

Concept and Role Forgetting in $\mathcal{ALCCOITH}_{\mu}^+(\nabla, \sqcap)$ -Ontologies

Yizheng Zhao

Renate A. Schmidt

School of Computer Science, The University of Manchester, UK

Abstract: We revise a recently-developed method for forgetting of concept and role symbols in ontologies expressible in the description logic $\mathcal{ALCCOITH}_{\mu}^+(\nabla, \sqcap)$. Being based on an Ackermann approach, the method is one of only few approaches that can eliminate role symbols, that can handle role inverse and ABox statements, and is the only approach so far providing support for forgetting in description logics with nominals.

1 Introduction

Forgetting is a non-standard reasoning problem concerned with creating restricted views for ontologies relative to subsets of their initial signatures while preserving all logical consequences up to the symbols in the restricted views. It turns out to be a very useful technique in various tasks crucial for effective processing and management of ontologies. For example, forgetting can be used for ontology reuse, for creating ontology summaries, for information hiding, for computing logical difference between ontologies, for ontology debugging and repair, and for query answering.

Early work in the area primarily focused on forgetting concept symbols, as role forgetting was realised to be significantly harder than forgetting of concept symbols, because the result of forgetting role symbols often requires more expressivity than is available in the target logic.

The contribution of this work is a practical method for forgetting of concept and role symbols in expressive description logics not considered so far. The method accommodates ontologies expressible in the description logic $\mathcal{ALCCOITH}$ and the extension allowing positive occurrences (μ^+) of the least fixpoint operator μ , the top role ∇ and role conjunction \sqcap . The added expressivity has the advantage that it reduces information loss; for instance, the solution of forgetting the role symbol r in the ontology $\{A \sqsubseteq \exists r.B, \exists r.B \sqsubseteq B\}$ is $\{A \sqsubseteq \exists \nabla.B, A \sqsubseteq B\}$, whereas in a description logic without the top role (or ABox axioms or nominals) the solution is $\{A \sqsubseteq B\}$, which is weaker.

Definition 1 (Forgetting in $\mathcal{ALCCOITH}_{\mu}^+(\nabla, \sqcap)$) Let \mathcal{O} and \mathcal{O}' be $\mathcal{ALCCOITH}_{\mu}^+(\nabla, \sqcap)$ -ontologies and let Σ be any subset of $\text{sig}(\mathcal{O})$. \mathcal{O}' is the solution of forgetting the Σ -symbols in \mathcal{O} , if the following conditions hold: (i) $\text{sig}(\mathcal{O}') \subseteq \text{sig}(\mathcal{O})$ and $\text{sig}(\mathcal{O}') \cap \Sigma = \emptyset$, 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} .

2 The Forgetting Method

The forgetting process in our method consists of three main phases: the reduction to a set of $\mathcal{ALCCOITH}_{\mu}^+(\nabla, \sqcap)$ -clauses, the forgetting phase and the definer elimination phase (see Figure 1). In the forgetting phase, an analyser may be used to generate forgetting orderings, \succ^C and \succ^R , of the concept symbols and role symbols in Σ .

The input to the method is an ontology \mathcal{O} of TBox and RBox axioms expressible in $\mathcal{ALCCOITH}_{\mu}^+(\nabla, \sqcap)$ (ABox

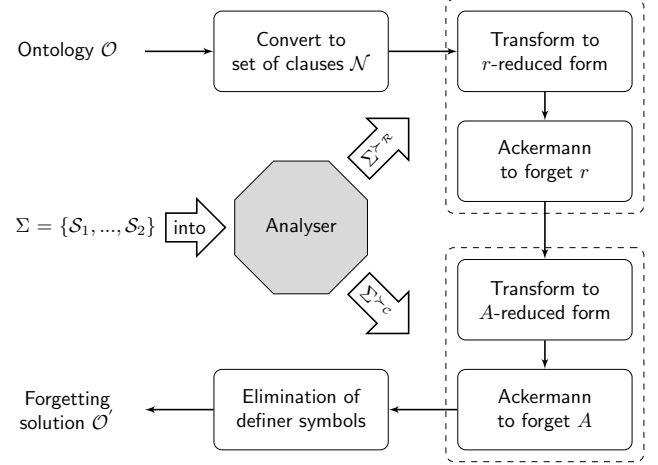


Figure 1: Overview of the forgetting method

axioms are assumed to be equivalently expressed as TBox axioms in our method), and a set Σ with the concept and role symbols to be forgotten. The first phase transforms the input ontology \mathcal{O} into a set \mathcal{N} of clauses.

The forgetting phase is an iteration of several rounds in which individual concept and role symbols are eliminated. An important feature of the method is that concept symbols and role symbols are forgotten in a focused way, i.e., the rules for concept forgetting and the rules for role forgetting are mutually independent; concept and role symbols can thus be forgotten in any order. In the forgetting phase, if $S \in \Sigma$ is a symbol to be forgotten, the idea is to transform the S -clauses into S -reduced form, so that the forgetting rules (the Ackermann rules) can be applied to eliminate S . The conversion to S -reduced form is performed using the rewrite rules in the underlying calculi of our method, which can be found in [4]. To provide crucial control and flexibility in how the steps are performed, auxiliary concept symbols, called *definer symbols*, are introduced in the role forgetting rounds. The final phase attempts to eliminate these definer symbols via concept forgetting.

Previous research has shown that the success rates of forgetting depend very much upon the order in which the Σ -symbols are forgotten [1, 2, 3]. Our method either follows the user-specified ordering, or it uses a heuristic analysis based on frequency counts of the Σ -symbols to generate good orderings. We refer to the maximal symbol of Σ w.r.t. the forgetting ordering \succ as the *pivot* in our method.

Non-Cyclic Ackermann^C	
$\frac{\mathcal{N}, C_1 \sqcup A, \dots, C_n \sqcup A}{\mathcal{N}_{\neg C_1 \sqcup \dots \sqcup \neg C_n}^A}$	
provided: (i) A does not occur in the C_i , and (ii) \mathcal{N} is negative w.r.t. A .	
Cyclic Ackermann^C	
$\frac{\mathcal{N}, C_1[A] \sqcup A, \dots, C_n[A] \sqcup A}{\mathcal{N}_{\mu X.(\neg C_1 \sqcup \dots \sqcup \neg C_n)[X]}^A}$	
provided: (i) the C_i are negative w.r.t. A , and (ii) \mathcal{N} is negative w.r.t. A .	
Purify^C	
$\frac{\mathcal{N}}{\mathcal{N}_{(\neg)\top}^A}$	provided: \mathcal{N} is positive (negative) w.r.t. A .

Figure 2: Rules for forgetting pivot $A \in \mathbf{N}_C$

Theorem 1 For any $\mathcal{ALCCOITH}_{\mu^+}(\nabla, \sqcap)$ -ontology \mathcal{O} and any set $\Sigma \subseteq \text{sig}(\mathcal{O})$ of symbols to be forgotten, the method always terminates and returns a set \mathcal{O}' of clauses. If \mathcal{O}' does not contain any Σ -symbols, the method was successful. \mathcal{O}' is then a solution of forgetting the symbols in Σ from \mathcal{O} . If neither \mathcal{O} nor \mathcal{O}' uses fixpoints, \mathcal{O}' is Σ -equivalent to \mathcal{O} in $\mathcal{ALCCOITH}(\nabla, \sqcap)$. Otherwise, it is Σ -equivalent to \mathcal{O} in $\mathcal{ALCCOITH}_{\mu^+}(\nabla, \sqcap)$.

3 The Forgetting Rules

Let \mathbf{N}_C and \mathbf{N}_R be sets of *concept symbols* and *role symbols*.

Definition 2 (A-Reduced Form) For $A \in \mathbf{N}_C$, a clause is in A -reduced form if it is negative w.r.t. A , or it has the form $A \sqcup C$, where C is an $\mathcal{ALCCOITH}_{\mu^+}(\nabla, \sqcap)$ -concept that does not have any occurrences of A , or is negative w.r.t. A . A set \mathcal{N} of clauses is in A -reduced form if every A -clause in \mathcal{N} is in A -reduced form.

The (Non-)Cyclic Ackermann^C rules and the Purify^C rule, given in Figure 2, are the *forgetting rules* that lead to the elimination of concept symbols in a set of clauses. For $A \in \mathbf{N}_C$ the pivot and C a concept expression, \mathcal{N}_C^A denotes the set obtained from \mathcal{N} by replacing every occurrence of A by C . The (Non-)Cyclic Ackermann^C rules are applied only if \mathcal{N} is in A -reduced form. The Purify^C rule can be applied anytime provided A is *pure* in \mathcal{N} , i.e., every occurrence of A in \mathcal{N} is positive or negative (under an even number of explicit and implicit negations or otherwise).

Definition 3 (r-Reduced Form) For $r \in \mathbf{N}_R$, a clause is in r -reduced form if it has the form $C \sqcup \forall r.D$ or $C \sqcup \neg \forall r \sqcap Q.D$, where C and D are concepts that do not contain any occurrence of r and Q is a role that does not contain any occurrence of r ; or it has the form $\neg S \sqcup r$ or $S \sqcup \neg r$, where S is a role symbol or an inverted role symbol. A set \mathcal{N} of clauses is in r -reduced form if every r -clause in \mathcal{N} is in r -reduced form.

Finding the r -reduced form of a clause is not always possible, unless definer symbols are introduced. *Definer symbols* are specialised concept symbols that do not occur in the present ontology, and are introduced as follows: given a clause of the form $C \sqcup \forall r^{(-)}.D$ or $C \sqcup \neg \forall r^{(-)}.D$, with

Ackermann^R	
$\mathcal{N}, \boxed{C^1 \sqcup \neg \forall r \sqcap Q^1.D^1, \dots, C^k \sqcup \neg \forall r \sqcap Q^k.D^k},$	
$\boxed{\neg T^1 \sqcup r, \dots, \neg T^u \sqcup r},$	
$\boxed{\neg C_1 \sqcup \forall r.D_1, \dots, \neg C_m \sqcup \forall r.D_m,}$	
$\neg r \sqcup S_1, \dots, \neg r \sqcup S_n$	
<hr/>	
$\mathcal{N}, \boxed{\mathcal{T}\text{-Block}^{\mathcal{H}}(1, m), \dots, \mathcal{T}\text{-Block}^{\mathcal{H}}(k, m)},$	
$\boxed{\mathcal{R}\text{-Block}^C(1, m), \dots, \mathcal{R}\text{-Block}^C(u, m)},$	
$\boxed{\mathcal{R}\text{-Block}^R(1, n), \dots, \mathcal{R}\text{-Block}^R(u, n)},$	
<hr/>	
provided: (i) r does not occur in \mathcal{N} , and (ii) \mathcal{N} is in r -reduced form.	
Purify^R	
$\frac{\mathcal{N}}{\mathcal{N}_{(\neg)\nabla}^r}$	provided: \mathcal{N} is positive (negative) w.r.t. r .
Notation in the Ackermann ^R rule ($1 \leq j \leq k, 1 \leq l \leq u$):	
$\mathcal{T}\text{-Block}^{\mathcal{H}}(j, m)$ denotes the set	
$\{C^j \sqcup C^Y \sqcup \neg \forall \mathcal{H}^j.(D^j \sqcup D^Y) \mid Y \subseteq [m]\}$, where	
$C^Y = \begin{cases} \neg \top & \text{if } Y = \emptyset \\ \bigcup_{i \in Y} C_i & \text{otherwise,} \end{cases} \quad D^Y = \begin{cases} \neg \top & \text{if } Y = \emptyset \\ \bigcup_{i \in Y} \neg D_i & \text{otherwise,} \end{cases}$	
and $\mathcal{H} = \begin{cases} S_1 \sqcap \dots \sqcap S_n \sqcap Q^j & \text{if } \mathcal{P}_{\mathcal{R}}^- \neq \emptyset \\ \nabla \sqcap Q^j & \text{otherwise.} \end{cases}$	
$\mathcal{R}\text{-Block}^C(l, m)$ denotes the set	
$\{C_1 \sqcup \forall T^l.D_1, \dots, C_m \sqcup \forall T^l.D_m\}.$	
$\mathcal{R}\text{-Block}^R(l, n)$ denotes the set $\{\neg T^l \sqcup S_1, \dots, \neg T^l \sqcup S_n\}.$	

Figure 3: Rules for forgetting pivot $r \in \mathbf{N}_R$

r being the pivot and occurring in $Q \in \{C, D\}$, the definer symbols are used as substitutes, incrementally replacing C and D until neither contains any occurrences of r . A new clause $\neg D \sqcup Q$ is added to the clause set for each replaced subconcept Q , where D is a fresh definer symbol.

Given a set \mathcal{N} of clauses with $r \in \mathbf{N}_R$ being the pivot, once \mathcal{N} has been transformed into r -reduced form, we apply the Ackermann^R rule given in Figure 3 to eliminate r . The Purify^R rule can be applied anytime provided r is pure.

References

- [1] P. Koopmann and R. A. Schmidt. Count and forget: Uniform interpolation of \mathcal{SHQ} -ontologies. In *Proc. IJCAR'14*, volume 8562 of *Lecture Notes in Computer Science*, pages 434–448. Springer, 2014.
- [2] R. A. Schmidt. The Ackermann approach for modal logic, correspondence theory and second-order reduction. *Journal of Applied Logic*, 10(1):52–74, 2012.
- [3] Y. Zhao and R. A. Schmidt. Concept Forgetting in $\mathcal{ALCCOIT}$ -Ontologies Using an Ackermann Approach. In *Proc. ISWC'15*, volume 9366 of *Lecture Notes in Computer Science*, pages 587–602. Springer, 2015.
- [4] Y. Zhao and R. A. Schmidt. Forgetting concept and role symbols in $\mathcal{ALCCOITH}_{\mu^+}(\nabla, \sqcap)$ -ontologies. In *Proc. DL'16*, volume 1577 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2016. To appear in *Proc. IJCAI'16*.