Count and Forget: Uniform Interpolation of $\mathcal{SHQ}$-Ontologies

Patrick Koopmann,* Renate A. Schmidt

The University of Manchester, UK

Abstract. We propose a method for forgetting concept symbols and non-transitive roles symbols of $\mathcal{SHQ}$-ontologies, or for computing uniform interpolants in $\mathcal{SHQ}$. Uniform interpolants restrict the symbols occuring in an ontology to a specified set, while preserving all logical entailments that can be expressed using this set in the description logic under consideration. Uniform interpolation has applications in ontology reuse, information hiding and ontology analysis, but so far no method for computing uniform interpolants for expressive description logics with number restrictions has been developed. Our results are not only interesting because they allow to compute uniform interpolants of ontologies using a more expressive language. Using number restrictions also allows to preserve more information in uniform interpolants of ontologies in less complex logics, such as $\mathcal{ALC}$ or $\mathcal{EL}$. The presented method computes uniform interpolants on the basis of a new resolution calculus for $\mathcal{SHQ}$. The output of our method is expressed using $\mathcal{SHQ}\mu$, which is $\mathcal{SHQ}$ extended with fixpoint operators, to always enable a finite representation of the uniform interpolant. If the uniform interpolant uses fixpoint operators, it can be represented in $\mathcal{SHQ}$ without fixpoints operators using additional concept symbols or by approximation.

1 Introduction

Ontologies are at the center of the semantic web and knowledge-based systems in an increasing number of domains. They model terminological domain knowledge and are usually represented using a description logic to allow reasoning to be performed automatically. Uniform interpolation and forgetting deal with the problem of reducing the vocabulary used in an ontology in such a way that entailments expressed in this reduced vocabulary are preserved. Eliminating concepts or relations from an ontology is referred to as *forgetting* them, and the result is a *uniform interpolant* for the reduced vocabulary.

Uniform interpolation has applications in a range of areas. **(i) Ontology Reuse and Distributed Ontologies.** Big ontologies such as the National Cancer Institute Thesaurus often cover a huge amount of terms, whereas for applications often only a subset is needed. A uniform interpolant can provide a basis in applications where too many symbols in the ontology that users are unfamiliar

* Patrick Koopmann is supported by an EPSRC doctoral training award.

with could be harmful [20]. **(ii) Information Hiding.** In a lot of applications, an ontology may be used by a number of people with different privileges. For such an environment it is crucial to have safe techniques to hide confidential information from users that are not privileged to access them [4]. Uniform interpolation provides a way to remove confidential concepts and relations from an ontology without affecting the entailments over the remaining terminology. **(iii) Understanding concept relations.** Relations between concepts in big ontologies are often indirect and hard to understand with growing complexity of the ontology. Uniform interpolation can be used to compute an ontology that only uses a small number of symbols of interest, to get a direct representation of the relations between them [7]. **(iv) Ontology Maintenance.** A related task is understanding how changes to an ontology, for example the addition of new concept definitions, affect the meaning of other concepts. Uniform interpolants can be used to determine whether the meaning of certain concepts changed, and to get a direct representation of these changes [12]. Further applications of uniform interpolation can be found in [7, 14].

So far, the only expressive description logics for which methods for computing uniform interpolants exists are $\mathcal{ALC}$ and $\mathcal{ALCH}$ [10, 8, 12, 19]. In this paper we extend the methods of [10, 8] to the description logic $\mathcal{SHQ}$, which extends $\mathcal{ALCH}$ with transitive roles and number restrictions. This way, we broaden the application of uniform interpolants to ontologies that use a more expressive description logic. But the expressivity of the underlying description logic also determines what information is included in the uniform interpolant. Consider for example the following simple $\mathcal{ALC}$-ontology $\mathcal{T}_{bike}$.

$$\begin{aligned} \mathsf{Bicycle} &\sqsubseteq \exists \mathsf{hasWheel}.\mathsf{FrontWheel} \sqcap \exists \mathsf{hasWheel}.\mathsf{RearWheel} \\ \mathsf{FrontWheel} &\sqsubseteq \mathsf{Wheel} \sqcap \neg \mathsf{RearWheel} \\ \mathsf{RearWheel} &\sqsubseteq \mathsf{Wheel} \sqcap \neg \mathsf{FrontWheel} \end{aligned}$$

This TBox states that every bicycle has a front wheel and a rear wheel, and that those are disjoint types of wheels. Assume we are not interested in the distinction between front wheels and rear wheels. If we want to preserve all logical entailments in $\mathcal{ALC}$ over the remaining symbols Bicycle, hasWheel and Wheel, this can be done by the single TBox axiom $\mathsf{Bicycle} \sqsubseteq \exists \mathsf{hasWheel}.\mathsf{Wheel}$, which states that every bicycle has a wheel, and which is the $\mathcal{ALC}$-uniform interpolant of $\mathcal{T}_{bike}$ for $\{\mathsf{Bicycle}, \mathsf{hasWheel}, \mathsf{Wheel}\}$. We do lose however the indirectly expressed information that a bicycle has at least two wheels, since we cannot express this in $\mathcal{ALC}$ without using at least one of the concepts FrontWheel and RearWheel. Using number restrictions however, we can express this. The $\mathcal{SHQ}$-uniform interpolant of $\mathcal{T}_{bike}$ consists therefore of the axiom $\mathsf{Bicycle} \sqsubseteq \geq 2\mathsf{hasWheel}.\mathsf{Wheel}$, which states that every bicycle has at least two wheels.

The results in [10, 8, 12] suggest that resolution-based approaches allow for an efficient computation of uniform interpolants in a lot of cases, since they make it possible to derive consequences for a specified symbol in a goal-oriented manner. Motivated by this, we follow a similar approach as in [10, 8]. In Section 4, we present a new resolution calculus for $\mathcal{SHQ}$. Based on this calculus, we present

respectively two methods for forgetting concept symbols and non-transitive role symbols in Sections 5 and 6. Since a finite representation of uniform interpolants is not always possible in pure $\mathcal{SHQ}$, the result may involve the use of fixpoint operators. This way uniform interpolants can always be represented finitely. For this reason, the output of our method is at worst represented in $\mathcal{SHQ}\mu$, which is $\mathcal{SHQ}$ extended with fixpoint operators. If fixpoint operators are not desired, it is possible to obtain a finite presentation in $\mathcal{SHQ}$ using additional symbols, or to approximate the uniform interpolant.

All proofs, some examples and an empirical evaluation of our method are provided in the long version of this paper [11].

2 Definition of $\mathcal{SHQ}\mu$ and Uniform Interpolation

To begin with, we define the description logic $\mathcal{SHQ}\mu$, which is $\mathcal{SHQ}$ extended with fixpoint operators.

Let N_r be a set of *role symbols*. An *RBox* $\mathcal{R}$ is a set of *role axioms* of the form $r \sqsubseteq s$ (*role inclusion*), $r \equiv s$ (*role equivalence*) and $\mathsf{trans}(r)$ (*transitivity axiom*), where $r, s \in N_r$. $r \equiv s$ is defined as abbreviation for the two role inclusions $r \sqsubseteq s$ and $s \sqsubseteq r$. Given an RBox $\mathcal{R}$, we denote by $\sqsubseteq_\mathcal{R}$ the reflexive transitive closure of the role inclusions in $\mathcal{R}$. A role r is *transitive* in $\mathcal{R}$ if $\mathsf{trans}(r) \in \mathcal{R}$. r is *simple* in $\mathcal{R}$ if there is no role s with $s \sqsubseteq_\mathcal{R} r$ and $\mathsf{trans}(s) \in \mathcal{R}$.

Let N_c and N_v be two sets of respectively *concept symbols* and *concept variables*. $\mathcal{SHQ}\mu$-concepts have the following form:

$$\bot \mid A \mid X \mid \neg C \mid C \sqcup D \mid \geq nr.C \mid \nu X.C[X],$$

where $A \in N_c$, $X \in N_v$, $r \in N_r$, C and D are arbitrary concepts, n is a non-zero natural number, and $C[X]$ is a concept expression in which X occurs under an even number of negations. We define further concept expressions as abbreviations: $\top = \neg\bot$, $C \sqcap D = \neg(\neg C \sqcup \neg D)$, $\leq mr.C = \neg(\geq nr.C)$ with $m = n - 1$, $\exists r.C = \geq 1r.C$, $\forall r.C = \leq 0r.\neg C$ and $\mu X.C[X] = \neg\nu X.\neg C[X/\neg X]$, where $C[E_1/E_2]$ denotes the concept obtained by replacing every E_1 in C by E_2. Concepts of the form $\geq nr.C$ and $\leq nr.C$ are called *number restrictions*, and concepts of the form $\nu X.C[X]$ and $\mu X.C[X]$ are called *fixpoint expressions*. $\nu X.C[X]$ and $\mu X.C[X]$ denote respectively the *greatest* and the *least fixpoint* of $C[X]$, and ν and μ are respectively the *greatest* and *least fixpoint operator*. A concept variable X is *bound* if it occurs in the scope $C[X]$ of a fixpoint expression $\nu X.C[X]$ or $\mu X.C[X]$. Otherwise it is *free*. A concept is *closed* if it does not contain any free variables, otherwise it is *open*.

A *TBox* $\mathcal{T}$ is a set of *concept axioms* of the forms $C \sqsubseteq D$ (*concept inclusion*) and $C \equiv D$ (*concept equivalence*), where C and D are closed concepts. $C \equiv D$ is short-hand for the two concept axioms $C \sqsubseteq D$ and $D \sqsubseteq C$. An *ontology* $\mathcal{O} = \langle \mathcal{T}, \mathcal{R} \rangle$ consists of a TBox $\mathcal{T}$ and an RBox $\mathcal{R}$ with the additional restriction that non-simple roles in $\mathcal{R}$ occur only in number restrictions of the form $\leq 0r.C$ or $\geq 1r.C$ in $\mathcal{T}$. This restriction in necessary to ensure decidability of common $\mathcal{SHQ}$

reasoning tasks [6], and our method for uniform interpolation assumes that it is satisfied.

Next, we define the semantics of $\mathcal{SHQ}\mu$. An *interpretation* $\mathcal{I}$ is a pair $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ of the *domain* $\Delta^{\mathcal{I}}$ is a nonempty set and the *interpretation function* $\cdot^{\mathcal{I}}$ assigns to each concept symbol $A \in N_c$ a subset of $\Delta^{\mathcal{I}}$ and to each role symbol $r \in N_r$ a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The interpretation function is extended to $\mathcal{SHQ}\mu$-concepts as follows.

$$\bot^{\mathcal{I}} = \emptyset \qquad (\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \qquad (C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$$
$$(\geq nr.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \#\{(x,y) \in r^{\mathcal{I}} \mid y \in C^{\mathcal{I}}\} \geq n\}$$

The semantics of fixpoint expressions is defined following [2]. Whereas concept symbols are assigned fixed subsets of the domain, concept variables range over arbitrary subsets, which is why only closed concepts have a fixed interpretation. Open concepts are interpreted using *valuations* ρ that map concept variables to subsets of $\Delta^{\mathcal{I}}$. Given a valuation ρ and a set $W \subseteq \Delta^{\mathcal{I}}$, $\rho[X \mapsto W]$ denotes a valuation identical to ρ except that $\rho[X \mapsto W](X) = W$. Given an interpretation $\mathcal{I}$ and a valuation ρ, the function $\cdot^{\mathcal{I}}_{\rho}$ is $\cdot^{\mathcal{I}}$ extended with the cases $X^{\mathcal{I}}_{\rho} = \rho(X)$ and

$$(\nu X.C)^{\mathcal{I}}_{\rho} = \bigcup \{W \subseteq \Delta^{\mathcal{I}} \mid W \subseteq C^{\mathcal{I},\rho[X \mapsto W]}\}.$$

If C is closed, we define $C^{\mathcal{I}} = C^{\mathcal{I}}_{\rho}$ for any valuation ρ. Since C does not contain any free variables in this case, this defines $C^{\mathcal{I}}$ uniquely.

A concept inclusion $C \sqsubseteq D$ is *true* in an interpretation $\mathcal{I}$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, a role inclusion $r \sqsubseteq s$ is true in $\mathcal{I}$ iff $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$ and a transitivity axiom $\mathsf{trans}(r)$ is true in $\mathcal{I}$ if for any domain elements $x, y, z \in \Delta^{\mathcal{I}}$ we have $(x,z) \in r^{\mathcal{I}}$ if $(x,y), (y,z) \in r^{\mathcal{I}}$. $\mathcal{I}$ is a *model* of an ontology $\mathcal{O}$ if all axioms in $\mathcal{O}$ are true in $\mathcal{I}$. An ontology $\mathcal{O}$ is *satisfiable* if there exists a model for $\mathcal{O}$, otherwise it is *unsatisfiable*. Two TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ are *equi-satisfiable* if every model of $\mathcal{T}_1$ can be extended to a model of $\mathcal{T}_2$, and vice versa. $\mathcal{T} \models C \sqsubseteq D$ holds iff in every model $\mathcal{I}$ of $\mathcal{T}$ we have $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. If an axiom α is true in all models of $\mathcal{O}$, we write $\mathcal{O} \models \alpha$. Observe that $\mathcal{O} \models r \sqsubseteq s$ iff $r \sqsubseteq_{\mathcal{R}} s$. Interestingly, allowing number restrictions and fixpoint operators does not affect the complexity of deciding satisfiability of ontologies: for $\mathcal{SHQ}$ as well as for $\mathcal{SHQ}\mu$ it is ExpTime-complete [17, 2].[1]

Let $sig(E)$ denote the concept and role symbols occurring in E, where E can denote a concept, an axiom, a TBox, an RBox or an ontology.

Definition 1 (Uniform Interpolation). *Given an ontology $\mathcal{O}$ and a set of concept and role symbols $\mathcal{S}$, an ontology $\mathcal{O}^{\mathcal{S}}$ is a* uniform interpolant of $\mathcal{O}$ for $\mathcal{S}$ *iff the following conditions are satisfied:*

1. *$sig(\mathcal{O}^{\mathcal{S}}) \subseteq \mathcal{S}$, and*
2. *$\mathcal{O}^{\mathcal{S}} \models \alpha$ iff $\mathcal{O} \models \alpha$ for every $\mathcal{SHQ}$-axiom α with $sig(\alpha) \subseteq \mathcal{S}$.*

[1] [2] proves only ExpTime-completeness for $\mathcal{ALCQ}\mu$, but the result can be easily extended to incorporate role hierarchies and transitive roles using the technique proposed in [16].

3 The Normal Form

Our method for computing uniform interpolants in $\mathcal{SHQ}$ is based on a new resolution calculus $Res_{\mathcal{SHQ}}$ which provides a decision procedure for satisfiability of $\mathcal{SHQ}$-ontologies, and which allows for goal-oriented elimination of concept symbols. Before this calculus can be applied to an ontology, its TBox has to be normalised into a set of clauses using structural transformation or flattening. Let $N_d \subseteq N_c$ be a set of *definer (concept) symbols* that is disjoint with the signature of the given TBox.

Definition 2 (Normal form). *An* $\mathcal{SHQ}$-literal *is a concept description of the form* A, $\neg A$, $\geq nr.\mathcal{D}$ *or* $\leq mr.\neg\mathcal{D}$, *where* $A \in N_c$, $r \in N_r$, $n \geq 1$, $m \geq 0$ *are natural numbers and* $\mathcal{D} = D_1 \sqcup \ldots \sqcup D_n$ *is a disjunction of definer symbols. A literal of the form* $\neg D, D \in N_d$, *is called* negative definer literal. *An* $\mathcal{SHQ}$-clause *is an unordered set of* $\mathcal{SHQ}$*-literals* $l_1, \ldots, l_n$, *represented as* $l_1 \sqcup \ldots \sqcup l_2$. *The empty clause and the empty disjunction are represented as* $\bot$. $\mathcal{SHQ}$*-clauses are abbreviations for TBox axioms of the respective forms* $\top \sqsubseteq l_1 \sqcup \ldots \sqcup l_2$ *and* $\top \sqsubseteq \bot$. *A TBox is in* $\mathcal{SHQ}$-clausal form *if every axiom in it is an* $\mathcal{SHQ}$*-clause.*

Number restrictions of the form $\leq nr.C$ contain a hidden negation of the concept under the restriction (they are equivalent to $\neg\geq(n+1)r.C$). Hence C occurs negatively in $\leq nr.C$. The normal form ensures that every concept under a role restriction occurs positively. This is why $\leq$-literals have the form $\leq nr.\neg\mathcal{D}$.

A TBox is converted into $\mathcal{SHQ}$-clausal form as follows. First we replace existential and universal role restrictions $\exists r.C$ and $\forall r.C$ by corresponding number restrictions $\geq 1r.C$ and $\leq 0r.\neg C$. Then every axiom is converted into negation normal form (every axiom is of the form $\top \sqsubseteq C$, and in C negation only occurs in front of concept symbols or directly under $\leq$-restrictions, and every $\leq$-restriction is of the form $\leq nr.\neg C$). Next, we replace each concept C that occurs under a role restriction of the form $\geq nr.C$ or $\leq nr.\neg C$ by a new concept definer symbol D and add the new axiom $\neg D \sqcup C$ to the TBox. This flattens the TBox, which means every role restriction is of the form $\geq nr.D$ or $\leq nr.\neg D$, where D is a definer symbol. The flattened TBox can be converted into $\mathcal{SHQ}$-clausal form using standard CNF transformations.

Observe that the definition of the $\mathcal{SHQ}$-clausal form allows for disjunctions of arbitrary length under role restrictions $\geq nr.\mathcal{D}$ and $\leq nr.\neg\mathcal{D}$. These disjunctions are not introduced by the initial transformation of the TBox, but may be produced by the rules of the calculus.

Example 1 ($\mathcal{SHQ}$*-normal form).* Consider the TBox $\mathcal{T} = \{A_1 \sqsubseteq \geq 5r.(A \sqcup B), A_2 \sqsubseteq \leq 3r.A\}$. Observe that the normal form requires a negation under each $\leq$-restriction. The $\mathcal{SHQ}$-clausal form of $\mathcal{T}$ consists of the following clauses.

1. $\neg A_1 \sqcup \geq 5r.D_1$
2. $\neg D_1 \sqcup A \sqcup B$
3. $\neg A_2 \sqcup \leq 3r.\neg D_2$
4. $\neg D_2 \sqcup \neg A$

Since the normal form has to be preserved, some rule applications of the calculus require the introduction of new definer symbols that represent the conjunction of existing definer symbols. In particular, during the derivation a new

definer symbol D_{12} is introduced for the conjunction $D_1 \sqcap D_2$ by adding two clauses $\neg D_{12} \sqcup D_1$ and $\neg D_{12} \sqcup D_2$, which are equivalent to the concept inclusion $D_{12} \sqsubseteq D_1 \sqcap D_2$. As exemplified here, throughout the paper we indicate which conjunction an introduced definer symbol represents using its index. To avoid the infinite introduction of new definer symbols, we check whether a definer symbol representing this conjunction already exists. This way the number of introduced definer symbols is limited to 2^k, where k is the number of definer symbols introduced by the initial transformation of the TBox.

4 The Underlying Calculus

We now introduce a sound and refutationally complete calculus $Res_{\mathcal{SHQ}}$ that decides satisfiability of TBoxes in $\mathcal{SHQ}$-normal form. This calculus serves as the basis for the method of computing uniform interpolants.

The calculus consists of the rules shown in Figure 1. Most of the rules are motivated by the tautology $(C_1 \sqcup L_1) \sqcap (C_1 \sqcup L_2) \sqsubseteq (C_1 \sqcup C_2 \sqcup (L_1 \sqcap L_2))$. Therefore, the conclusion often contains literals entailed by $L_1 \sqcap L_2$, where L_1 and L_2 occur in the premises. In the case of the resolution rule, which is known from propositional resolution calculi, we have that $(A \sqcap \neg A)$ entails $\bot$.

For the transitivity rule, observe that $\leq 0 r.\neg D$ is equivalent to $\forall r.D$, and due to the restrictions on $\mathcal{SHQ}$ ontologies, roles with transitive sub-roles do not occur in number restrictions of the form $\leq n r.\neg D$, where $n > 0$. If a domain element a satisfies $\forall r_1.D$, and we have a transitive role $r_2 \sqsubseteq r_1$, the transitive closure of r_2-successors of a are all r_1-successors of a, and they all have to satisfy D. We put this information into clausal form by adding a new cyclic definer symbol D' that is subsumed by D, and by stating that every r_2-successor of a and every r_2-successor of an D'-instance has to satisfy D' (this is similar to what is done in [16] to incorporate transitivity axioms into formulae).

For the $\geq$-combination rule, observe that our normal form does not allow for conjunctions under number restrictions. We can however express the conjunction $\mathcal{D}_1 \sqcap \mathcal{D}_2$ using a disjunction $\mathcal{D}_{12}$ of possibly new definer symbol symbols that represent the conjunctions of each pair of definer symbols from $\mathcal{D}_1$ and $\mathcal{D}_2$. The $\geq$-combination rule becomes more intuitive by interpreting the last two literals of each conclusion as an implication. For example, $\geq (n_1 + n_2) r.(\mathcal{D}_1 \sqcup \mathcal{D}_2) \sqcup \geq 1 r.\mathcal{D}_{12}$ is equivalent to $\leq (n_1 + n_2 - 1) r.(\mathcal{D}_1 \sqcup \mathcal{D}_2) \rightarrow \geq 1 r.\mathcal{D}_{12}$. Figure 2 illustrates the idea. Every column represents an r-successor. If an upper cell is light, it satisfies $\mathcal{D}_1$, if a lower cell is light, it satisfies $\mathcal{D}_2$. The two columns in the middle represent r-sucessors satisfying both $\mathcal{D}_1$ and $\mathcal{D}_2$, that is, satisfying $\mathcal{D}_{12}$. All except the rightmost column represent r-successors satisfying the union $\mathcal{D}_1 \sqcup \mathcal{D}_2$. Depending on how many r-successors satisfy $\mathcal{D}_1 \sqcup \mathcal{D}_2$, the set of r-successors in $\mathcal{D}_{12}$ gets smaller or bigger according to the conclusions of the $\geq$-rule.

For the $\geq\leq$-combination rule, observe that in Figure 2, if there are more elements in $\mathcal{D}_1$ than in $\neg\mathcal{D}_2$, $\mathcal{D}_1$ and $\mathcal{D}_2$ have to overlap. If $\mathcal{D}_1$ contains at least n_1 elements and the complement of $\mathcal{D}_2$ contains at most n_2 elements, the intersection $\mathcal{D}_{12}$ must contain at least $n_1 - n_2$ elements.

Resolution:

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

Transitivity:

$$\frac{C \sqcup \leq 0 r_1.\neg D \qquad \mathsf{trans}(r_2) \in \mathcal{R} \qquad r_2 \sqsubseteq_{\mathcal{R}} r_1}{C \sqcup \leq 0 r_2.\neg D' \qquad \neg D' \sqcup D \qquad \neg D' \sqcup \leq 0 r_2.\neg D'}$$

where D' is a new definer symbol.

$\geq$-Combination:

$$\frac{C_1 \sqcup \geq n_1 r_1.\mathcal{D}_1 \qquad C_2 \sqcup \geq n_2 r_2.\mathcal{D}_2 \qquad r_1 \sqsubseteq_{\mathcal{R}} r \qquad r_2 \sqsubseteq_{\mathcal{R}} r}{\begin{array}{c} C_1 \sqcup C_2 \sqcup \geq (n_1+n_2) r.(\mathcal{D}_1 \sqcup \mathcal{D}_2) \sqcup \geq 1 r.\mathcal{D}_{12} \\ \vdots \\ C_1 \sqcup C_2 \sqcup \geq (n_1+1) r.(\mathcal{D}_1 \sqcup \mathcal{D}_2) \sqcup \geq n_2 r.\mathcal{D}_{12} \end{array}}$$

where $\mathcal{D}_{12} = \bigsqcup_{D_i \in \mathcal{D}_1, D_j \in \mathcal{D}_2} D_{ij}$ represents the conjunction of $\mathcal{D}_1$ and $\mathcal{D}_2$.

$\geq\leq$-Combination:

$$\frac{C_1 \sqcup \geq n_1 r_1.(D_1 \sqcup \ldots \sqcup D_m) \qquad C_2 \sqcup \leq n_2 r_2.\neg D_a \qquad r_1 \sqsubseteq_{\mathcal{R}} r_2}{C_1 \sqcup C_2 \sqcup \geq (n_1 - n_2) r_2.(D_{1a} \sqcup \ldots \sqcup D_{ma})} \quad n_1 > n_2$$

$\geq$-Resolution:

$$\frac{C \sqcup \geq n r.(\mathcal{D} \sqcup D) \qquad \neg D}{C \sqcup \geq n r.\mathcal{D}}$$

$\geq$-Elimination:

$$\frac{C \sqcup \geq n r.\bot}{C}$$

Fig. 1. Inference rules of $Res_{\mathcal{SHQ}}$.

The $\geq$-resolution rule is a variant of the classical resolution rule, and the $\geq$-elimination rule eliminates unsatisfiable literals. The six rules form a sound and refutationally complete calculus for ontologies in $\mathcal{SHQ}$-clausal form, as the following theorem shows.

Theorem 1. *$Res_{\mathcal{SHQ}}$ is sound and refutationally complete. Given any set $\boldsymbol{N}$ of $\mathcal{SHQ}$-clauses and any RBox $\mathcal{R}$, the saturation of $\boldsymbol{N}$ using the rules of $Res_{\mathcal{SHQ}}$ contains the empty clause iff the ontology $\mathcal{O} = \langle \boldsymbol{N}, \mathcal{R} \rangle$ is unsatisfiable.*

Observe that the $\geq$-combination rule can be applied arbitrarily often, resulting in clauses with larger and larger numbers occurring in the number restrictions. For this reason, $Res_{\mathcal{SHQ}}$ on its own is not a decision procedure, since we can derive infinitely many clauses. In order to achieve termination, we need to add redundancy elimination. This is also essential to make the uniform interpolation method practical. Our notion of redundancy is close to the one introduced in [8],

$\mathcal{D}_1$							$\neg\mathcal{D}_1$
$\neg\mathcal{D}_2$						$\mathcal{D}_2$	

Fig. 2. Diagram illustrating the $\geq$- and the $\geq\leq$-combination rules.

but is extended to incorporate number restriction literals and disjunctions under role restrictions.

Definition 3 (Subsumption and Reduction). *A* definer symbol D_1 *is* subsumed by a definer symbol D_2 *($D_1 \sqsubseteq_d D_2$), if either $D_1 = D_2$ or there is a clause $\neg D_1 \sqcup D_2$ in the current clause set. A* disjunction $\mathcal{D}_1$ of definer symbols *is* subsumed by a disjunction $\mathcal{D}_2$ of definer symbols *($\mathcal{D}_1 \sqsubseteq_d \mathcal{D}_2$) if every definer symbol in $\mathcal{D}_1$ is subsumed by a definer symbol in $\mathcal{D}_2$. A* literal l_1 *is* subsumed by a literal l_2 *($l_1 \sqsubseteq_l l_2$) if one of the following is satisfied: (i) $l_1 = l_2$, (ii) $l_1 = \geq n_1 r_1.\mathcal{D}_1$ and $l_2 = \geq n_2 r_2.\mathcal{D}_2$, where $n_1 \geq n_2$, $r_1 \sqsubseteq_{\mathcal{R}} r_2$ and $\mathcal{D}_1 \sqsubseteq_d \mathcal{D}_2$, or (iii) $l_1 = \leq n_1 r_1.\neg\mathcal{D}_1$ and $l_2 = \leq n_2 r_2.\neg\mathcal{D}_2$, where $n_1 \leq n_2$, $r_2 \sqsubseteq_{\mathcal{R}} r_1$ and $\mathcal{D}_1 \sqsubseteq_d \mathcal{D}_2$. A* clause C_1 *is* subsumed by a clause C_2 *($C_1 \sqsubseteq_c C_2$) if every literal in C_1 is subsumed by a literal in C_2. A* clause C *is* redundant with respect to a clause set $\mathbf{N}$*, if $\boldsymbol{N}$ contains a clause C' with $C' \sqsubseteq_c C$.*

The reduction of a disjunction $\mathcal{D}$*, denoted by red($\mathcal{D}$), is obtained from $\mathcal{D}$ by removing every definer symbol from $\mathcal{D}$ that is subsumed by another definer symbol in $\mathcal{D}$. The* reduction of a clause C*, denoted by red(C), is obtained from C by removing every literal that is subsumed by another literal in C and reducing every disjunction that occurs under a number restriction in the remaining literals.*

Observe that the roles and the numbers for $\leq$-restrictions are compared in the other direction as for $\geq$-restrictions. This is due to the hidden negation present in $\leq$-restrictions.

Example 2 (Subsumption and reduction). Assume D_{12} represents $D_1 \sqcap D_2$, which means we have the clauses $\neg D_{12} \sqcup D_1$ and $\neg D_{12} \sqcup D_2$, and we have $r \sqsubseteq s \in \mathcal{R}$. Then $\geq 3r.D_{12}$ is subsumed by $A \sqcup \geq 2s.D_1$ and $\leq 2s.\neg D_{12}$ is subsumed by $B \sqcup \leq 3r.\neg D_2$ ($A \sqcup \geq 2s.D_1$ and $B \sqcup \leq 3r.\neg D_2$ are redundant). The reduction of $\geq 1r.(D_{12} \sqcup D_1)$ is $\geq 1r.D_1$ and the reduction of $\geq 1r.(D_1 \sqcup D_2) \sqcup \geq 2r.D_1$ is $\geq 1r(D_1 \sqcup D_2)$.

In addition to subsumption deletion and reduction, we also remove tautological clauses which contain pairs of contradictory literals. This leads to the set of simplification rules shown in Figure 3. Our use of the terminology for subsumption follows the traditional use in description logics. This means that it is C_2 that is deleted when C_1 is subsumed by C_2 ($C_1 \sqsubseteq_c C_2$), and not vice versa. We denote the calculus $Res_{\mathcal{SHQ}}$ extended with these rules by $Res^{s}_{\mathcal{SHQ}}$.

Theorem 2. *$Res^{s}_{\mathcal{SHQ}}$ is sound and refutationally complete, and provides a decision procedure for $\mathcal{SHQ}$-ontology satisfiability.*

Tautology deletion: $\dfrac{N \cup \{C \sqcup A \sqcup \neg A\}}{N}$

Subsumption deletion: $\dfrac{N \cup \{C_1,\ C_2\}}{N \cup \{C_1\}}$ provided $C_1 \sqsubseteq_c C_2$

Reduction: $\dfrac{N \cup \{C\}}{N \cup \{\mathit{red}(C)\}}$

Fig. 3. Simplification rules of $\mathit{Res}^s_{\mathcal{SHQ}}$.

$\leq$-Combination:

$$\frac{C_1 \sqcup {\leq}n_1 r_1.\neg D_1 \qquad C_2 \sqcup {\leq}n_2 r_2.\neg D_2 \qquad r \sqsubseteq r_1 \qquad r \sqsubseteq r_2}{C_1 \sqcup C_2 \sqcup {\leq}(n_1+n_2) r.\neg D_{12}}$$

$\leq\geq$-Combination:

$$\frac{C_1 \sqcup {\leq}n_1 r_1.\neg D_1 \qquad C_2 \sqcup {\geq}n_2 r_2.D_2 \qquad r_2 \sqsubseteq_{\mathcal{R}} r_1 \qquad n_1 \geq n_2}{\begin{array}{c} C_1 \sqcup C_2 \sqcup {\leq}(n_1-n_2) r_1.\neg(D_1 \sqcup D_2) \sqcup {\geq}1 r_1.D_{12} \\ \vdots \\ C_1 \sqcup C_2 \sqcup {\leq}(n_1-1) r_1.\neg(D_1 \sqcup D_2) \sqcup {\geq}n_2 r_1.D_{12} \end{array}}$$

Fig. 4. Additional inference rules of $\mathit{Forget}_{\mathcal{SHQ}}$.

5 Forgetting Concept Symbols

We reduce the problem of computing uniform interpolants to the problem of forgetting single symbols. We denote the result of forgetting a single symbol x from an ontology $\mathcal{O}$ by $\mathcal{O}^{-x}$, where x can be a role or a concept symbol. $\mathcal{O}^{-x}$ is the uniform interpolant of $\mathcal{O}$ over $\mathit{sig}(\mathcal{O}) \setminus \{x\}$.

The general idea for forgetting a concept symbol A is to saturate the clausal representation of $\mathcal{O}$ in such a way that every clause that cannot be represented in an $\mathcal{SHQ}$-ontology in the signature $\mathit{sig}(\mathcal{O}) \setminus \{A\}$ becomes superfluous. In addition to the rules of $\mathit{Res}^s_{\mathcal{SHQ}}$, we need two more inference rules for the forgetting procedure, which are shown in Figure 4. It turns out that we do not have to consider number restrictions with disjunctions in our rules, which is the situation with all number restrictions after the transformation into $\mathcal{SHQ}$-clausal form.

Assume a domain element has maximally n_1 r-successors satisfying $\neg D_1$ and maximally n_2 r-successors satisfying $\neg D_2$. If we sum them up without further knowledge, we have that there are at most $n_1 + n_2$ r-successors satisfying either $\neg D_1$ or $\neg D_2$. Since formulae under $\leq$-restrictions are negated, and because $(\neg D_1 \sqcup \neg D_2) \equiv \neg(D_1 \sqcap D_2)$, we can verify that the $\leq$-combination rule is sound.

Fig. 5. Diagram illustrating the $\leq\geq$-combination rule.

For the $\leq\geq$-combination rule, we again interpret the last two literals of each conclusion as an implication. For example for the first conclusion, the implication is $\leq 0r.(D_{12}) \rightarrow \leq(n_1 - n_2)r.\neg(D_1 \sqcup D_2)$. That this implication follows from $\leq n_1 r.D_1$ and $\geq n_2 r.D_2$ if $n_1 \geq n_2$ is illustrated in the diagram in Figure 5.

Let $\mathbf{N}$ be the $\mathcal{SHQ}$-normal form of the TBox of $\mathcal{O}$. In order to forget A, or to compute $\mathcal{O}^{-A}$, we saturate $\mathbf{N}$, where we apply resolution only on the symbol A we want to forget, or on definer symbols. The combination rules are only applied if they lead to the introduction of new clauses that make further resolution steps on A possible. For example, if we have the clauses $\leq 5r.\neg D_1$, $\geq 3r.D_2$, $\neg D_1 \sqcup A$ and $\neg D_2 \sqcup \neg A$, we apply the $\leq\geq$-combination rule, since this leads to the introduction of a new definer symbol D_{12}, and, after resolving on the definer symbols, the clauses $\neg D_{12} \sqcup A$ and $\neg D_{12} \sqcup \neg A$. These two clauses can be resolved on A. If we do not have $\neg D_1 \sqcup A$ or $\neg D_2 \sqcup \neg A$, we do not have to apply the $\leq\geq$-combination rule in order to compute the uniform interpolant.

After this saturation is computed, we can remove all clauses that contain the symbol A we want to forget, or that are of the form $\neg D_1 \sqcup D_2$ or $\neg D_1 \sqcup \neg D_2 \sqcup C$. Clauses of the form $\neg D_1 \sqcup D_2$ become superfluous since we computed all resolvents on D_2. Clauses of the form $\neg D_1 \sqcup \neg D_2 \sqcup C$ can also be discarded, as is proved in the long version of this paper.

$\mathbf{N}^{-A}$ is the clausal representation of the result of forgetting A from $\mathbf{N}$, as the following lemma shows.

Lemma 1. *Given a set of clauses $\boldsymbol{N}$ and an RBox $\mathcal{R}$, $\boldsymbol{N}^{-A}$ does not contain A and we have $\langle \boldsymbol{N}^{-A}, \mathcal{R}\rangle \models \alpha$ iff $\langle \boldsymbol{N}, \mathcal{R}\rangle \models \alpha$ for all $\mathcal{SHQ}$-axioms α that do not contain A.*

It remains to eliminate all introduced definer symbols, so that the ontology is completely represented in the desired signature. Since every clause in $\mathbf{N}^{-A}$ contains at most one negative definer literal $\neg D$, we can compute for each definer symbol D a unique concept inclusion $D \sqsubseteq C_1 \sqcap \ldots \sqcap C_n$, where $\neg D \sqcup C_1, \ldots, \neg D \sqcup C_n$ are the clauses in which $\neg D$ occurs outside of a role restriction. We call this concept inclusion the *definition of* D. $D \sqsubseteq C_1 \sqcap \ldots \sqcap C_n$ is equivalent to the set of clauses $\neg D \sqcup C_1, \ldots, \neg D \sqcup C_n$, and we obtain therefore an equivalent TBox by replacing these clauses by the corresponding definitions. We denote the result of this transformation by $\mathcal{T}_D^{-A}$.

In order to compute a TBox representation of $\mathcal{T}_D^{-A}$ without definer symbols, we apply the definer elimination rules shown in Figure 6, where $\mathcal{T}^{[D \mapsto C]}$ denotes the TBox obtained by replacing D with C. If a definer symbol occurs only on the

Non-cyclic definer elimination:

$$\frac{\mathcal{T} \cup \{D \sqsubseteq C\}}{\mathcal{T}^{[D \mapsto C]}} \quad \text{provided } D \notin \mathit{sig}(C)$$

Definer purification:

$$\frac{\mathcal{T}}{\mathcal{T}^{[D \mapsto \top]}} \quad \text{provided } D \text{ occurs in } \mathcal{T} \text{ only under number restrictions}$$

Cyclic definer elimination:

$$\frac{\mathcal{T} \cup \{D \sqsubseteq C[D]\}}{\mathcal{T}^{[D \mapsto \nu X.C[X]]}} \quad \text{provided } D \in \mathit{sig}(C[D])$$

Fig. 6. Rules for eliminating definer symbols

left-hand side of its definition, we can replace all positive occurrences of it using the non-cyclic definer elimination rule. If there is no definition of D, D occurs only positively and we can replace all its occurrences by $\top$. If a definer symbol occurs on both sides of its definition, applying the non-cyclic definer elimination rule would lead to an infinite derivation. Instead we apply the cyclic-definer elimination rule, which introduces a greatest fixpoint operator.

Since ontologies can have cyclic definitions, it is in general not always possible to find a finite uniform interpolant that does not use fixpoint operators. If we want to compute a representation of the uniform interpolant that is completely in $\mathcal{SHQ}$ and does not use fixpoint operators, we can either keep the cyclic definer symbols, or approximate the uniform interpolant. Keeping the cyclic definer symbols has the advantage that we preserve all entailments of the uniform interpolant and obtain an ontology that can be processed by any common reasoner supporting $\mathcal{SHQ}$. The remaining definer symbols can be seen as "helper concepts" that make a finite representation possible.

If we want an ontology without fixpoints that is completely in the desired signature, we can in general only approximate the uniform interpolant, since it might be infinite. This approximation can be performed by replacing the cyclic definer symbols a finite number of times following their definitions, and then replacing them by $\top$ (see also [10] for this).

Theorem 3. *Given any ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R} \rangle$ and concept symbol A, $\mathcal{O}^{-A} = \langle \mathcal{T}^{-A}, \mathcal{R} \rangle$ is a uniform interpolant of $\mathcal{O}$ for $\mathit{sig}(\mathcal{O}) \setminus \{A\}$ in $\mathcal{SHQ}\mu$. If O^{-A} does not use any fixpoint operators, it is the uniform interpolant of $\mathcal{O}$ in $\mathcal{SHQ}$ for $\mathit{sig}(\mathcal{O}) \setminus \{A\}$.*

We conducted a small evaluation of forgetting concept symbols from real-life ontologies, details of which can be found in the long version of this paper [11]. Our results suggest that at least for smaller ontologies of up to 700 axioms, forgetting half of the concept symbols in the signature can be performed in a few minutes in the majority of cases.

Role hierarchy:	**$\leq$-Monotonicity:**	**$\geq$-Monotonicity:**
$\dfrac{s \sqsubseteq r \quad r \sqsubseteq t}{s \sqsubseteq t}$	$\dfrac{C \sqcup \leq nr.\neg D \qquad s \sqsubseteq r}{C \sqcup \leq ns.\neg D}$	$\dfrac{C \sqcup \geq nr.D \qquad r \sqsubseteq s}{C \sqcup \geq ns.D}$

Fig. 7. Inference rules for forgetting the role symbol r.

6 Forgetting Role Symbols

We can adapt the method from [8] to obtain a procedure for forgetting role symbols from $\mathcal{SHQ}$ ontologies, provided the role to be forgotten is not transitive. For forgetting role symbols, we have to process the RBox as well, and act differently depending on what we can make of the role hierarchy.

Forgetting transitive roles is not possible if we want to express the uniform interpolant in $\mathcal{SHQ}\mu$, as the following theorem shows.

Theorem 4. *There are ontologies $\mathcal{O}$ and role symbols r, where r is transitive in $\mathcal{O}$, without a finite uniform interpolant of $\mathcal{O}$ for $sig(\mathcal{O}) \setminus \{r\}$ in $\mathcal{SHQ}\mu$.*

Proof. Consider an ontology $\mathcal{O}$ with an RBox $\mathcal{R} = \{s \sqsubseteq r, r \sqsubseteq t, \mathsf{trans}(r)\}$. We have the following infinite number of entailments of $\mathcal{O}$, where C is any concept: $\mathcal{O} \models \forall t.C \sqsubseteq \forall s.C, \forall t.C \sqsubseteq \forall s.\forall s.C, \ldots$. Since neither t nor s are transitive, there can be no finite ontology in $\mathcal{SHQ}\mu$ defined over $sig(\mathcal{O}) \setminus \{r\}$ and entails all these consequences.

In order to forget a non-transitive role symbol r, we have to apply the combination rules on number restrictions with r exhaustively. As for forgetting concept symbols, the other rules have to be applied only if they lead to the introduction of new definer symbols and clauses that make further derivations on r possible. On the resulting clause set, we apply the monotonicity rules shown in Figure 7. We denote the result by $\mathbf{N}^{*r}$.

If r has a super-role s (that is, $r \sqsubseteq s$), we can afterwards filter out all clauses that contain r and obtain a clausal representation of the uniform interpolant. The $\geq$-monotonicity role ensures that all information regarding $\geq$-restrictions in the desired signature is preserved. If r has no super-role, we have to check which clauses of the form $C \sqcup \geq nr.D$ can be filtered out. If $\mathbf{N}^{*r} \models \neg D$, we replace $C \sqcup \geq nr.D$ with C, otherwise, we can remove $C \sqcup \geq nr.D$ from the clause set, since all derivations on r have already been performed. To decide whether $\mathbf{N}^{*r} \models \neg D$, we can either use an external reasoner or the calculus $Res^{s}_{\mathcal{SHQ}}$ presented in Section 4. We also remove all clauses that are of the form $\neg D \sqcup D$ or $\neg D_1 \sqcup \neg D_2 \sqcup C$. The resulting set $\mathbf{N}^{-r}$ of clauses is transformed into an $\mathcal{SHQ}\mu$- or $\mathcal{SHQ}$-ontology using the definer elimination techniques described in the previous section. The resulting TBox $\mathcal{T}^{-r}$ is the TBox of the uniform interpolant. The RBox $\mathcal{R}^{-r}$ is computed by applying the role hierarchy rule from Figure 7 on the RBox and filtering out role inclusions containing r. We have the following result.

Theorem 5. *Given any ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R} \rangle$ and any role symbol r such that $\mathsf{trans}(r) \notin \mathcal{R}$, $\mathcal{O}^{-r} = \langle \mathcal{T}^{-r}, \mathcal{R}^{-r} \rangle$ is a uniform interpolant of $\mathcal{O}$ for $sig(\mathcal{O}) \setminus \{r\}$ in $\mathcal{SHQ}\mu$. If $\mathcal{O}^{-r}$ does not contain any fixpoint operators, it is a uniform interpolant of $\mathcal{O}$ for $sig(\mathcal{O}) \setminus \{r\}$ in $\mathcal{SHQ}$.*

7 Discussion and Related Work

There has been research in developing methods that deal with uniform interpolation in several description logics, starting from simple ones, such as DL-Lite [20] and $\mathcal{EL}$ [7, 15, 13], to more expressive ones such as $\mathcal{ALC}$ and $\mathcal{ALCH}$ [19, 12, 10, 9, 8]. Forgetting in more expressive description logics was first investigated by [18] and [14]. In [14], it was shown that deciding the existence of uniform interpolants that can be represented finitely in $\mathcal{ALC}$ without fixpoints is 2-ExpTime and that these uniform interpolants can in the worst case have a size triple exponential with respect to size of the original ontology. It can be shown that, using fixpoint operators, this bound can be reduced to a double-exponential complexity. [18, 19] were first to consider the computation of uniform interpolants in $\mathcal{ALC}$ and presented a tableau-based approach. A more goal-oriented method was presented in [12], following a resolution approach based on a different calculus than our method. [12] also included first experimental results, showing the practicality for a lot of applications.

The first resolution-based method incorporating fixpoints, using ideas from the area of second-order quantifier elimination [3], was presented in [10]. This method was implemented and evaluated on a large set of real-life ontologies [9], showing that the worst case complexity of uniform interpolants is hardly reached in reality, and that uniform interpolants can often be computed in a few seconds. In [8], this method was extended by redundancy elimination techniques and the ability to forget role symbols, and an evaluation showed even better results with respect to the size and use of fixpoint operators in the result.

Forgetting for description logics with number restrictions was first considered in [20], where a method for the description logic DL-Lite$^{\mathcal{N}}_{bool}$ is presented. DL-Lite$^{\mathcal{N}}_{bool}$ extends DL-Lite with unqualified number restrictions and Boolean operators [1]. The logic allows for inverse roles, but does not allow concepts under number restrictions. More specifically, it cannot express universal restrictions or qualified existential role restrictions. This makes it possible to implement forgetting in DL-Lite$^{\mathcal{N}}_{bool}$ using propositional resolution.

Apart from number restrictions, there are more extensions to $\mathcal{SHQ}$ that have not been investigated yet, such as inverse roles or nominals. Whereas it is possible that a method for computing uniform interpolants in an expressive description logic with inverse roles will be discovered in the future, for nominals this will likely not be the case. This follows from result for module extraction from [5]. Given two ontologies $\mathcal{O}$ and $\mathcal{M}$ and a signature $\mathcal{S}$, $\mathcal{M}$ is an *$\mathcal{S}$-module* of $\mathcal{O}$, if $\mathcal{M}$ is a subset of $\mathcal{O}$ and has the same logical entailments over $\mathcal{S}$ as $\mathcal{O}$. Different from uniform interpolants, modules can contain symbols that are not in $\mathcal{S}$. Uniform interpolation can be used to test whether an ontology $\mathcal{M}$ is an $\mathcal{S}$-module of

another ontology $\mathcal{O}$. More specifically, $\mathcal{M}$ is an $\mathcal{S}$-module of $\mathcal{O}$ iff $\mathcal{M} \subseteq \mathcal{O}$ and $\mathcal{M} \models \mathcal{O}^{\mathcal{S}}$, where $\mathcal{O}^{\mathcal{S}}$ is the uniform interpolant of $\mathcal{O}$ over $\mathcal{S}$. But in [5] it was shown that determining whether $\mathcal{M}$ is an $\mathcal{S}$-module of $\mathcal{O}$ is undecidable already for the description logic $\mathcal{ALCO}$, which extends $\mathcal{ALC}$ with nominals. From this follows that there can be no general method for computing uniform interpolants of $\mathcal{ALCO}$-ontologies that are represented in a decidable logic.

8 Conclusion and Future Work

We have presented a method for uniform interpolation of $\mathcal{SHQ}$-ontologies. The method allows to compute uniform interpolants for ontologies in more expressive description logics than previous approaches, and also to preserve indirect cardinality information from ontologies that do not use number restrictions. The method makes use of a new sound and refutationally complete resolution-calculus for the description logic $\mathcal{SHQ}$. If a finite representation cannot be computed in pure $\mathcal{SHQ}$, fixpoint operators are used in the result, which can be simulated using helper concepts. The result of forgetting transitive roles cannot be represented in $\mathcal{SHQ}\mu$, and is not computed by our method. A solution might be to use a description logic that allows for transitive closures of roles.

Results of a preliminary evaluation of our method indicate that at least for smaller ontologies, forgetting of concept symbols can be performed in short amounts of time [11]. Even for input ontologies that do not contain number restrictions, we sometimes obtained interesting results where further entailments using $\geq$-number restrictions where derived, that would not have been part of an $\mathcal{ALCH}$ uniform interpolant. It is likely that for smaller signatures (100–2000 symbols, depending on the ontology), uniform interpolants of large ontologies can in most cases still be computed in short times, if only concept symbols have to be forgotten, and there are optimisations we have not investigated yet. Forgetting role symbols is likely a much more expensive task, since our combination rules derive a lot of consequences. An evaluation of this is future work.

We have not investigated the complexity of our method and uniform interpolation in $\mathcal{SHQ}$ in general. It is therefore open whether our method is optimal. Additionally, we are currently investigating uniform interpolation for description logics with inverse roles, such as $\mathcal{SHI}$ and $\mathcal{SHIQ}$. Besides extending the expressivity of the supported description logic, there are further ways in which the framework of uniform interpolation can be extended. So far, most work on uniform interpolation for more expressive description logics focused on the TBox and the RBox. An open problem is uniform interpolation of ontologies in $\mathcal{SHQ}$ or more expressive logics that also have an ABox, which would be useful in many applications, including privacy and ontology analysis.

References

1. Artale, A., Calvanese, D., Kontchakov, R., Zakharyaschev, M.: DL-Lite in the Light of First-Order Logic. In: AAAI 2007. pp. 361–366. AAAI Press (2007)

2. Calvanese, D., Giacomo, G.D., Lenzerini, M.: Reasoning in Expressive Description Logics with Fixpoints based on Automata on Infinite Trees. In: Proc. IJCAI '99. pp. 84–89. Morgan Kaufmann (1999)
3. Gabbay, D.M., Schmidt, R.A., Szałas, A.: Second Order Quantifier Elimination: Foundations, Computational Aspects and Applications. College Publ. (2008)
4. Grau, B.C.: Privacy in ontology-based information systems: A pending matter. Semantic Web 1 (1–2), 137–141 (2010)
5. Grau, B.C., Horrocks, I., Kazakov, Y., Sattler, U.: Modular reuse of ontologies: theory and practice. J. Artif. Intell. Res. 31 (1), 273–318 (2008)
6. Horrocks, I., Sattler, U., Tobies, S.: Practical Reasoning for Very Expressive Description Logics. Logic J. IGPL 8 (3), 239—264 (2000)
7. Konev, B., Walther, D., Wolter, F.: Forgetting and Uniform Interpolation in Large-Scale Description Logic Terminologies. In: Proc. IJCAI '09. pp. 830–835. AAAI Press (2009)
8. Koopmann, P., Schmidt, R.A.: Forgetting Concept and Role Symbols in $\mathcal{ALCH}$-Ontologies. In: Proc. LPAR'13. LNCS, vol. 8312, pp. 552–567. Springer (2013)
9. Koopmann, P., Schmidt, R.A.: Implementation and Evaluation of Forgetting In $\mathcal{ALC}$-Ontologies. In: Proc. WoMO'13. CEUR-WS.org (2013)
10. Koopmann, P., Schmidt, R.A.: Uniform Interpolation of $\mathcal{ALC}$-Ontologies Using Fixpoints. In: Proc. FroCoS'13. LNCS, vol. 8152, pp. 87–102. Springer (2013)
11. Koopmann, P., Schmidt, R.A.: Count and Forget: Uniform Interpolation of $\mathcal{SHQ}$-Ontologies—Long Version. Tech. Rep., The University of Manchester (2014). `http://www.cs.man.ac.uk/~koopmanp/IJCAR_KoopmannSchmidt2014_long.pdf`
12. Ludwig, M., Konev, B.: Towards Practical Uniform Interpolation and Forgetting for $\mathcal{ALC}$ TBoxes. In: Proc. DL'13. pp. 377–389. CEUR-WS.org (2013)
13. Lutz, C., Seylan, I., Wolter, F.: An Automata-Theoretic Approach to Uniform Interpolation and Approximation in the Description Logic $\mathcal{EL}$. In: Proc. KR'12. pp. 286–296. AAAI Press (2012)
14. Lutz, C., Wolter, F.: Foundations for Uniform Interpolation and Forgetting in Expressive Description Logics. In: Proc. IJCAI '11. pp. 989–995. AAAI Press (2011)
15. Nikitina, N.: Forgetting in General $\mathcal{EL}$ Terminologies. In: Proc. DL'11. pp. 345–355. CEUR-WS.org (2011)
16. Schmidt, R.A., Hustadt, U.: A principle for incorporating axioms into the first-order translation of modal formulae. In: Automated Deduction–CADE-19, LNAI, vol. 2741, pp. 412–426. Springer (2003)
17. Tobies, S.: Complexity results and practical algorithms for logics in knowledge representation. Ph.D. thesis, RWTH-Aachen, Germany (2001)
18. Wang, K., Wang, Z., Topor, R., Pan, J., Antoniou, G.: Concept and Role Forgetting in $\mathcal{ALC}$ Ontologies. In: Proc. ISWC'09. LNCS, vol. 5823, pp. 87–102. Springer (2009)
19. Wang, K., Wang, Z., Topor, R., Pan, J.Z., Antoniou, G.: Eliminating concepts and roles from ontologies in expressive descriptive logics. Comput. Intell., to appear
20. Wang, Z., Wang, K., Topor, R.W., Pan, J.Z.: Forgetting for knowledge bases in DL-Lite. Ann. Math. Artif. Intell. 58 (1–2), 117–151 (2010)