ackermann I rule. If $\mathcal{N}$ contains only negative premises, we apply the Ackermann II rule. If $\mathcal{N}$ contains only positive TBox premises, as well as negative premises, we apply the Ackermann III rule. If $\mathcal{N}$ contains only positive RBox premises, as well as negative premises, we apply the Ackermann IV rule. If $\mathcal{N}$ contains both positive TBox and RBox premises, as well as negative premises, we apply the Ackermann V rule. Note that there is a gap in the scope of the rules in the Ackermann V rule; it is applicable only to the cases where all negative TBox premises (if they are present in $\mathcal{N}$) are of the form $E \sqcup {\leq}0r.F$.

Theorem 3 *Let $\mathcal{I}$ be any $\mathcal{ALCOQH}(\triangledown)$-interpretation. For $r \in \mathsf{N_R}$ the pivot, when an Ackermann rule is applicable, the* *conclusion of the rule is true in $\mathcal{I}$ iff for some interpretation $\mathcal{I}'$ r-equivalent to $\mathcal{I}$, the premises are true in $\mathcal{I}'$.*

This implies that the *conclusion* of an Ackermann rule is a solution of forgetting the pivot from the *premises* of the rule.

5 Description of the Forgetting Method

Given an ontology $\mathcal{O}$ of axioms and a set Σ of role symbols to be forgotten, the forgetting process in our method comprises three main phases (see Figure 2): the conversion of $\mathcal{O}$ into a set $\mathcal{N}$ of clauses (**the first phase**), the Σ-symbol elimination phase (**the central phase**), and the definer elimination phase (**the final phase**). It is assumed that as soon as a forgetting solution is computed, the remaining phases are skipped.

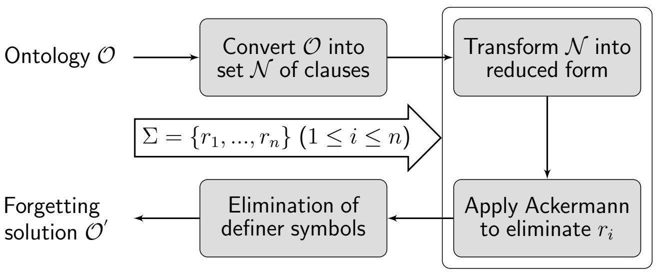

Figure 2: Main phases in the forgetting process

Input: Given as input to the method are an $\mathcal{ALCOQH}(\triangledown)$-ontology $\mathcal{O}$ of axioms and a set $\Sigma \subseteq \mathrm{sig}_{\mathsf{R}}(\mathcal{O})$ of role symbols to be forgotten. An important feature of the method is that Σ-symbols can be flexibly specified.

The first phase: The first phase of the forgetting process internalizes all ABox assertions in $\mathcal{O}$ (if they are present) into TBox axioms, and then transforms $\mathcal{O}$ into a set $\mathcal{N}$ of clauses using standard clausal form transformations.

The central phase: Central to the forgetting process is the Σ-symbol elimination phase, which is an iteration of several rounds in which the elimination of Σ-symbols is attempted. Specifically, the method attempts to eliminate the Σ-symbols one by one using the approach as described in the previous section. In each elimination round, the method performs two steps. The first step transforms every TBox pivot-clause (not in reduced form) into reduced form, so that one of the Ackermann rules can be applied. The second step then applies the Ackermann rule to the pivot-clauses to eliminate the pivot. Upon the intermediate result being returned at the end of each round, the method repeats the same steps in the next round for the elimination of the remaining symbols in Σ (if necessary). If a Σ-symbol has been found ineliminable from the present ontology (i.e., none of the Ackermann rules is applicable to the current reduced form), the method skips the current round and attempts to eliminate another symbol in Σ.

The final phase: To facilitate the transformation of TBox pivot-clauses (not in reduced form) into reduced form, definer symbols might have been introduced during the elimination rounds. The final phase of the forgetting process attempts to eliminate these definer symbols by using Ackermann-based rules for concept forgetting, which are very similar to the rule in (1) (and its dual); for details see [Koopmann and Schmidt, 2014; Zhao and Schmidt, 2015; 2016]. This allows definer symbols in many cases to be eliminated, because occurrences of one polarity of any definer symbol will be top-level occurrences. There is no guarantee however that all definer symbols can be eliminated, even if we use the generalization of Ackermann's Lemma involving the use of fixpoint operators. In practice, most real-world ontologies are normalized and therefore in reduced form, which means that for such ontologies definer introduction and elimination are obsolete.

Output: What the method returns as output at the end of the forgetting process is a finite set $\mathcal{O}'$ of clauses. If $\mathcal{O}'$ does not contain any symbols in Σ, then the method was *successful* (in computing a solution of forgetting Σ from $\mathcal{O}$). The following theorem states termination and soundness of the method.

Theorem 4 *For any $\mathcal{ALCOQH}(\triangledown)$-ontology $\mathcal{O}$ and any set $\Sigma \subseteq \mathrm{sig}_{\mathsf{R}}(\mathcal{O})$ of role symbols to be forgotten, the method always terminates and returns a finite set $\mathcal{O}'$ of clauses. (i) If $\mathcal{O}'$ does not contain any symbols in Σ or any newly-introduced definer symbols, then $\mathcal{O}'$ is a solution of forgetting Σ from $\mathcal{O}$ (i.e., $\mathcal{O}'$ is equivalent to the original ontology $\mathcal{O}$ up to the symbols in Σ). (ii) If $\mathcal{O}'$ does not contain any symbols in Σ but it contains newly-introduced definer symbols, then $\mathcal{O}'$ is a solution of forgetting Σ from $\mathcal{O}$ in an extended language (and $\mathcal{O}'$ and $\mathcal{O}$ are equivalent up to the symbols in Σ, as well as the newly-introduced definer symbols present in $\mathcal{O}'$).*

The method may return an '$\mathcal{O}'$' that still contains some Σ-symbols. In this case, the method was *not successful*. This is because there is a gap in the scope of the rules in the Ackermann V rule, as mentioned before Theorem 3.

Theorem 5 *Given an $\mathcal{ALCOQH}(\triangledown)$-ontology $\mathcal{O}$ in clausal form and a subset Σ of $\mathrm{sig}_{\mathsf{R}}(\mathcal{O})$, our method is guaranteed to compute a solution of forgetting Σ from $\mathcal{O}$, possibly with concept definer symbols, iff any one of the following conditions holds for each $r \in \Sigma$: (i) $\mathcal{O}$ does not contain any RBox axioms of the form $\neg\mathcal{S} \sqcup r$ for $\mathcal{S} \neq r$; (ii) $\mathcal{O}$ does not contain any TBox axioms with number restrictions of the form $\geq mr.D$ for $m \geq 1$; or (iii) $\mathcal{O}$ does not contain any TBox axioms with number restrictions of the form $\leq nr.D$ for $n \geq 1$.*

An explanation of Case (ii) is the following: let $\mathcal{O}$ be an $\mathcal{ALCOQH}(\triangledown)$-ontology in clausal form, and let Σ be a subset of $\mathrm{sig}_{\mathsf{R}}(\mathcal{O})$. For $r \in \Sigma$ the pivot, if $\mathcal{O}$ does not contain any TBox axioms with number restrictions of the form $\geq mr.D$ for $m \geq 1$, then there will be no positive TBox premises occurring in $\mathcal{O}$ (when $\mathcal{O}$ is transformed into reduced form). $\mathcal{O}$ is thus in the form suitable for application of the Ackermann IV rule. Explanations of Cases (i) and (iii) are similar, i.e., $\mathcal{O}$ of Cases (i) and (iii) in reduced form are suitable for application of the Ackermann III and Ackermann V rules, respectively.

6 Evaluation and Empirical Results

To gain insight into the practical applicability of the method, we implemented a prototype in Java using the OWL-API,[2] and evaluated it on two corpora of slightly adjusted real-world ontologies from the NCBO BioPortal repository.[3] The experiments were run on a desktop computer with an Intel® Core™

[2]http://owlapi.sourceforge.net/

[3]http://bioportal.bioontology.org/

i7-4790 processor, four cores running at up to 3.60 GHz and 8 GB of DDR3-1600 MHz RAM.

Ontology	TA	RA	CS	RS	IS	$\geq$	$\leq$	Expressivity
PANDA	102	44	99	49	0	4	20	$\mathcal{ALCIQH}(\mathcal{D})$
OPB	973	68	779	59	0	179	140	$\mathcal{ALCIQH}(\mathcal{D})$
ROO	1285	296	1183	209	0	278	0	$\mathcal{SHIQ}(\mathcal{D})$
EPO	1995	131	1388	44	0	322	78	$\mathcal{SHIQ}(\mathcal{D})$
SDO	2738	114	1382	77	59	1305	14	$\mathcal{SHOIQ}(\mathcal{D})$

TA = TBox and ABox Axioms, RA = RBox Axioms, CS = Concept Symbols, RS = Role Symbols, IS = Individual Symbols, $\geq$ = $\geq$-restrictions, $\leq$ = $\leq$-restrictions

Figure 3: Ontologies selected from NCBO BioPortal

The corpora used for our experiments were constructed as follows. First, we selected from the NCBO BioPortal repository ontologies containing both number restrictions and role inclusions. Then, we filtered out those containing less than 40 role symbols (since they were less challenging). As a result, 5 ontologies stood out from the repository (see Figure 3 for their profiles). We further adjusted these 5 ontologies to the language of $\mathcal{ALCOQH}$ (i.e., none of them included the universal role $\triangledown$). This was done by removing those axioms not expressible in $\mathcal{ALCOQH}$ and using simple simulations. For example, an exact number restriction $=1r.D$ was simulated by $(\geq 1r.D) \sqcap (\leq 1r.D)$, and a functional role $\mathsf{func}(r)$ was simulated by $\leq 1r.\top$. In this way we obtained a corpus (**Corpus I**) of 5 $\mathcal{ALCOQH}$-ontologies. By removing all role inclusions in each of the ontologies in Corpus I, we obtained another corpus (**Corpus II**) that contained 5 $\mathcal{ALCOQ}$-ontologies. Using Corpora I and II as test data sets for our experiments, we considered how the presence of role inclusions affected the results of role forgetting, in particular, the success rates.

To fit in with real-world use, we evaluated the performance of forgetting different numbers of role symbols from each test ontology. In particular, we forgot 30% (i.e., a relatively small number) and 70% (i.e., a relatively large number) of role symbols in the signature of each test ontology. The symbols to be forgotten were randomly chosen. We ran the experiments 50 times on each ontology and averaged the results to verify the accuracy of our findings. A timeout of 100 seconds was imposed on each run of the experiment (unit of time: sec.).

Σ (30%)	Corpus I				Corpus II			
Ontology	DI	Time	SR	GC	DI	Time	SR	GC
PANDA	0	0.576	100%	0.0%	0	0.571	100%	0.0%
OPB	0	1.734	100%	4.2%	0	1.695	100%	4.3%
ROO	0	4.674	100%	0.0%	0	4.339	100%	0.0%
EPO	0	7.183	100%	7.1%	0	7.171	100%	7.3%
SDO	0	18.325	71.1%	7.3%	0	17.817	69.4%	7.7%
Σ (70%)	Corpus I				Corpus II			
PANDA	0	1.267	100%	0.0%	0	1.252	100%	0.0%
OPB	0	3.937	100%	6.1%	0	3.869	100%	6.5%
ROO	0	9.663	100%	0.0%	0	9.602	100%	0.0%
EPO	0	15.874	100%	8.2%	0	15.389	100%	8.5%
SDO	0	39.196	32.1%	8.5%	0	38.084	30.9%	8.9%

DI = Definers Introduced, SR = Success Rate, GC = Growth of Clauses

Figure 4: Results of forgetting 30% and 70% of role symbols

The results are shown in Figure 4, which is rather revealing in several ways. The most encouraging result was that our implementation was successful (i.e., forgot all symbols in Σ) in all test ontologies (within a short period of time) except in the case of SDO, despite role inclusions being present in them. This was unexpected, but there are obvious explanations (for the 100% success rate cases): inspection revealed that these ontologies did not contain axioms with number restrictions of the form $\leq n\mathcal{S}.D$ for $n \geq 1$, and the likelihood of Σ-symbols occurring positively in the RBox axioms was very low. What was as expected was that definer symbols were not introduced in the test ontologies (as most real-world ontologies were by design flat and therefore already in reduced form). This gave us best benefits of using our Ackermann approach. Because of the nature of the Ackermann III and V rules, forgetting a role symbol could lead to growth of clauses in the forgetting solution, which was however modest (see the GC column in Figure 4) compared to the theoretical worst case (i.e., $2^n - 1$ for n the cardinality of $\mathcal{P}^+_{\mathcal{T}}$). In the case of SDO the 'hasPart' role occurred positively in more than 50 different TBox clauses in reduced form. This means that if 'hasPart' was chosen as one of the Σ-symbols to be forgotten, then there were more than 50 positive TBox premises in the ontology SDO in reduced form (i.e., $n \geq 50$), which led to a blow-up of clauses in the forgetting solution (i.e., $\geq 2^{50} - 1$ clauses). Indeed, the failures on SDO were due to space explosion caused by the high frequency of the 'hasPart' role. We found that without this role in Σ, the success rate was 100%.

7 Conclusion and Future Work

In this paper, we have presented a practical method of semantic role forgetting for ontologies expressible in the description logic $\mathcal{ALCOQH}(\triangledown)$. The method is the only approach so far for forgetting role symbols in description logics with number restrictions. This is very useful from the perspective of ontology engineering as it increases the arsenal of tools available to create decompositions and restricted views of ontologies. We have shown that the method is terminating and is sound in the sense that the forgetting solution is equivalent to the original ontology up to the forgotten symbols, sometimes with new concept definer symbols. Although our method is not complete, performance results of an evaluation with a prototypical implementation have shown very good success rates on two corpora of real-world biomedical ontologies.

Though the focus of this paper has been the problem of role forgetting, (non-nominal) concept forgetting can be reduced to role forgetting by substituting $\geq 1r.\top$ for every occurrence of the concept symbol one wants to forget, where r is a fresh role symbol, and then forgetting $\{r\}$. For example, forgetting the concept symbol $\{B\}$ from $\{\neg A \sqcup \geq 1s.B\}$ can be reduced to forgetting the role symbol $\{r\}$ from $\{\neg A \sqcup \geq 1s.\geq 1r.\top\}$. Thus our method also provides an incomplete approach to concept forgetting for $\mathcal{ALCOQH}(\triangledown)$-ontologies.

A natural next step for future work is to extend the method to accommodate transitive properties on roles, though it is realized when forgetting role symbols, the interaction between transitivity and role inclusions can lead to results where it is not clear how to represent them finitely [Koopmann, 2015].

Acknowledgments

We thank the anonymous referees for very useful comments.

References

[Ackermann, 1935] W. Ackermann. Untersuchungen über das Eliminationsproblem der mathematischen Logik. *Mathematische Annalen*, 110(1):390–413, 1935.

[Bicarregui *et al.*, 2001] J. Bicarregui, T. Dimitrakos, D. M. Gabbay and T. S. E. Maibaum Interpolation in Practical Formal Development. *Logic Journal of the IGPL*, 9(2):231–244, 2001.

[Botoeva *et al.*, 2016] E. Botoeva, B. Konev, C .Lutz, V .Ryzhikov, F. Wolter and M. Zakharyaschev Inseparability and Conservative Extensions of Description Logic Ontologies: A Survey. In *Proc. RW'16*, vol. 9885 of *LNCS*, pp. 27–89. Springer, 2016.

[Conradie *et al.*, 2006] W. Conradie, V. Goranko, and D. Vakarelov. Algorithmic correspondence and completeness in modal logic. I. The core algorithm SQEMA. *Logical Methods in Computer Science*, 2(1), 2006.

[D'Agostino and Hollenberg, 2000] G. D'Agostino and M. Hollenberg. Logical Questions Concerning the μ-Calculus: Interpolation, Lyndon and Los-Tarski. *Journal of Symbolic Logic*, 65(1):310–332, 2000.

[Delgrande and Wang, 2015] J. P. Delgrande and K. Wang. A Syntax-Independent Approach to Forgetting in Disjunctive Logic Programs. In *Proc. AAAI'15*, pp. 1482–1488. AAAI Press, 2015.

[Doherty *et al.*, 1997] P. Doherty, W. Łukaszewicz, and A. Szałas. Computing Circumscription Revisited: A Reduction Algorithm. *Journal of Automated Reasoning*, 18(3):297–336, 1997.

[Gabbay *et al.*, 2008] D. M. Gabbay, R. A. Schmidt, and A. Szałas. *Second Order Quantifier Elimination*. College Publi., 2008.

[Grau and Motik, 2012] B. C. Grau and B. Motik. Reasoning over Ontologies with Hidden Content: The Import-by-Query Approach. *J. Artif. Intell. Res.*, 45:197–255, 2012.

[Herzig and Mengin, 2008] A. Herzig and J. Mengin. Uniform Interpolation by Resolution in Modal Logic. In *Proc. JELIA'08*, vol. 5293 of *LNCS*, pp. 219–231. Springer, 2008.

[Konev *et al.*, 2009a] B. Konev, C. Lutz, D. Walther and F. Wolter. Formal Properties of Modularisation. In *Modular Ontologies*, vol. 5445 of *LNCS*, pp. 25–66. Springer, 2009.

[Konev *et al.*, 2009b] B. Konev, D. Walther and F. Wolter. Forgetting and Uniform Interpolation in Large-Scale Description Logic Terminologies. In *Proc. IJCAI'09*, pp. 830–835. IJCAI/AAAI Press, 2009.

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

[Koopmann and Schmidt, 2013a] P. Koopmann and R. A. Schmidt. Forgetting Concept and Role Symbols in $\mathcal{ALCH}$-Ontologies. In *Proc. LPAR'13*, vol. 8312 of *LNCS*, pp. 552–567. Springer, 2013.

[Koopmann and Schmidt, 2013b] P. Koopmann and R. A. Schmidt. Uniform Interpolation of $\mathcal{ALC}$-Ontologies Using Fixpoints. In *Proc. FroCoS'13*, vol. 8152 of *LNCS*, pp. 87-102. Springer, 2013.

[Koopmann and Schmidt, 2014] P. Koopmann and R. A. Schmidt. Count and Forget: Uniform Interpolation of $\mathcal{SHQ}$-Ontologies. In *Proc. IJCAR'14*, vol. 8562 of *LNCS*, pp. 434–448. Springer, 2014.

[Koopmann and Schmidt, 2015b] P. Koopmann and R. A. Schmidt. Saturated-Based Forgetting in the Description Logic $\mathcal{SIF}$. In *Proc. DL'15*, vol. 1350 of *CEUR Workshop Proceedings*. 2015.

[Koopmann and Schmidt, 2015a] P. Koopmann and R. A. Schmidt. Uniform Interpolation and Forgetting for $\mathcal{ALC}$ Ontologies with ABoxes. In *Proc. AAAI'15*, pp. 175–181. AAAI Press, 2015.

[Koopmann, 2015] P. Koopmann. *Practical Uniform Interpolation for Expressive Description Logics*. PhD Thesis, The University of Manchester, UK, 2015.

[Lang *et al.*, 2003] J. Lang, P. Liberatore and P. Marquis. Propositional Independence: Formula-Variable Independence and Forgetting. *J. Artif. Intell. Res.*, 18:391–443, 2003.

[Lin and Reiter, 1994] F. Lin and R. Reiter. Forget it! In *Proc. AAAI Fall Symposium on Relevance*, pp. 154–159, 1994.

[Ludwig and Konev, 2014] M. Ludwig and B. Konev. Practical Uniform Interpolation and Forgetting for $\mathcal{ALC}$ TBoxes with Applications to Logical Difference. In *Proc. KR'14*, pp. 318–327. AAAI Press, 2014.

[Lutz and Wolter, 2011] C. Lutz and F. Wolter. Foundations for Uniform Interpolation and Forgetting in Expressive Description Logics. In *Proc. IJCAI'11*, pp. 989–995. IJCAI/AAAI Press, 2011.

[Lutz *et al.*, 2012] C. Lutz, I. Seylan, and F. Wolter. An Automata-Theoretic Approach to Uniform Interpolation and Approximation in the Description Logic $\mathcal{EL}$. In *Proc. KR'12*, pp. 286–297. AAAI Press, 2012.

[Nikitina and Rudolph, 2014] N. Nikitina and S. Rudolph. (Non-)Succinctness of uniform interpolants of general terminologies in the description logic $\mathcal{EL}$. *Artificial Intelligence*, 215:120–140, 2014.

[Nonnengart and Szałas, 1999] A. Nonnengart and A. Szałas. A fixpoint approach to second-order quantifier elimination with applications to correspondence theory. In E. Orlowska, editor, *Logic at Work*, pp. 307–328. Springer, 1999.

[Schmidt, 2012] R. A. Schmidt. The Ackermann approach for modal logic, correspondence theory and second-order reduction. *Journal of Applied Logic*, 10(1):52–74, 2012.

[Szałas, 1993] A. Szałas. On the Correspondence between Modal and Classical Logic: An Automated Approach. *Journal of Logic and Computation*, 3:605–620, 1993.

[Szałas, 2006] A. Szałas. Second-order reasoning in description logics. *Journal of Applied Non-Classical Logics*, 16(3-4):517–530, 2006.

[Visser, 1996] A. Visser. *Bisimulations, Model Descriptions and Propositional Quantifiers*. Logic Group Preprint Series. Department of Philosophy, Utrecht University, 1996.

[Wang *et al.*, 2008] Z. Wang, K. Wang, R. W. Topor, and J. Z. Pan. Forgetting Concepts in DL-Lite. In *Proc. ESWC'08*, vol. 5021 of *LNCS*, pp. 245–257. Springer, 2008.

[Wang *et al.*, 2010] Z. Wang, K. Wang, R. W. Topor and J. Z. Pan. Forgetting for Knowledge Bases in DL-Lite. *Annals of Math. Artif. Intell.*, 58(1-2):117–151, 2010.

[Wang *et al.*, 2014] K. Wang, Z. Wang, R. Topor, J. Z. Pan, and G. Antoniou. Eliminating concepts and roles from ontologies in expressive description logics. *Computational Intelligence*, 30(2):205–232, 2014.

[Zhao and Schmidt, 2015] Y. Zhao and R. A. Schmidt. Concept Forgetting in $\mathcal{ALCOI}$-Ontologies Using an Ackermann Approach. In *Proc. ISWC'15*, vol. 9366 of *LNCS*, pp. 587–602. Springer, 2015.

[Zhao and Schmidt, 2016] Y. Zhao and R. A. Schmidt. Forgetting Concept and Role Symbols in $\mathcal{ALCOIH}\mu^+(\nabla,\sqcap)$-Ontologies. In *Proc. IJCAI'16*, pp. 1345–1352. IJCAI/AAAI Press, 2016.