Count and Forget: Uniform Interpolation of $\mathcal{SHQ}$-Ontologies—Long Version*

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

* The final publication of the short version is available at Springer.

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

applications where too many symbols in the ontology that users are unfamiliar with could be harmful [21]. **(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 [5]. 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 [8]. **(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 [8, 14].

So far, the only expressive description logics for which methods for computing uniform interpolants exists are $\mathcal{ALC}$ and $\mathcal{ALCH}$ [11, 9, 12, 20]. In this paper we extend the methods of [11, 9] 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 [11, 9, 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 [11, 9]. 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 appendix.

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 [7], 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 [3]. 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 [18, 3].[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] [3] 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 [17].

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 $\leq 0r.\neg D$ 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 $\leq nr.\neg D$, 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 [17] 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, $\geq(n_1+n_2)r.(\mathcal{D}_1 \sqcup \mathcal{D}_2) \sqcup \geq 1r.\mathcal{D}_{12}$ is equivalent to $\leq(n_1+n_2-1)r.(\mathcal{D}_1 \sqcup \mathcal{D}_2) \rightarrow \geq 1r.\mathcal{D}_{12}$. 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:

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

where D' is a new definer symbol.

$\geq$-Combination:

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

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:

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

$\geq$-Resolution:

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

$\geq$-Elimination:

$$\frac{C \sqcup {\geq}nr.\bot}{C}$$

Fig. 1. Inference rules of $\mathit{Res}_{\mathcal{SHQ}}$.

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. *$\mathit{Res}_{\mathcal{SHQ}}$ 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 $\mathit{Res}_{\mathcal{SHQ}}$ 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, $\mathit{Res}_{\mathcal{SHQ}}$ 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 [9],

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 $\mathit{Res}^s_{\mathcal{SHQ}}$.

≤-Combination:

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

≤≥-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 ≤-restrictions are negated, and because $(\neg D_1 \sqcup \neg D_2) \equiv \neg(D_1 \sqcap D_2)$, we can verify that the ≤-combination rule is sound.

Fig. 5. Diagram illustrating the ≤≥-combination rule.

For the ≤≥-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 $\leq n_1 r.\neg D_1$ 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 ≤≥-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 ≤≥-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 appendix.

$\mathbf{N}^{-A}$ 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:

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

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:

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

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 [11] 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 $sig(\mathcal{O}) \setminus \{A\}$ in $\mathcal{SHQ}\mu$. If $\mathcal{O}^{-A}$ does not use any fixpoint operators, it is the uniform interpolant of $\mathcal{O}$ in $\mathcal{SHQ}$ for $sig(\mathcal{O}) \setminus \{A\}$.*

We conducted a small evaluation of forgetting concept symbols from real-life ontologies, details of which can be found in the appendix. 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:	≤-Monotonicity:	≥-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 [9] 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 [21] and $\mathcal{EL}$ [8, 15, 13], to more expressive ones such as $\mathcal{ALC}$ and $\mathcal{ALCH}$ [20, 12, 11, 10, 9]. Forgetting in more expressive description logics was first investigated by [19] 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. [19, 20] 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 [4], was presented in [11]. This method was implemented and evaluated on a large set of real-life ontologies [10], 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 [9], 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 [21], 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 [2]. 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 [6]. 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 [6] 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 (see appendix). 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. Ackermann, W.: Untersuchungen über das Eliminationsproblem der mathematischen Logik. Mathematische Annalen 110(1), 390–413 (1935)

2. 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)
3. 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)
4. Gabbay, D.M., Schmidt, R., Szalas, A.: Second Order Quantifier Elimination: Foundations, Computational Aspects and Applications. College Publ. (2008)
5. Grau, B.C.: Privacy in ontology-based information systems: A pending matter. Semantic Web 1(1-2), 137–141 (2010)
6. Grau, B.C., Horrocks, I., Kazakov, Y., Sattler, U.: Modular reuse of ontologies: theory and practice. J. Artif. Int. Res. 31(1), 273–318 (2008)
7. Horrocks, I., Sattler, U., Tobies, S.: Practical Reasoning for Very Expressive Description Logics. J. of the IGPL 8(3), 239—264 (2000)
8. 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)
9. 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)
10. Koopmann, P., Schmidt, R.A.: Implementation and evaluation of forgetting in $\mathcal{ALC}$-ontologies. In: Proc. WoMO'13. CEUR-WS.org (2013)
11. 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)
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. Nonnengart, A., Szałas, A.: A fixpoint approach to second-order quantifier elimination with applications to correspondence theory. In: Logic at Work, pp. 307–328. Springer (1999)
17. Schmidt, R.A., Hustadt, U.: A principle for incorporating axioms into the first-order translation of modal formulae. In: Automated Deduction–CADE-19, pp. 412–426. Springer (2003)
18. Tobies, S.: Complexity results and practical algorithms for logics in knowledge representation. Ph.D. thesis, LuFG Theoretical Computer Science, RWTH-Aachen, Germany (2001)
19. Wang, K., Wang, Z., Topor, R., Pan, J., Antoniou, G.: Concept and role forgetting in $\mathcal{ALC}$ ontologies pp. 666–681 (2009)
20. Wang, K., Wang, Z., Topor, R., Pan, J.Z., Antoniou, G.: Eliminating concepts and roles from ontologies in expressive descriptive logics. Computational Intelligence (2012)
21. 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)

A Examples

A.1 The Bicycle Example

Let us compute the uniform interpolant for the example given in the introduction. Assume we have an TBox $\mathcal{T}_{bike}$ of the following form:

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

The $\mathcal{SHQ}$-normal form $\mathbf{N}$ of this TBox is the following:

1. $\neg\mathsf{Bicycle} \sqcup \geq 1 \mathsf{hasWheel}.D_1$
2. $\neg D_1 \sqcup \mathsf{FrontWheel}$
3. $\neg\mathsf{Bicycle} \sqcup \geq 1 \mathsf{hasWheel}.D_2$
4. $\neg D_2 \sqcup \mathsf{RearWheel}$
5. $\neg\mathsf{FrontWheel} \sqcup \mathsf{Wheel}$
6. $\neg\mathsf{FrontWheel} \sqcup \neg\mathsf{RearWheel}$
7. $\neg\mathsf{RearWheel} \sqcup \mathsf{Wheel}$

We want to compute a uniform interpolant for $\mathcal{S} = \{\mathsf{Bicycle}, \mathsf{Wheel}, \mathsf{hasWheel}\}$, and therefore to forget the concept symbols FrontWheel and RearWheel. First resolution is applied on FrontWheel.

8. $\neg D_1 \sqcup \mathsf{Wheel}$ *(Resolution 2, 5)*
9. $\neg D_1 \sqcup \neg\mathsf{RearWheel}$ *(Resolution 2, 6)*

The combination rules do not make further inferences on FrontWheel possible. All clauses containing the concept symbol FrontWheel (2, 5, and 6) can therefore be directly filtered out. Eliminating D_1 and D_2 and transforming the result back into clausal form results in the same set of clauses, which is why we can directly proceed to forgetting RearWheel. First resolution is aplied on RearWheel in the remaining clauses. Observe that since clauses of the form $\neg D_1 \sqcup \neg D_2 \sqcup C$ can be discarded, they do not have to be derived. There is only one resolvent that contains less than two negative definer literals.

10. $\neg D_2 \sqcup \mathsf{Wheel}$ *(Resolution 4, 7)*

Applying the $\geq$-combination rule leads to new clauses on which we can apply resolution on RearWheel.

~~11.~~ $\neg\mathsf{Bicycle} \sqcup {\geq}2\mathsf{hasWheel}(D_1 \sqcup D_2)$
$\sqcup\, {\geq}1\mathsf{hasWheel}.D_{12}$ *($\geq$-Combination 1, 3)*
~~12.~~ $\neg D_{12} \sqcup D_1$
~~13.~~ $\neg D_{12} \sqcup D_2$
~~14.~~ $\neg D_{12} \sqcup \mathsf{Wheel}$ *(Resolution 12, 8)*
~~15.~~ $\neg D_{12} \sqcup \neg\mathsf{RearWheel}$ *(Resolution 12, 9)*
~~16.~~ $\neg D_{12} \sqcup \mathsf{RearWheel}$ *(Resolution 13, 4)*
17. $\neg D_{12}$ *(Resolution 15, 16)*

Clause 17 makes clauses 12–16 become redundant. Clause 11 becomes redundant if the $\geq$-resolution is applied with the newly derived unit clause $\neg D_{12}$.

~~18.~~ $\neg\mathsf{Bicycle} \sqcup {\geq}2\mathsf{hasWheel}(D_1 \sqcup D_2)$
$\sqcup\, {\geq}1\mathsf{hasWheel}.\bot$ *($\geq$-Resolution 11, 17)*
19. $\neg\mathsf{Bicycle} \sqcup {\geq}2\mathsf{hasWheel}(D_1 \sqcup D_2)$ *($\geq$-Elimination on 18)*

Clause 18 makes Clause 11 become redundant, and Clause 19 makes Clause 18 become redundant. After removing all clauses that are redundant, of the form $\neg D_a \sqcup D_b$, or contain the concept symbol RearWheel, $\mathbf{N}^{\mathcal{S}}$ contains the Clauses 1, 3, 8, 10, 17 and 19 (the boldly numbered clauses). For $\mathcal{T}_D^{\mathcal{S}}$ replace all clauses containing a negative definer literal by the definitions $D_1 \sqsubseteq \mathsf{Wheel}$, $D_2 \sqsubseteq \mathsf{Wheel}$ and $D_3 \sqsubseteq \bot$. By applying acyclic definer elimination on all definer symbols, the following set of axioms for the uniform interpolant $\mathcal{T}_{bike}^{\mathcal{S}}$ is obtained.

$$\top \sqsubseteq \neg\mathsf{Bicycle} \sqcup {\geq}2\mathsf{hasWheel}.(\mathsf{Wheel} \sqcup \mathsf{Wheel})$$
$$\top \sqsubseteq \neg\mathsf{Bicycle} \sqcup {\geq}1\mathsf{hasWheel}.\mathsf{Wheel}$$

The last axiom is redundant, and the first one can be reformulated to the following axiom.

$$\mathsf{Bicycle} \sqsubseteq {\geq}2\mathsf{hasWheel}.\mathsf{Wheel}$$

This is the uniform interpolant $\mathcal{T}_{bike}^{\Sigma}$ of $\mathcal{T}_{bike}$ for $\Sigma = \{\mathsf{Bicycle}, \mathsf{hasWheel}, \mathsf{Wheel}\}$, as claimed in Section 1.

A.2 Example with $\leq$-Restrictions

Consider the following TBox $\mathcal{T}_1$.

$$A_1 \sqsubseteq {\geq}5r.(A \sqcup B) \qquad A_2 \sqsubseteq {\leq}3r.A$$

The $\mathcal{SHQ}$-clausal form $\mathbf{N}_1$ of $\mathcal{T}_1$ 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. $\neg D_2 \sqcup \neg A$.

Suppose we want to forget A. The $\geq\leq$-combination rule can be applied on Clauses 1 and 3.

5. $\neg A_1 \sqcup \neg A_2 \sqcup \geq 2r.D_{12}$	*($\geq\leq$-Combination 1, 3)*
6. $\neg D_{12} \sqcup D_1$	*($D_{12} \sqsubseteq D_1$)*
7. $\neg D_{12} \sqcup D_2$	*($D_{12} \sqsubseteq D_2$)*
$\not 8$. $\neg D_{12} \sqcup A \sqcup B$	*(Resolution 2, 6)*
9. $\neg D_{12} \sqcup \neg A$	*(Resolution 7, 4)*
10. $\neg D_{12} \sqcup B$	*(Resolution 8, 9)*

$\mathbf{N}_1^{-A}$ consists of the bold-numbered clauses. Applying the definer elimination rules results in the following set of axioms.

$$\bot \sqsubseteq \neg A_1 \sqcup \geq 5r.\top \qquad \bot \sqsubseteq \neg A_2 \sqcup \leq 3r.\neg\top$$
$$\bot \sqsubseteq \neg A_1 \sqcup \neg A_2 \sqcup \geq 2r.B$$

The second axiom is tautological, since $\leq 3r.\neg\top$ is always satisfied. The uniform interpolant of $\mathcal{T}_1^{-A}$ can therefore be represented in the following way.

$$A_1 \sqsubseteq \geq 5r.\top \qquad A_1 \sqcap A_2 \sqsubseteq \geq 2r.B$$

A.3 Example with Fixpoints

If the TBox is cyclic, it is possible that there is no finite uniform interpolant in $\mathcal{SHQ}$, and we have to express it in $\mathcal{SHQ}\mu$ or with the help of additional concepts.

Consider the following TBox $\mathcal{T}_3$.

1. $A \sqsubseteq B$ 2. $B \sqsubseteq \geq 3r.(\neg C)$ 3. $B \sqsubseteq \exists r.(B \sqcup C)$

Due to the third axiom, $\mathcal{T}$ is cyclic. Suppose we want to compute the uniform interpolant $\mathcal{T}_3^{\mathcal{S}}$ for $\mathcal{S} = \{A, r\}$. The $\mathcal{SHQ}$-clausal form $\mathbf{N}_3$ of $\mathcal{T}_3$ is the following.

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

First we forget C.

6. $\neg B \sqcup \geq 4r.(D_1 \sqcup D_2) \sqcup \geq 1r.D_{12}$ *(≥-combination 2, 4)*
7. $\neg D_{12} \sqcup D_1$
8. $\neg D_{12} \sqcup D_2$
9. $\neg D_{12} \sqcup \neg C$ *(Resolution 3, 7)*
10. $\neg D_{12} \sqcup B \sqcup C$ *(Resolution 5, 8)*
11. $\neg D_{12} \sqcup B$ *(Resolution 9, 10)*

$\mathbf{N}_3^{-C}$ contains the clauses 1, 2, 4, 7 and 12 (clauses containing C, positive definer literals or more than one negative definer literal are ignored). Next B is forgotten.

12. $\neg A \sqcup \geq 3r.D_1$ *(Resolution 1, 2)*
13. $\neg A \sqcup \geq 1r.D_2$ *(Resolution 1, 4)*
14. $\neg A \sqcup \geq 4r.(D_1 \sqcup D_2) \sqcup \geq 1r.D_{12}$ *(Resolution 1, 6)*
15. $\neg D_{12} \sqcup \geq 3r.D_1$ *(Resolution 2, 11)*
16. $\neg D_{12} \sqcup \geq 1r.D_2$ *(Resolution 4, 11)*
17. $\neg D_{12} \sqcup \geq 4r.(D_1 \sqcup D_2) \sqcup \geq 1r.D_{12}$ *(Resolution 6, 11)*

Clauses 12–17 form the clausal representation of the uniform interpolant $\mathcal{T}_3^{\mathcal{S}}$. Clauses 15–17 can be represented the following way as concept inclusion.

$$D_{12} \sqsubseteq \geq 3r.D_1 \sqcap \geq 1r.D_2 \sqcap (\geq 4r.(D_1 \sqcup D_2) \sqcup \geq 1r.D_{12})$$

Since D_1 and D_2 occur only positively, definer purification can be applied, resulting in the following TBox $\mathcal{T}_3'$.

$$\begin{aligned} \top &\sqsubseteq \neg A \sqcup \geq 3r.\top \\ \top &\sqsubseteq \neg A \sqcup \geq 4r.\top \sqcup \geq 1r.D_{12} \\ D_{12} &\sqsubseteq \geq 3r.\top \sqcap (\geq 4r.\top \sqcup \geq 1r.D_{12}) \end{aligned}$$

The definition of D_{12} is cyclic. Applying cyclic definer elimination leads to the following TBox.

$$\begin{aligned} \top &\sqsubseteq \neg A \sqcup \geq 3r.\top \\ \top &\sqsubseteq \neg A \sqcup \geq 4r.\top \sqcup \geq 1r.\nu X.(\geq 3r.\top \sqcap (\geq 4r.\top \sqcup \geq 1r.X)) \end{aligned}$$

By distributing the conjunction, exploiting the equivalence $\geq 4r.\top \sqcap \geq 3r.\top \equiv \geq 4r.\top$ and moving the fixpoint operator outside, we can simplify the result and obtain the following uniform interpolant $\mathcal{T}_3^{\mathcal{S}}$.

$$A \sqsubseteq \nu X.(\geq 4r.\top \sqcup (\geq 3r.\top \sqcap \exists r.X))$$

There is no finite $\mathcal{SHQ}$ axiom without fixpoint operators that is equivalent to this axiom. Therefore, there is no finite $\mathcal{SHQ}$ ontology which is the uniform interpolant of $\mathcal{T}_3$ for $\mathcal{S}$. However, $\mathcal{T}_3^S$ can be represented without fixpoints in two ways. First by omitting the cyclic definer elimination rule, that is by taking the TBox $\mathcal{T}_3'$ above. $\mathcal{T}_3'$ contains an additional symbol D_{12}, which can be seen as a helper concept to enable a finite representation. $\mathcal{T}_3'$ does not contain the symbols B and C, and has the same entailments in $\mathcal{S}$ as the uniform interpolant. Moreover $\mathcal{T}_3'$ can be represented in OWL and processed by modern description logic reasoners without fixpoint abilities.

The other possibility is to approximate the uniform interpolant. This can be done by applying the non-cyclic definer elimination rule on D_{12} for a finite number of times (each time this adds new occurrences of D_{12}), and then replace D_{12} by $\top$. For example, by proceeding this way, after two iterations we obtain the following ontology that approximates $\mathcal{T}_3^{\mathcal{S}}$ in $\mathcal{SHQ}$:

$$A \sqsubseteq {\geq}4r.\top \sqcup ({\geq}3r.\top \sqcap \exists r.({\geq}4r.\top \sqcup ({\geq}3r.\top \sqcap \exists r.({\geq}3r.\top)))).$$

A.4 Transitive Roles

To illustrate the function of the transitivity rule, the ontology in this example has both a TBox and an RBox. Consider the following ontology $\mathcal{O}$:

$$A \sqsubseteq \forall s.(A \sqcup B) \qquad A \sqsubseteq \exists r.\exists r.\neg B \qquad r \sqsubseteq s \qquad \mathsf{trans}(r) \tag{1}$$

We want to compute $\mathcal{O}^{-B}$, or to forget B. The $\mathcal{SHQ}$-clausal form of $\mathcal{O}$ is the following:

1. $\neg A \sqcup {\leq}0s.\neg D_1$
2. $\neg D_1 \sqcup A \sqcup B$
3. $\neg A \sqcup {\geq}1r.D_2$
4. $\neg D_2 \sqcup {\geq}1r.D_3$
5. $\neg D_3 \sqcup \neg B$

We also list the RBox axioms, as they are still taken into account by the rules.

6. $r \sqsubseteq s$
7. $\mathsf{trans}(r)$

The transitivity rule is applied on the Clause 1 and the two RBox axioms of the ontology $\mathcal{O}$.

8. $\neg A \sqcup {\leq}0r.\neg D_1'$ *(Transitivity 1,6,7)*
9. $\neg D_1' \sqcup D_1$ *(Transitivity 1,6,7)*
10. $\neg D_1' \sqcup {\leq}0r.\neg D_1'$ *(Transitivity 1,6,7)*
11. $\neg D_1' \sqcup A \sqcup B$ *(Resolution 2, 9)*

In order to obtain clauses which enable to infer resolvents on B with at most one negative definer literal, there are several applications of the $\geq\leq$-combination rule than can be performed.

12. $\neg A \sqcup {\geq}1r.D'_{12}$ *($\geq\leq$-combination 3, 8)*
13. $\neg D'_{12} \sqcup D'_1$
14. $\neg D'_{12} \sqcup D_2$
15. $\neg D'_{12} \sqcup {\geq}1r.D_3$ *(Resolution 4, 13)*
16. $\neg D'_{12} \sqcup D_1$ *(Resolution 9, 13)*
17. $\neg D'_{12} \sqcup {\leq}0r.\neg D'_1$ *(Resolution 10, 13)*
18. $\neg D'_{12} \sqcup A \sqcup B$ *(Resolution 11, 13)*
19. $\neg D'_{12} \sqcup {\geq}1r.\neg D'_{13}$ *($\geq\leq$-combination 15, 17)*
20. $\neg D'_{13} \sqcup D'_1$
21. $\neg D'_{13} \sqcup D_3$
22. $\neg D'_{13} \sqcup A \sqcup B$ *(Resolution 11, 20)*
23. $\neg D'_{13} \sqcup \neg B$ *(Resolution 5, 21)*
24. $\neg D'_{13} \sqcup A$ *(Resolution 22, 23)*

All clauses necessary for the uniform interpolant are already derived. There are more rule applications possible, but to shorten the presentation, we stop here. (All further inference steps just produce redundant information.) The boldly numbered clauses are in the clausal form of the uniform interpolant. We obtain the following definitions:

$$\begin{aligned} D_2 &\sqsubseteq {\geq}1r.D_3 \\ D'_1 &\sqsubseteq {\leq}0r.\neg D'_1 \\ D'_{12} &\sqsubseteq {\geq}1r.D_3 \sqcap {\leq}0r.\neg D'_1 \sqcap {\geq}1r.D'_{13} \\ D'_{13} &\sqsubseteq A \end{aligned}$$

In general, we do not have to include the direct results of the transitivity rule into the uniform interpolant, which is why the definition of D' can be omitted. In this case however, the definition of D'_1 can be used to illustrate a situation where cyclic definers do not lead to fixpoint operators in the result. By applying cyclic definer elimination for D'_1, occurrences of D'_1 are replaced by $\nu X.{\leq}0r.\neg X$. But the greatest fixpoint of ${\leq}0r.\neg X$ is $\top$. ($\top$ is a fixpoint since ${\leq}0r.\neg\top \equiv \top$, and it is the greatest fixpoint since it always covers the whole domain.) Therefore, D' can simply be replaced by $\top$. (Tautological fixpoints like this are actually detected in our prototypical implementation, see also [10]). All other definers are eliminated using the non-cyclic definer elimination rule. We obtain the following uniform interpolant $\mathcal{O}^{-B}$.

$$A \sqsubseteq \exists r.\exists r.A \qquad r \sqsubseteq s \qquad \mathsf{trans}(r)$$

B Proofs of Theorems

B.1 Completeness of the Underlying Calculus

For any set $\mathbf{N}$ of $\mathcal{SHQ}$-clauses, we denote by $\mathbf{N}^*$ tbe saturation of $\mathbf{N}$ using the rules of $Res_{\mathcal{SHQ}}$ with respect to some given RBox $\mathcal{R}$. To prove the completeness of the calculus, we show that either $\mathbf{N}^*$ contains the empty clause, or we can build a model for $\mathcal{O} = \langle \mathbf{N}, \mathcal{R} \rangle$ based on the clauses in $\mathbf{N}^*$. To this intent a constructive proof approach is followed. Based on the clauses in $\mathbf{N}^*$, we first build for each definer symbol a segment of the model. Then these segments are connected to a complete model of $\mathcal{O}$. This approach is very similar to what is done in [11].

The construction is only based on clauses with at most one negative definer literal. Since $\mathbf{N}$ does not contain any clause with more than one negative definer literal and $\mathbf{N} \subseteq \mathbf{N}^*$, any model that satisfies all clauses in $\mathbf{N}^*$, except those with more than one negative definer literal, will also be a model of $\mathbf{N}$. For this reason, in the following we modify the definition of $\mathbf{N}^*$ so that $\mathbf{N}^*$ only contains inferences with at most one negative definer literal. (It is for this reason that inferences of clauses with more than one negative definer literal can be ignored, as done in the forgetting procedure.)

We start by extending the ordering $\sqsubseteq_d$ defined in Definition 3, Section 4, to a strict total ordering between definer symbols $\prec_d$. A definer D is *maximal* in a disjunction $\mathcal{D}$ of definers if D is maximal in $\mathcal{D}$ according to $\prec_d$. $\prec_d$ is extended to a strict total ordering between disjunctions of definers based on the multiset extension $(\prec_d)_{mul}$.[2] We further define a strict total ordering $\prec_l$ on literals that satisfies the following constraints, where disjunctions of definer symbols are treated as sets.

1. $D \prec_l \neg D \prec_l A \prec_l \neg A \prec_l {\geq} nr.\mathcal{D} \prec_l {\leq} nr'.\neg\mathcal{D}'$ for all $A \in N_c$, $r, r' \in N_r$, $D \in N_d$ and $\mathcal{D} \in 2^{N_d}$.
2. ${\geq} n_1 r_1.\mathcal{D}_1 \prec_l {\geq} n_2 r_2.\mathcal{D}_2$ if $\mathcal{D}_1 \prec_d \mathcal{D}_2$ for all $\mathcal{D}_1, \mathcal{D}_2 \in 2^{N_d}$, $r_1, r_2 \in N_r$ and $n_1, n_2 > 0$.

Since $\prec_d$ is already a strict total ordering, a strict total ordering satisfying Condition 2 can easily be constructed. An ordering satisfying the first condition can be created by assuming a total ordering between concept symbols, role symbols and numbers.

A literal L is *maximal* in a clause C if we have $L' \prec_l L$ for all $L' \in C$. Observe that since clauses are represented as sets and $\prec_l$ is strict and total, the maximal literal of a clause is always unique.

$\prec_l$ is extended to an ordering $\prec_c$ on clauses using the multiset extension $(\prec_l)_{mul}$ of $\prec_l$. This means $C_1 \prec_c C_2$ iff for every literal $L_1 \in C_1$, $L_1 \prec_l L_2$ for some literal $L_2 \in C_2$.

Define the set N_d^+ to contain all definers used in $\mathbf{N}^*$ and the special symbol ϵ. $\mathbf{N}^*$ is partitioned into a set of *definition sets* by a function $d : N_d^+ \longrightarrow 2^{\mathbf{N}^*}$. d

[2] Given a strict ordering $\prec$, the multiset extension $(\prec)_{mul}$ is defined as $A(\prec)_{mul}B$ iff for every $a \in A$ there is a $b \in B$ such that $a \prec b$.

maps each definer $D \in N_d^+$ to the subset of clauses in $\mathbf{N}^*$ that have $\neg D$ as the only negative definer literal, and maps ϵ to the set of clauses that do not contain any negative definer literals. $d(D)$ contains the clauses that make up the *definition* of D, in the sense that they can be represented by an axiom of the form $D \sqsubseteq ...$, hence we use the terminology "definition set" for $d(D)$. $d(\epsilon)$ contains all the remaining clauses which are not related to the definition of any definer. Let $d^e(D) = d(D) \cup d(\epsilon)$ be the definition set extended with these clauses. The ordering $\prec_c$ provides an enumeration of the clauses in each $d^e(D)$. In the following, C_i^D denotes the ith clause in $d^e(D)$ according to $\prec_c$, starting from the smallest clause C_1^D.

Following this enumeration, we define a model segment I^D for each definer $D \in N_d^+$ that satisfies every clause in $d^e(D)$. These model segment are parts of the model rooted in a single domain element, that only refer to its direct successors. A model segment consists of a set of concept symbols the corresponding domain element satisfies, and role connections to other definer symbols. Due to the $\geq$-number restrictions, it might be necessary to connect a domain element to more than one successor satisfying the same definer. For this reason, model segments are represented using multisets.

Definition 4. Additive literals *are of the form* A *or* $+1r.D$*, where* $A \in N_c$*,* $r \in N_r$*,* $D \in N_d$*.* A model segment *is a multiset of additive literals. A model segment* I satisfies a literal L*, written* $I \models L$*, iff*

1. *L is a positive literal of the form A and $A \in I$,*
2. *L is a negative literal of the form $\neg A$ and $A \notin I$,*
3. *L is of the form $\geq nr.\mathcal{D}$ and $\#\{+1r.D' \in I \mid D' \sqsubseteq_d \mathcal{D}\} \geq n$.*
4. *L is of the form $\leq nr.\neg\mathcal{D}$ and $\#\{+1r.D' \in I \mid D' \not\sqsubseteq_d \mathcal{D}\} \leq n$.*

A model segment I satisfies a clause C*, written* $I \models C$*, if there is a literal* $L \in C$ *such that* $I \models L$*.*

I^D is defined formally in six steps:

1. If $\neg D \in d(D)$, set $I^D = \emptyset$. Otherwise, let
2. $I_0^D = \{D\}$ if $D \neq \epsilon$ and $I_0^D = \emptyset$ otherwise.
3. $I_i^D = I_{i-1}^D \cup \{L\}$, if $I_{i-1}^D \not\models C_i^D$ and the maximal literal L of C_i^D is an unnegated concept symbol.
4. $I_i^D = I_{i-1}^D \cup \{(+1s.D')^n \mid r \sqsubseteq_{\mathcal{R}} s\}$, if $I_{i-1}^D \not\models C_i^D$ and the maximal literal of C_i^D is of the form $\geq n'r.\mathcal{D}$, where D' is the largest disjunct in $\mathcal{D}$ and n is the smallest number such that $I_{i-1}^D \cup \{(+1s.D')^n \mid r \sqsubseteq_{\mathcal{R}} s\} \models C_i^D$,
5. $I_i^D = I_{i-1}^D$ otherwise.
6. $I^D = I_n^D$, where n is the number of clauses in $d^e(D)$.

A step in the model construction is called *effective* if the step ensures that $I_i^D \models C_i$ and $I^D \models C_i$. It is easy to verify that Steps 3 and 4 of the model construction are effective. Following Step 4, the further properties of the created model segments can be proven.

Lemma 2. I^D *satisfies the following properties.*

1. *For every* $+1r.D' \in I^D$ *and* s *with* $r \sqsubseteq_{\mathcal{R}} s$, $+1s.D' \in I^D$.
2. *If* $I^D \models {\geq}nr.\mathcal{D}$*, then also* $I^D \models {\geq}ns.\mathcal{D}$ *for all* s *with* $r \sqsubseteq_{\mathcal{R}} s$.
3. *If* $I^D \models {\leq}nr.\neg\mathcal{D}$*, then also* $I^D \models {\leq}ns.\neg\mathcal{D}$ *for all* s *with* $s \sqsubseteq_{\mathcal{R}} r$.
4. *If* $I^D_{i-1} \not\models C^D_i$ *and the maximal literal of* C^D_i *is* ${\geq}nr.\mathcal{D}$*, the cardinality of* $\{+1r.D \in I^D_{i+1} \mid \{D\} \sqsubseteq_d \mathcal{D}\}$ *is* n.

Proof. Lemma 2.1 holds since, if $+1r.D'$ is added in Step 4, for every $r \sqsubseteq s$, $+1s.D'$ is added as well. The other properties are step-wise consequences of Lemma 2.1.

Before we prove that $I^D \models C$ holds for every model segment I^D and every clause $C \in \mathbf{N}^*$ in case $\perp \notin \mathbf{N}^*$, we consider a few special cases. First we prove that conclusions of the $\geq$-combination rule do not lead to further additive literals in the model segments.

Lemma 3. *Let* $\mathbf{N}^*$ *contain two clauses of the form* $C^D_{j_i} = C_i \sqcup {\geq}n_i r_i.\mathcal{D}_i$*, where* $D \in N_d^+$ *and* $i \in \{1,2\}$*, such that* ${\geq}n_i r_i.\mathcal{D}_i$ *is maximal in* $C^D_{j_i}$*, and such that there is a role* r *with* $r_i \sqsubseteq_{\mathcal{R}} r$*. Let* C^D_j *be a conclusion of applying the* $\geq$*-combination rule on* $C^D_{j_1}$ *and* $C^D_{j_2}$*. Then,* $I^D_{j-1} \models C^D_j$.

Proof. First observe that $C_{j_i} \prec_c C^D_j$ for both $i \in \{1,2\}$, since the maximal literal in C^D_j is ${\geq}nr.(\mathcal{D}_1 \sqcup \mathcal{D}_2)$, and $\mathcal{D}_i \sqsubseteq_d \mathcal{D}_1 \sqcup \mathcal{D}_2$ for $i \in \{1,2\}$. For the indices, this implies $j_i < j$ for each $i \in \{1,2\}$. Due to Lemmata 2.4 and 2.2, we therefore have $I^D_{j-1} \models {\geq}n_1 r.\mathcal{D}_1$ and $I^D_{j-1} \models {\geq}n_2 r.\mathcal{D}_2$. Denote the sets of additive literals in I^D_{j-1} that make this true by $I^D_{+\mathcal{D}_i}$: $I^D_{+\mathcal{D}_i} = \{+1r.D' \in I^D_j \mid \{D'\} \sqsubseteq \mathcal{D}_i\}$, $i \in \{1,2\}$. By definition, (i) $I^D_{+\mathcal{D}_i} \models {\geq}n_i r.\mathcal{D}_i$ and (ii) $I^D_{+\mathcal{D}_i} \subseteq I^D_{j-1}$, and due to (i) also (iii) $\#I^D_{+\mathcal{D}_i} \geq n_i$ and (iv) $\{D'\} \sqsubseteq \mathcal{D}_i$ for every D' with $+1r.D' \in I^D_{+\mathcal{D}_i}$.

If $I^D_{+\mathcal{D}_1} \cap I^D_{+\mathcal{D}_2} = \emptyset$, due to (iii) we have $\#(I^D_{+\mathcal{D}_1} \cup I^D_{+\mathcal{D}_2}) \geq (n_1 + n_2)$. Together with (ii) and (iv), from this follows $I^D_{j-1} \models {\geq}(n_1 + n_2)r.(\mathcal{D}_1 \sqcup \mathcal{D}_2)$ and also $I^D_{j-1} \models C_j$. Assume $I^D_{+\mathcal{D}_1} \cap I^D_{+\mathcal{D}_2} \neq \emptyset$. Then let $I^D_{\sqcap} = I^D_{+\mathcal{D}_1} \cap I^D_{+\mathcal{D}_2}$. Due to (iv), we then have $\{D'\} \sqsubseteq_d \mathcal{D}_1$ and $\{D'\} \sqsubseteq_d \mathcal{D}_2$ for all D' with $+1r.D' \in I^D_{\sqcap}$. If $\mathcal{D}_{12}$ represents the conjunction of $\mathcal{D}_1$ and $\mathcal{D}_2$ (see definition of the $\geq$-rule), this implies $\{D'\} \sqsubseteq_d \mathcal{D}_{12}$ for every D' with $+1r.D' \in I^D_{\sqcap}$. Consequently, $I^D_{\sqcap} \models {\geq}n_{\sqcap} r.\mathcal{D}_{12}$, where $n_{\sqcap} = \#I^D_{\sqcap}$. With (ii), this gives us (v) $I^D_{j-1} \models {\geq}n_{\sqcap} r.\mathcal{D}_{12}$ and (vi) $I^D_{j-1} \models {\geq}(n_1 + n_2 - n_{\sqcap})r.(\mathcal{D}_1 \sqcup \mathcal{D}_2)$. C^D_j is a conclusion of the $\geq$-rule, therefore it contains the literal $L_1 = {\geq}n' r.(\mathcal{D}_1 \sqcup \mathcal{D}_2)$, where $n' \leq (n_1 + n_2)$, and the literal $L_2 = {\geq}n'' r.\mathcal{D}_{12}$, where $n'' = (n_1 + n_2 + 1 - n')$. If $n' \leq (n_1 + n_2 - n_{\sqcap})$, $I^D_{j-1} \models L_1$ due to (vi). If $n' > (n_1 + n_2 - n_{\sqcap})$, $n'' < n_1 + n_2 + 1 - (n_1 + n_2 - n_{\sqcap}) = 1 + n_{\sqcap}$. Consequently $n_{\sqcap} \geq n''$ and due (v), $I^D_{j-1} \models L_2$. Hence, $I^D_{j-1} \models C^D_j$. □

Next we show that clauses with a $\leq$-restriction as maximal literal, which are not treated by the model construction, are satisfied by the model segments as well. Since this proof is a bit more complicated, we start with a simplified situation.

Lemma 4. *Assume we have a satisfiable set of clauses* $\boldsymbol{N} = \{\leq nr.\neg D_a\} \cup \{\geq n_i r_i.D_i \mid 1 \leq i \leq m, r_i \sqsubseteq_{\mathcal{R}} r\} \cup \boldsymbol{M}$*, where* $\boldsymbol{M}$ *only consists of clauses with at least one negative definer literal. If* $\boldsymbol{N}^*$ *is the saturation of* $\boldsymbol{N}$ *and* I^ϵ *the model segment for* ϵ *generated from* $\boldsymbol{N}^*$*, then* $I^\epsilon \models \leq nr.\neg D_a$*.*

Proof. We have to show that $\#\{+1r_i.D_i \in I^\epsilon \mid r_i \sqsubseteq r, D_i \not\sqsubseteq D_a\} \leq n$. If $\Sigma_{i=1}^m n_i < n$, this statement holds since in Step 4 of the model construction, maximally $\Sigma_{i=1}^m n_i$ additive literals are added to I^ϵ for the clauses $\mathbf{N}$, and no further additive literals are added for clauses that can be derived from $\mathbf{N}$ (Lemma 3).

Otherwise set $n^\Delta = \Sigma_{i=1}^m n_i - n$. The proof is by induction, where the base case is that $n^\Delta \leq n_i$ for all $i \leq m$.

Base case: Assume $n^\Delta < n_i$ for all $\geq n_i r_i.D_i \in \mathbf{N}$. Applying the $\geq$-rule incrementally on each clause in $\mathbf{N}$ yields a set of clauses of the form $\geq n^\Sigma r^\Sigma.\mathcal{D}^\Sigma \sqcup C^\Sigma$, where $n^\Sigma = \Sigma_{i=1}^m n_i$, $r^\Sigma \sqsubseteq r$, $\mathcal{D} = \{D_i \mid 0 < i \leq m\}$. C^Σ consists of number restrictions of the form $\geq n' r'.\mathcal{D}'$, $n' \leq n_i, i \leq m$, where $\mathcal{D}'$ contains definers representing conjunctions of definers in $\mathbf{N}$.

Applying the $\geq\leq$-rule with on clause of this form and $\leq nr.\neg D_a$ yields clauses of the form $C^\Delta = \geq n^\Delta r^\Sigma.\mathcal{D}^\Delta \sqcup C^\Sigma$, where $\mathcal{D}^\Delta = \{D_{ia} \mid 1 \leq i \leq m\}$ consists of definers D_{ia} representing $D_i \sqcap D_a$ for $1 \leq i \leq m$ (definition of $\geq\leq$-rule). Every number restriction in C^Δ contains definers that represent conjunctions of definers in $\mathbf{N}$. Therefore, C^Δ is smaller than the clauses $\geq n_i r_i.D_i \in \mathbf{N}$ and processed before these clauses by the construction of I^ϵ. Due to Lemma 3, we can also ignore saturations of C^Δ via the $\geq$-combination rule.

For these clauses C^Δ, in Step 4 of the construction of I^ϵ, we add n^Δ additive literals of the form $+1r.D_{ij}$, where each D_{ij} subsumes at least two definer symbols from $\mathbf{N}$. Since C^Δ is smaller than the clauses in $\mathbf{N}$, for each of these additive literals where $D_{ij} \not\sqsubseteq D_a$, 2 additive definer literals less are added when the $\geq$-restrictions in $\mathbf{N}$ are processed. For each of these additive literals where $D_{ij} \sqsubseteq D_a$, 1 additive definer literal less is added when $\geq$-restrictions in $\mathbf{N}$ are processed. As a result, only $n^\Sigma - n^\Delta = n$ definer literals $+1r'.D'$ with $D' \not\sqsubseteq D_a$ are added, and hence $\#\{+1r_i.D_i \in I^\epsilon \mid r_i \sqsubseteq r, D_i \not\sqsubseteq D_a\} = n$ and $I^\epsilon \models \leq nr.\neg D_a$.

Induction Step: Let $\mathbf{N}_1$ be a clause set for which the inductive hypothesis holds, and $\geq n_2 r.D_2$ be a clause not in $\mathbf{N}_1$ with $D' \prec_d D_2$ for all definers occurring in $\mathbf{N}_1$. We show that the hypothesis holds for $\mathbf{N} = \mathbf{N}_1 \cup \{\geq n_2 r.D_2\}$.

Let $\mathbf{N}_1^*$ be the saturation of $\mathbf{N}_1$ using $Res_{\mathcal{SHQ}}$. As in the base case, there is a clause $C_1^\Sigma = \geq n^\Sigma r.\mathcal{D}^\Sigma \sqcup \ldots \in \mathbf{N}_1^*$ which is the result of incrementally applying the $\geq$-combination rule on all $\geq$-restrictions in $\mathbf{N}_1$ and a clause $C_1^\Delta = \geq n^\Delta r.\mathcal{D}^\Delta \sqcup \ldots \in \mathbf{N}_1^*$ which is the result of applying the $\geq\leq$-combination rule on this clause and $\leq nr.\neg D_a$. Applying the $\geq$-combination rule on C_1^Σ and $\geq n_2 r.D_2$ results in a new clause $C_2^\Sigma = \geq(n^\Sigma + n_2)r.(\mathcal{D} \sqcup D_2) \sqcup \ldots$, and applying the $\geq\leq$-rule on this clause results in a clause $C_2^\Delta = \geq(n^\Delta + n_2)r.(\mathcal{D}^\Delta \sqcup D_{2a}) \sqcup \ldots$. Since $D' \prec_d D_2$ for all definer symbols in $\mathbf{N}_1$, we also have that D_{2a} is larger than all definer symbols in $\mathcal{D}^\Delta$, which represent conjunctions of definers from $\mathbf{N}_1$. Therefore, $C_1^\Delta \prec_c C_2^\Delta$. If $\neg D_{2a} \in \mathbf{N}^*$, $\mathbf{N}^*$ would further contain conclusions of the $\geq$-resolution rule with $\neg D_{2a}$.

(i) Assume $\neg D_{2a} \notin \mathbf{N}^*$, or that the $\geq$-resolution rule is not applicable on D_{2a}. Since $C_1^\Delta \prec_c C_2^\Delta$, C_2^Δ is processed after C_1^Δ by the model construction, but still before $\geq n_2 r.D_2$. $\geq(n^\Delta + n_2)r.(\mathcal{D}^\Delta \sqcup D_{2a})$ is the maximal literal in C_2^Δ, and D_{2a} is larger than all definers in $\mathcal{D}^\Delta$. Therefore, in Step 4 of the model construction, n_2 additive literals of the form $+1r.D_{2a}$ are added to I^ϵ. $D_{2a} \sqsubseteq_d D_2$, therefore, when $\geq n_2 r.D_2$ is processed, no additive literals of the form $+1r.D_2$ are added. We also have $D_{2a} \sqsubseteq D_a$, and due to the inductive hypothesis, $\#\{+1r_i.D_i \in I^\epsilon \mid r_i \sqsubseteq r, D_i \not\sqsubseteq D_a\} = n$, and hence $I^D \models \leq nr.\neg D_a$.

(ii) If $\neg D_{2a} \in \mathbf{N}^*$, then applying the $\geq$-resolution rule on the clauses $\neg D_{2a}$ and $\geq(n^\Delta + n_2)r.(\mathcal{D}^\Delta \sqcup D_{2a}) \sqcup {C^{\Sigma'}}_a$, results in a set of smaller clauses of the form $C_a^\Delta = \geq(n^\Delta + n_2)r.(\mathcal{D}^\Delta) \sqcup {C^{\Sigma'}}_a$. But ${C^{\Sigma'}}_a$ contains a literal $\geq n_2 r.\mathcal{D}_{2i}$, where $\mathcal{D}_{2i} \sqsubseteq D_2$ and $\mathcal{D}_{2i} \sqsubseteq \mathcal{D}_i$ for at least one disjunction of definers $\mathcal{D}_i$ in $\mathbf{N}_1^*$. Since $D_j \prec_D D_2$ for all definer symbols D_j in $\mathbf{N}_1$, and since every definer $D_{jk} \neq D_{2i}$ in ${C^{\Sigma'}}_a$ represents the disjunctions over conjunctions of definers that are smaller than D_2, $\geq n_2 r.\mathcal{D}_{2i}$ is the maximal literal in C_a^Δ. As a result, n_2 literals of the form $+1r.D_{2i}$, $D_{2i} \in \mathcal{D}_{2i}$, are added in Step 4 of the model construction. Since $\geq n_2 r.\mathcal{D}_{2i}$ is maximal in C_a^Δ, all clauses in $\mathbf{N}_1^*$ where $\geq n' r'.\mathcal{D}_i$ is maximal are larger than C_a^Δ, and therefore processed later by the model construction. As a result, n_2 additive literals less are added for these, than it would be the case for the clause set $\mathbf{N}_1^*$. Also, no additive literals are added for the clause $\geq n_2 r.D_2$. By the inductive hypothesis, for $\mathbf{N}_1^*$, maximally n additive literals of the form $+1r_i.D_i, r_i \sqsubseteq r, D_i \not\sqsubseteq D_a$ are part of the model segment. For $\mathbf{N}$, we add n_2 additive literals of this form less when processing the clauses in $\mathbf{N}_1^*$, and we add n_2 additive literals of the form $+1r.\mathcal{D}_{2i}$. Therefore, we have again that $\#\{+1r_i.D_i \in I^\epsilon \mid r_i \sqsubseteq r, D_i \not\sqsubseteq D_a\} \leq n$, and hence $I^\epsilon \models \leq nr.D_a$. □

Lemma 4 can be generalised to the following lemma.

Lemma 5. *Let $\mathbf{N}^*$ be a saturated set of clauses and $C_i^D = \leq nr.D' \sqcup C_i^{D'} \in \mathbf{N}^*$, $D \in N_d^+$, be a clause in which $\leq nr.D'$ is maximal. Let I^D be the model segment for D generated from $\mathbf{N}^*$. Then, $I^D \models C_i^D$.*

Proof. Assume $I^D \not\models C_i^D$. Since $\leq nr.D' \in C_i^D$, this implies $I^D \not\models \leq nr.D'$. This can only be the case due to clauses of the form $C_{i_j}^D = \geq n_j r_j.D_j \sqcup C_{i_j}^{D'}$, $r_j \sqsubseteq r$, $D_j \not\sqsubseteq D_i$ where $\geq n_j r_j.D_j$ is maximal, and $I_{j-1}^D \not\models C_j^D$. Let $\mathbf{N}'$ be a subset of $\mathbf{N}'$ that only contains these clauses, C_i^D and any unary clause of the form $\neg D' \in \mathbf{N}^*$, and let ${\mathbf{N}'}^*$ be the saturation of $\mathbf{N}'$. Let ${I'}^D$ be the model segment for D generated from ${\mathbf{N}'}^*$. $\mathbf{N}'^*$ contains all clauses from $\mathbf{N}^*$ that are crucial for $I^D \not\models \leq nr.D'$, and therefore we have ${I'}^D \not\models \leq nr.D'$. Let $\mathbf{N}''$ be a clause set that contains a unary clause C_{max} for each clause $C \in \mathbf{N}'$, where C_{max} contains the maximal literal of C. Let ${I''}^D$ be the model segment for D generated from the saturation ${\mathbf{N}''}^*$. Since ${I'}^D \not\models \leq nr.D'$, ${I''}^D \not\models \leq nr.D'$ holds as well. On the other hand, $\mathbf{N}''$ is a clause set of the same form as in Lemma 4, which implies ${I''}^D \models \leq nr.D'$. We arrived at a contradiction, which means the initial assumption cannot be true, and therefore we have $I^D \models C_i^D$. □

We can now prove that each non-empty model segment I^D satisfies all clauses in $\mathbf{N}^*$.

Lemma 6. *For any $D \in N_d^+$, if I^D is nonempty, then $I^D \models C$ for all clauses $C \in \boldsymbol{N}^*$.*

Proof. Remember that $d^e(D)$ contains all clauses C_i^D that either contain no negative definer literal, or $\neg D$ as the negative definer literal. Assume $C \notin d^e(D)$. Then, C must be of the form $\neg D' \sqcup C'$ with $D' \neq D$. If $D' \notin I^D$, $I^D \not\models C$ holds trivially. If $D' \in I^D$, there must be a clause $\neg D \sqcup D' \in d^e(D)$, and then due to resolution on D' we have the clause $\neg D \sqcup C' \in d^e(D)$. For this reason, it is sufficient to restrict our attention to the clauses $C_i^D \in d^e(D)$.

We validate that for each $C_i^D \in d^e(D)$ we have $I^D \models C_i^D$. Observe that the conclusion C of a rule application on clauses in $d^e(D)$ will also have either no negative definer literal or $\neg D$ as the only negative definer literal, and is hence included in $d^e(D)$.

The proof is by contradiction. Assume i is the smallest i with $I^D \not\models C_i^D$.

1. The maximal literal in C_i^D is of the form A or $\geq nr.D'$. Then $I^D \models C_i^D$ since Steps 3 and 4 of the construction of I^D are effective. This contradicts the initial assumption that $I^D \not\models C_i^D$.
2. If the maximal literal in C_i^D is of the form $\neg A$, we have $I^D \not\models \neg A$ and therefore $I^D \models A$. This means there must be a smaller clause C_j^D, where A is maximal in C_j^D and $I_j^D \not\models C_j^D \setminus \{A\}$, otherwise A is not added to I^D. But then, due to the resolution rule, there is also a clause $C = (C_i^D \cup C_j^D) \setminus \{A, \neg A\}$, which is also in $d^e(D)$. Since $\prec_c$ is the multiset extension of the ordering between literals, and since $\neg A$ as the maximal literal in C_i^D is larger than all elements in C, C is smaller than C_i^D.
   Since both $I^D \not\models C_j^D \setminus \{A\}$ and $I^D \not\models C_i^D \setminus \{\neg A\}$, we have $I^D \not\models C$. Because C belongs to $d^e(D)$ and is smaller than C_i^D, there is a $k < i$ with $C = C_k^D$, which contradicts our initial assumption that i is the smallest i with $I^D \not\models C_i^D$.
3. If the maximal literal in C_i^D is of the form $\leq nr.\neg D_1$, due to Lemma 5 $I^D \models C_i^D$, which contradicts the initial assumption that $I^D \not\models C_i^D$. □

Based on the models segments, the candidate model $\mathcal{I}_{\mathbf{N}^*} = \langle \Delta^{\mathcal{I}_{\mathbf{N}^*}}, \cdot^{\mathcal{I}_{\mathbf{N}^*}} \rangle$, is constructed as follows. For every definer D with $\neg D \notin \mathbf{N}^*$, $\Delta^{\mathcal{I}_{\mathbf{N}^*}}$ contains n_{max} domain elements x_i^D, where n_{max} is the largest number occurring in a role restriction in $\mathbf{N}^*$. Additionally, $\Delta^{\mathcal{I}_{\mathbf{N}^*}}$ contains the individual x^ϵ.

We first define the interpretation function $\cdot^{\mathcal{I}_{\mathbf{N}^*}}$ for concept symbols, based on the concept symbols in the model segments.

– For every $A \in N_c$, $A^{\mathcal{I}_{\mathbf{N}^*}} = \{x_i^D \mid A \in I^D, i \leq n_{max}\}$

The model segments I^D serve as guide to assign concepts to the corresponding individuals x_i^D, or to x^ϵ if $D = \epsilon$. If $A \in I^D$, we have $x_i^D \in A^{\mathcal{I}_{\mathbf{N}^*}}$.

The interpretation of role names is based on a base-interpretation function $r_0^{\mathcal{I}_{\mathbf{N}^*}}$, which is defined as follows.

– For every $r \in N_r$, $r_0^{\mathcal{I}_{\mathbf{N}^*}} = \{(x_i^{D_1}, x_j^{D_2}) \mid i \leq n_{max}, j \leq \#\{+1s.D_2 \in I^D, s \sqsubseteq_{\mathcal{R}} r\}, I^{D_2} \neq \emptyset\}$

For each $+r.D_2 \in I^{D_1}$, an r-relation between each $x_i^{D_1}$ and one domain element $x_j^{D_2}$ is added, if these domain elements exist. $r_0^{\mathcal{I}_{\mathbf{N}^*}}$ is extended to take into account the role axioms in $\mathcal{R}$. Given any binary relation r_0, let $(r_0)^*$ denote the transitive closure of r_0. The interpretation function $\cdot^{\mathcal{I}_{\mathbf{N}^*}}$ is then defined as follows.

– For every $r \in N_r$, $r^{\mathcal{I}_{\mathbf{N}^*}} = r_0^{\mathcal{I}_{\mathbf{N}^*}} \cup \{(x, x') \in (s_0^{\mathcal{I}_{\mathbf{N}^*}})^* \mid s \sqsubseteq_{\mathcal{R}} r, \mathsf{trans}(s) \in \mathcal{R}\}$

There are two cases where the model differs from the model segments. First, additive literals $+r.D$ are only taken into account if there is an individual $x_i^D \in \Delta^{\mathcal{I}_{\mathbf{N}}^*}$ for the definer, that is, if $\neg D \notin \mathbf{N}^*$. Second, if a role r is transitive, additional r-relations to the ones specified in the model segments are added. To prove that $\mathcal{I}_{\mathbf{N}^*}$ is a model of $\mathbf{N}^*$, we have to show that these differences do not affect the satisfaction of the clauses in $\mathbf{N}^*$.

Lemma 7. *If $\bot \notin \boldsymbol{N}^*$, then $\mathcal{I}_{\boldsymbol{N}^*}$ is a model of $\mathcal{O} = \langle \boldsymbol{N}, \mathcal{R} \rangle$.*

Proof. Assume $\bot \notin \mathbf{N}^*$. We first prove that all RBox axioms $\alpha \in \mathcal{R}$ are satisfied by the model. For every $\mathsf{trans}(r) \in \mathcal{R}$, $\mathcal{I}_{\mathbf{N}^*} \models \mathsf{trans}(r)$ holds since we add the transitive closure of any role r with $\mathsf{trans}(r) \in \mathcal{R}$ to $\mathcal{I}_{\mathbf{N}^*}$. This construction step, together with Lemma 2.1, also ensures that $\mathcal{I}_{\mathbf{N}^*} \models r \sqsubseteq s$ for each $r \sqsubseteq s \in \mathcal{R}$.

It remains to show that the clauses are satisfied by the model as well, that is, that $\mathcal{I}_{\mathbf{N}^*} \models C$ for all $C \in \mathbf{N}^*$. The proof is done by contradiction. Assume C_i^D is the smallest clause in $\mathbf{N}^*$ with $\mathcal{I}_{\mathbf{N}^*} \not\models C_i^D$. We check the different cases for the maximal literal in C_i^D.

1. The maximal literal in C_i^D is of the form A. Then $\mathcal{I}_{\mathbf{N}^*} \models C_i^D$ follows immediately from Lemma 6, which contradicts our initial assumption.
2. The maximal literal in C_i^D is of the form $\geq nr.\mathcal{D}$. Since $I^D \models \geq nr.\mathcal{D}$, we have $\#\{+1s.D_j \in I^D, s \sqsubseteq_{\mathcal{R}} r, \{D_j\} \sqsubseteq \mathcal{D}\} \geq n$.
   (i) Assume $I^{D_j} \neq \emptyset$ for each $D_j \in \{D_j \mid +1s.D_j \in I^D, s \sqsubseteq_{\mathcal{R}} r,\ D_j \sqsubseteq \mathcal{D}\}$. Then there is an r-relation from every x_i^D to at least n domain elements. Note that if $\{D_j\} \sqsubseteq_d \mathcal{D}$, either $D_j \in \mathcal{D}$ or $\neg D_j \sqcup D'_j \in \mathbf{N}^*$ for one $D'_j \in \mathcal{D}$. In both cases, we have $D'_j \in I^{D_j}$ for some $D'_j \in \mathcal{D}$ and $x_k^{D_j} \in \mathcal{D}^{\mathcal{I}_{\mathbf{N}^*}}$. This implies $x_i^D \in (\geq nr.\mathcal{D})^{\mathcal{I}_{\mathbf{N}^*}}$, which contradicts our initial assumption.
   (ii) Assume $I^{D_2} = \emptyset$ for some $D_2 \in \{D_j \mid +1s.D_j \in I^D, s \sqsubseteq_{\mathcal{R}} r,\ D_j \sqsubseteq \mathcal{D}\}$. Then there is a clause $\neg D_2 \in \mathbf{N}^*$. Additive literals $+1s.D_2 \in I^D$ are only added in Step 4 in the construction of I^D if there is a clause C_j^D with a maximal literal $\geq ms.\mathcal{D}_2$, where D_2 is the maximal definer symbol in $\mathcal{D}_2$, and for which $I_{j-1}^D \not\models C_j$ holds. Applying the $\geq$-resolution rule on $\neg D_2$ and C_j^D produces the smaller clause $C_k^D = ((C_j^D \setminus \{\geq ms.\mathcal{D}_2\}) \cup \{\geq ms.(\mathcal{D}_2 \setminus \{D_2\})\})$. $I_{j-1}^D \not\models C_j^D$ implies $I_{j-1}^D \not\models C_k^D$. Since the maximal literal in C_j^D is a $\geq$-restriction and due to the ordering, C_k^D does not contain any $\leq$-restrictions.

Further no literal in C_k^D can be maximal in any clause larger than C_{j-1}^D. This means no step in the construction of I^D can make C_k^D satisfied. But due to Lemma 6, $I^D \models C$ for all clauses $C \in \mathbf{N}^*$. We arrived at a contradiction, which means our initial assumption $I^{D_2} = \emptyset$ must be wrong.

3. The maximal literal in C_i^D is of the form $\leq nr.\neg D_2$.
(i) Assume r is simple, that is $\mathsf{trans}(r) \notin \mathcal{R}$ and there is no role $s \sqsubseteq_{\mathcal{R}} r$ with $\mathsf{trans}(s) \in \mathcal{R}$. Then due to Lemmas 5 and 6, C_i^D is satisfied by the model.
(ii) Assume r is not simple, that is $\mathsf{trans}(r) \in \mathcal{R}$ or there is a role s with $s \sqsubseteq_{\mathcal{R}} r$ and $\mathsf{trans}(s) \in \mathcal{R}$. Recall that by the definition of $\mathcal{SHQ}$, only simple roles are allowed in number restrictions of the form $\leq n'r.D'$ with $n' > 0$. Therefore, since r is not simple, $n = 0$, and that the maximal literal in C_i^D is of the form $\leq 0r.\neg D_2$. Since $\mathcal{I}_{\mathbf{N}^*} \not\models \leq 0r.\neg D_2$, there must be a domain element $x_i^{D''}$ in the range of $r^{\mathcal{I}_{\mathbf{N}^*}}$ with $x_i^{D''} \notin D_2^{\mathcal{I}_{\mathbf{N}^*}}$.
Since there is a role s with $s \sqsubseteq_{\mathcal{R}} r$ and $\mathsf{trans}(s) \in \mathcal{R}$, the transitivity role applies, resulting in the three clauses $C_i^{D'} \sqcup \leq 0s.\neg D_2'$, $\neg D_2' \sqcup D_2$ and $\neg D_2' \sqcup \leq 0s.\neg D_2'$. Between $\neg D_2' \sqcup \leq 0s.\neg D_2'$ and any clause of the form $C' \sqcup \geq n's'.\neg D_3$, $s' \sqsubseteq s$, the $\geq\leq$-combination rule applies, resulting in the smaller clause $\neg D' \sqcup C' \sqcup \geq n's'.D_{23}'$, where $D_{23} \sqsubseteq D_2'$. Therefore, for any definer D' which occurs in an additive literal $+1s'.D' \in I^{D_2'}$, $s' \sqsubseteq s$, we will have the clause $\neg D' \sqcup D_2'$, and we will have that $D_2' \in I^{D'}$ and $x_x^{D'} \in D_2'^{\mathcal{I}_{\mathbf{N}^*}}$. Resolution on D_2' gives the clause $\neg D' \sqcup \leq 0s.\neg D_2'$, so that the situation for additive literals in $I^{D'}$ is the same. As a result, we have for every domain element x' that is in the domain of the transitive closure of $s^{\mathcal{I}_{\mathbf{N}^*}}$, that $x' \in D_2'^{\mathcal{I}_{\mathbf{N}^*}}$. The same holds for any transitive role s' with $s' \sqsubseteq r$. By assumption, there is a domain element $x_i^{D''}$ in the range of $r^{\mathcal{I}_{\mathbf{N}^*}}$ with $x_i^{D''} \notin D_2$. But then, since $x_i^{D''} \in D_2'^{\mathcal{I}_{\mathbf{N}^*}}$, there must be a clause $\neg D'' \sqcup D_2'$. Resolution with $\neg D_2' \sqcup D_2$ results in the clause $\neg D'' \sqcup D_2$. Since $x'' \in D''^{\mathcal{I}_{\mathbf{N}^*}}$ and $x'' \notin D_2^{\mathcal{I}_{\mathbf{N}^*}}$, we have $\mathcal{I}_{\mathbf{N}^*} \not\models \neg D'' \sqcup D_2$. But $\neg D'' \sqcup D_2$ is smaller than C_i^D, which contradicts the initial assumption that C_i^D is the smallest clause with $\mathcal{I}_{\mathbf{N}^*} \not\models C_i^D$. □

Lemma 7 allows us to establish the first theorem.

Theorem 1. *$Res_{\mathcal{SHQ}}$ is sound and refutationally complete.*

To prove the soundness and refutational completeness of $Res^s_{\mathcal{SHQ}}$, that is $Res_{\mathcal{SHQ}}$ with redundancy elimination techniques, observe that the orderings $\prec_l$ and $\prec_D$ are are compatible with the subsumption relations $\sqsubseteq_l$ and $\sqsubseteq_d$.

Lemma 8. *$Res^s_{\mathcal{SHQ}}$ is sound and refutationally complete.*

Proof. We first prove soundness. The tautology deletion and the subsumption deletion rule are sound because they only remove clauses. The reduction rule is sound since $A \sqsubseteq B$ implies $A \sqcup B \sqsubseteq B$.

Refutational completeness is proven by referring to the candidate model construction in Lemma 6.

Tautology deletion: Assume $C = C' \sqcup A \sqcup \neg A$. Since both $A \in C$ and $\neg A \in C$, for any literal set I^D we have $I^D \models_D C$, therefore the presence of C does not affect the candidate model construction, and it can as well be removed from the clause set.

Subsumption deletion: According to the orderings used to prove Lemma 6, $L_1 \sqsubseteq_l L_2$ implies either $L_1 = L_2$ or $L_1 \prec_l L_2$. Hence, $C_1 \sqsubseteq_c C_2$ implies either $C_1 = C_2$ or $C_1 \prec_c C_2$. Assume we have $C_1 \sqsubseteq_c C_2$ and $I^D \models C_1$ for any model segment I^D. Since $C_1 \sqsubseteq C_2$, for every literal $L \in C_1$, we have either $L \in C_2$ or there is a literal $L_2 \in C_2$ with $L_1 \sqsubseteq_l L_2$. $\sqsubseteq_l$ is defined in such a way that $I^D \models L_1$ implies $I^D \models L_2$ (see Definition 3 and 4). Therefore, $C_1 \sqsubseteq C_2$ and $I^D \models C_1$ imply $I^D \models C_2$. C_1 is processed before C_2 by the model construction, and therefore C_2 does not affect the construction of I^D, and the completeness is preserved if we eagerly apply subsumption deletion.

Reduction: Since $red(C) \sqsubseteq_c C$ and subsumption deletion preserves refutational completeness, reduction preserves refutational completeness as well. □

We now prove that adding redundancy elimination to $Res^s_{\mathcal{SHQ}}$ indeed establishes termination of the calculus.

Lemma 9. *Given a finite set of clauses* $\boldsymbol{N}$*, the saturation of* $\boldsymbol{N}^*$ *using* $Res^s_{\mathcal{SHQ}}$ *is finitely bounded.*

Proof. The $\geq$-combination rule is the only rule that increases the numbers occurring in the number restrictions in its conclusion. We already established that maximally 2^k definer symbols are introduced by the calculus, where k is the number of definer symbols occurring in the initial set of clauses (see end of Section 4). Since the normal form does not allow for nested disjunctions under number restrictions, this also gives a finite bound on the number of definer disjunctions occurring in the clause set. (Remember that both clauses and disjunctions of definer symbols are represented as sets. Therefore, they cannot have duplicate elements.) Assume the $\geq$-combination rule can be applied on two clauses $C_1 \sqcup {\geq} n_1 r_1.\mathcal{D}_1$ and $C_2 \sqcup {\geq} n_2 r_2.\mathcal{D}_2$. There are two possibilities: (i) $\mathcal{D}_1 \neq \mathcal{D}_2$ or $r_1 \neq r_2$, and (ii) $\mathcal{D}_1 = \mathcal{D}_2$ and $r_1 = r_2$. There are only finitely many possibilities of the first case, since both the number of possible definer disjunctions is bounded and the number of role symbols in the RBox is bounded, therefore we can restrict our attention to the second case. If $\mathcal{D}_1 = \mathcal{D}_2$ and $r_1 = r_2$, our conclusions are of the following form:

$$C_1 \sqcup C_2 \sqcup {\geq}(n_1 + n_2) r_2.\mathcal{D}_2 \sqcup {\geq} 1 r_2.\mathcal{D}_2$$
$$\vdots$$
$$C_1 \sqcup C_2 \sqcup {\geq}(n_1 + 1) r_2.\mathcal{D}_2 \sqcup {\geq} n_2 r_2.\mathcal{D}_2$$

These clauses are are all redundant with respect to the premise $C_2 \sqcup {\geq} n_2 r_2.\mathcal{D}_2$. Hence, the $\geq$-combination rule can only produce new clauses up to a certain bound, if subsumption deletion is applied eagerly. All other rules infer clauses where the numbers occurring in the premise do not increase, and all other symbols that can occur in a clause are limited. Hence, $\mathbf{N}^*$ is finitely bounded. □

Lemma 8 and Lemma 9 allow to establish the following theorem.

Theorem 2. *$Res^s_{\mathcal{SHQ}}$ is sound and refutationally complete, and provides a decision procedure for $\mathcal{SHQ}$.*

B.2 Forgetting Concept Symbols

We can now use the results from the last section to prove the correctness of our procedure for forgetting concept symbols. First we recall and introduce some further definitions.

Definition 5. *A definer symbol D_1 is* directly connected *to a symbol D_2 if there is a clause $\neg D_1 \sqcup C \sqcup \geq nr.\mathcal{D}$ or $\neg D_1 \sqcup C \sqcup \leq nr.\neg\mathcal{D}$ and $D_2 \in \mathcal{D}$. A definer symbol D_1 is connected to another definer symbol D_2 if D_1 and D_2 are directly connected, or if D_1 is directly connected to a definer D_3 that is connected to D_2.*

A definer symbol D is connected to a symbol A, if the negative definer $\neg D$ occurs in a clause together with the literal A or $\neg A$, or if it is connected to a definer D that is connected to A. A clause C is connected to a symbol A if it contains A or a definer that is connected to A under a number restriction.

Example 3. Assume **N** consists of the following clauses.

1. $\neg D_1 \sqcup A \sqcup \geq 3r.D_2$
2. $\neg D_2 \sqcup \geq 1r.D_3$
3. $\neg D_3 \sqcup \leq 2r.\neg D_4 \sqcup \neg E$
4. $\neg D_1 \sqcup \leq 5s.\neg(D_1 \sqcup D_5)$
5. $\neg D_5 \sqcup \geq 9t.D_6$
6. $\neg D_5 \sqcup B \sqcup C$

Clauses 1 and 4 are connected to A, B, C and E, Clauses 2 and 3 are connected to E, and Clauses 5 and 6 are connected to B and C.

Definition 6. *Given a definer D, clauses of the form $\neg D \sqcup C$ are called* defining clauses *for D.*

In order to prove Lemma 1, we have to show that for every concept inclusion α with $A \notin sig(\alpha)$, $\mathbf{N}^{-A} \models \alpha$ iff $\mathbf{N} \models \alpha$. For this we make use of a modified version of the calculus $Res_{\mathcal{SHQ}}$, by reducing entailment to clause set satisfiability. More precicely, for any clause set **N** and any concept inclusion $\alpha = C \sqsubseteq D$, $\mathbf{N} \models \alpha$ holds iff $\mathbf{N} \cup \mathbf{M}$ is unsatisfiable, where **M** is the clausal representation of the concept inclusion $\top \sqsubseteq \exists u.(C \sqcap \neg D)$, and u is a new role not occuring in **N**. Since $Res_{\mathcal{SHQ}}$ is sound and refutationally complete, it can be used to decide satisfiability of clause sets. But for our proof, we will make use of a further restricted version of $Res_{\mathcal{SHQ}}$, which we denote by $Res^{\prec}_{\mathcal{SHQ}}$. In $Res^{\prec}_{\mathcal{SHQ}}$, the resolution rule is only applied on maximal literals in each clause. $Res^{\prec}_{\mathcal{SHQ}}$ is still sound and refutationally complete, as the following theorem shows.

Theorem 6. *$Res^{\prec}_{\mathcal{SHQ}}$ is sound and refutationally complete, and provides a decision procedure for $\mathcal{SHQ}$.*

Proof. Observe that in each step in the proofs for the Lemmata 3–7, which show the refutational completeness of $Res_{\mathcal{SHQ}}$, only inferences of the rules on the maximal literals are used. Hence, in order to construct a model from a claus set, it is sufficient to include inferences on maximal literals. Therefore, if the empty clause cannot be derived from any clause set $\mathbf{N}$ using only inferences on maximal literals, then $\mathbf{N}$ is satisfiable. □

Assume we use an ordering that for a specific concept symbol A makes the literals A and $\neg A$ maximal. If the clause set M does not contain A, and we use $Res^{\prec}_{\mathcal{SHQ}}$ to decide satisfiability of the set $\mathbf{N} \cup \mathbf{M}$, resolution on A is performed before resolution on other symbols, and we use almost the same inferences then if we compute N^{-A} first and then decide satisfiability of $\mathbf{N}^{-A} \cup \mathbf{M}$. The problem is that not all resolvents on A are included in $\mathbf{N}^{-A}$. For example, assume $\mathbf{N}$ contains two clauses $\neg D_1 \sqcup A$ and $\neg D_2 \sqcup \neg A$. Resolvents containing more than one negative definer literals are not included in $\mathbf{N}^{-A}$, and therefore the resolvent $\neg D_1 \sqcup \neg D_2$ is not included in $\mathbf{N}^{-A}$. This is not a problem, since we already showed in the last section that these inferences are not necessary to decide satisfiability of clause sets. However, it might be the case that the definers D_1 and D_2 occur in number restrictions on which one of the combination rules can be applied, so that we introduce the new definer D_{12} representing D_1 and D_2. We then also have the two clauses $\neg D_{12} \sqcup A$ and $\neg D_{12} \sqcup \neg A$, and the resolvent $\neg D_{12}$ might be necessary to derive the empty clause, since it allows to apply the role resolution rule. Now, if D_1 and D_2 occur under number restrictions in clauses with different negative definer literals, the situation is similar, so that the situation can be generalised to definers that are connected to A. If it can be shown that for every pair (D_1, D_2) of definers that are connected to A, the conjunction of D_1 and D_2 will occur in the same way in $(\mathbf{N}^{-A} \cup \mathbf{M})^*$ as it does in $(\mathbf{N} \cup \mathbf{M})^*$, if it does occur in $(\mathbf{N} \cup \mathbf{M})^*$, then we can be sure that all resolvents on A that are crucial for inferring the empty clause from $\mathbf{N} \cup \mathbf{M}$ are included in $\mathbf{N}^{-A}$.

To summarise, in order to show that $\mathbf{N} \cup \mathbf{M} \models \bot$ iff $\mathbf{N}^{-A} \cup \mathbf{M} \models \bot$, we have to show that for every pair of definers D_1 and D_2 that are connected to A, if an inference in $(\mathbf{N} \cup \mathbf{M})^*$ introduces a new definer D_{12} representing the conjunction of D_1 and D_2, then D_{12} is also present in $(\mathbf{N}^{-A} \cup \mathbf{M})^*$, and used in an equivalent context as in $(\mathbf{N} \cup \mathbf{M})^*$.

Lemma 10. *Let A be a concept symbol and let $\boldsymbol{N}$ and $\boldsymbol{M}$ be two clause sets such that $A \notin sig(\boldsymbol{M})$. Let D be a definer that is introduced in the saturation $(\boldsymbol{N} \cup \boldsymbol{M})^*$, such that D is connected to A. Then D appears in $(\boldsymbol{N}^{-A} \cup \boldsymbol{M})^*$ in a logically equivalent context as in $(\boldsymbol{N} \cup \boldsymbol{M})^*$.*

Proof. We show that, whenever a definer D is introduced in $(\mathbf{N} \cup \mathbf{M})^*$, then D is also introduced in the computation of $(\mathbf{N}^{-A} \cup \mathbf{M})^*$, occurring in a logically equivalent context. The saturations $\mathbf{N}^*$ and $(\mathbf{N}^{-A})*$ will contain the same definers, since all rules in $Forget_{\mathcal{SHQ}}$ are sound, and we only remove clauses containing A. The interesting inferences are therefore between clauses in $(\mathbf{N} \cup \mathbf{M})^*$ and clauses in $(\mathbf{N} \cup \mathbf{M})^* \setminus \mathbf{N}^*$.

Let C_1 and C_2 be two clauses in $(\mathbf{N} \cup \mathbf{M})^*$ that each contain a number restriction with a definer D_1 and D_2 respectively that is connected to A. Let $C_3 \in (\mathbf{N} \cup \mathbf{M})^* \setminus \mathbf{N}^*$ be a clause that cannot be inferred just from $\mathbf{N}$. Denote by $\mathbf{C}_{inf}$ the inferences possible from C_1, C_2 and C_3 using $Res^{\prec}_{\mathcal{SHQ}}$, and by $\mathbf{C}'_{inf}$ the clauses that can be inferred by first applying $Forget_{\mathcal{SHQ}}$ on C_1 and C_2, and then $Res^{\prec}_{\mathcal{SHQ}}$ on the conclusions and C_3. We show that, whenever for $\mathbf{C}_{inf}$ a new definer D' connected to A is introduced, then the same definer D' is introduced in $\mathbf{C}'_{inf}$, and it appears in the same or a logically equivalent context as in $\mathbf{C}_{inf}$.

We distinguish the cases for the maximal literals L_1, L_2 and L_3 in C_1, C_2 and C_3, if they are number restrictions, where $C_1 = L_1 \sqcup C'$, $C_2 = L_2 \sqcup C'_2$ and $C_3 = L_3 \sqcup C'_3$.

1. The maximal literals of C_1, C_2 and C_3 are all $\leq$-restrictions. In $Res^{\prec}_{\mathcal{SHQ}}$ there is no rule for this case, and therefore $\mathbf{C}_{inf} = \emptyset$, for which the statement trivially holds.

2. The maximal literals of C_1 and C_2 are of the form ${\leq}n_i r.\neg D_i$, $i \in \{1,2\}$, and the maximal clause of C_3 is of the form ${\geq}n_3 r.D_3$, where $n_3 > n_1 + n_2$. By applying $Res^{\prec}_{\mathcal{SHQ}}$ on these clauses, the following set $\mathbf{C}_{inf}$ is inferred.

1. $C'_1 \sqcup C'_3 \sqcup {\geq}(n_3 - n_1)r.D_{13}$
2. $C'_2 \sqcup C'_3 \sqcup {\geq}(n_3 - n_2)r.D_{23}$
3. $C'_1 \sqcup C'_2 \sqcup C_3 \sqcup {\geq}(n_3 - n_1 - n_2)r.D_{123}$.

Here D_{123} is connected to A, so we have to ensure that D_{123} also occurs in $\mathbf{C}_{inf}$, in a clause that is equivalent to Clause 3. This is the case if we apply first the $\leq$-rule on the clauses C_1 and C_2, and then the the $\leq\geq$-combination rule on C_3:

1′. $C'_1 \sqcup C'_2 \sqcup {\leq}(n_1 + n_2)r.\neg D_{12}$
2′. $C'_1 \sqcup C'_2 \sqcup C_3 \sqcup {\geq}(n_3 - n_1 - n_2)r.D_{123}$

3. The maximal literals of C_1 and C_2 are of the form ${\geq}n_i r_i.D_i$, $i \in \{1,2\}$. Then the definer D_{12} representing D_1 and D_2 is introduced both in $\mathbf{C}_{inf}$ and in $\mathbf{C}_{inf}$ by applying the $\geq$-combination rule on C_1 and C_2.

4. The maximal literals of C_1 is of the form ${\geq}n_1 r_1.D_1$, and the maximal literals of C_2 and C_3 are of the form ${\leq}n_i r_i.D_i$, $i \in \{2,3\}$. The definer D_{12} is introduced by applying the $\geq\leq$-combination rule on C_1 and C_2, and therefore the situation is the same for $\mathbf{C}_{inf}$ as it is for $\mathbf{C}'_{inf}$.

5. The maximal literals of C_1 and C_3 are of the form ${\geq}n_i r_i.D_i$, $i \in \{1,3\}$ and the maximal literal of C_2 is of the form ${\leq}n_2 r.\neg D_2$, where $n_i \leq n_2 \leq n_1 + n_3$, and $r_i \sqsubseteq r$, $i \in \{1,3\}$. In $\mathbf{C}_{inf}$, due to the $\geq\leq$-combination rule, there are the following inferences.

$$C'_1 \sqcup C'_2 \sqcup {\geq}(n_1 + n_3)r.(D_1 \sqcup D_3) \sqcup {\geq}1r.D_{13}$$
$$\vdots$$
$$C'_1 \sqcup C'_2 \sqcup {\geq}n'r.(D_1 \sqcup D_3) \sqcup {\geq}n''r.D_{13},$$

where $n' = max(n_1, n_3) + 1$ and $n'' = min(n_1, n_3)$. Observe that $n_2 \geq n'$. The $\geq\leq$-combination rule can be applied on each of these clauses and C_3. Therefore, $\mathbf{C}_{inf}$ contains additionally the following clauses.

$$\begin{array}{c} C'_1 \sqcup C'_2 \sqcup C'_3 \sqcup \geq(n_1 + n_3 - n_2)r.(D_{12} \sqcup D_{23}) \sqcup \geq 1r.D_{13} \\ \vdots \\ C'_1 \sqcup C'_2 \sqcup C'_3 \sqcup \geq 1r.(D_{12} \sqcup D_{23}) \sqcup \geq(n_1 + n_3 - n_2)r.D_{13} \end{array} \tag{2}$$

The last two literals in these clauses describe all possible distributions of $n = n_1 + n_3 - n_2$ r-successors to sets of individuals satisfying $D_{12} \sqcup D_{23}$ and individuals satisfying D_{13}. If a domain element satisfies both $D_{12} \sqcup D_{23}$ and D_{13}, it satisfies all three definer concepts D_{12},D_{13} and D_{23}. Assume we abbreviate $D_{12} \sqcap D_{13} \sqcap D_{23}$ by D_{123}. The Clauses (2) are then equivalent to the following set of clauses:

$$\begin{array}{c} C'_1 \sqcup C'_2 \sqcup C'_3 \sqcup \geq(n_1 + n_3 - n_2)r.(D_{12} \sqcup D_{23} \sqcup D_{13}) \sqcup \geq 1r.D_{123} \\ \vdots \\ C'_1 \sqcup C'_2 \sqcup C'_3 \sqcup \geq 1r.(D_{12} \sqcup D_{23} \sqcup D_{13}) \sqcup (n_1 + n_3 - n_2)r.D_{123} \end{array} \tag{3}$$

We show how an equivalent of clauses is infered in $\mathbf{C}'_{inf}$. First, the $\leq\geq$-combination rule is applied on C_1 and C_2, which results in the following set of inferences.

$$\begin{array}{c} C'_1 \sqcup C'_2 \sqcup \leq(n_2 - n_1)r.\neg(D_1 \sqcup D_2) \sqcup \geq 1r.D_{12} \\ \vdots \\ C'_1 \sqcup C'_2 \sqcup \leq(n_2 - 1)r.\neg(D_1 \sqcup D_2) \sqcup \geq n_1 r.D_{12}. \end{array}$$

To make the following easier, assume we have in addition the following redundant clauses.

$$\begin{array}{c} C'_1 \sqcup C'_2 \sqcup \leq n_2 r.\neg(D_1 \sqcup D_2) \sqcup \geq(n_1 + 1)r.D_{12} \\ \vdots \\ C'_1 \sqcup C'_2 \sqcup \leq(n_3 - 1)r.\neg(D_1 \sqcup D_2) \sqcup \geq(n_3 - (n_2 - n_1))r.D_{12}. \end{array}$$

These are all subsume $C_2 \sqcup \leq n_2 r.\neg D_2$ and are therefore redundant. Therefore, adding these clauses does not lead to any inferences in $\mathbf{C}'_{inf}$ that are not entailed by the original set of clauses.

By applying the $\geq\leq$-combination rule on C_3 and each of these clauses, we obtain the following clauses.

$$\begin{array}{c} C'_1 \sqcup C'_2 \sqcup C'_3 \sqcup \geq(n_3 - (n_2 - n_1))r.(D_{13} \sqcup D_{23}) \sqcup \geq 1r.D_{12} \\ \vdots \\ C'_1 \sqcup C'_2 \sqcup C'_3 \sqcup \geq 1r.(D_{13} \sqcup D_{23}) \sqcup \geq(n_3 - (n_2 - n_1))r.D_{12}. \end{array}$$

Again the last two literals in these clauses describe all possible distributions of $n = n_3 - (n_2 - n_1) = n_1 + n_3 - n_2$ r-successors to sets of individuals satisfying D_{12}, D_{13} and D_{23}. Therefore, this set of clauses is equivalent to Clause Set (3), and D_{12} occurs in a equivalent context in $\mathbf{C}'_{inf}$ as it occurs in $\mathbf{C}_{inf}$. □

Lemma 10 shows that every definer connected to A occurring in $(\mathbf{N} \cup \mathbf{M})^*$ is included in $\mathbf{N}^{-A}$ as well. Since $\mathbf{N}^{-A}$ also contains all resolvents on A, every derivation of the empty clause from $\mathbf{N} \cup \mathbf{M}$ can be reconstructed from $\mathbf{N}^{-A} \cup \mathbf{M}$. Therefore, if $\mathbf{N} \cup \mathbf{M} \models \bot$, then also $\mathbf{N}^{-A} \cup \mathbf{M} \models \bot$. The other direction, $\mathbf{N} \cup \mathbf{M} \models \bot$ if $\mathbf{N}^{-A} \cup \mathbf{M} \models \bot$, follows from the fact that $\mathbf{N}^{-A}$ only contains clauses that are logically entailed by $\mathbf{N}$. We can therefore establish the following lemma.

Lemma 1. *Let* $\boldsymbol{N}$ *be a set of clauses and* $\mathcal{R}$ *an RBox. Then,* $\boldsymbol{N}^{-A}$ *does not contain* A*, and* $\langle \boldsymbol{N}^{-A}, \mathcal{R} \rangle \models \alpha$ *iff* $\langle \boldsymbol{N}, \mathcal{R} \rangle \models \alpha$ *for all concept inclusions* α *that do not contain* A*.*

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* $sig(\mathcal{O}) \setminus \{A\}$ *in* $\mathcal{SHQ}\mu$*. If* $\mathcal{O}^{-A}$ *does not use any fixpoint operators, it is the uniform interpolant of* $\mathcal{O}$ *in* $\mathcal{SHQ}$ *for* $sig(\mathcal{O}) \setminus \{A\}$*.*

Proof. Due to Lemma 1, the clausal representation of the uniform interpolant preserves all entailments in the target signature. Therefore, we only have to show that the rules for definer elimination are correct. The acyclic definer elimination and definer purification rule are adaptions of Ackermann's Lemma to description logic, which was first introduced in [1]. The cyclic definer elimination rule is an adaption of the generalised Ackermann's Lemma, which was introduced in [16]. The proof can therefore be done by adaptions of the proofs in [1, 16]. □

B.3 Forgetting Role Symbols

Using the results from the last two sections, we prove the correctness of the method for forgetting role symbols.

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* $\mathcal{SHQ}$*.*

Proof. Due to the construction of $\mathcal{O}^{-r}$, we clearly have $r \notin sig(\mathcal{O}^{-r})$. Therefore we only have to verify that $\mathcal{O}^{-r} \models \alpha$ iff $\mathcal{O} \models \alpha$ for all concept and role inclusions α with $r \notin sig(\alpha)$.

We first show that the clausal representation preserves all entailments α. $\mathbf{N}^{-r}$ is computed by saturating all clauses on number restrictions of the form $\leq nr.D$ and $\geq nr.D$, where r is the role forgotten. Any remaining combination rules on clauses containing r are exploiting the role hierarchy, that is they are

Ontology	Axioms	Axiom Size	Symbols	Num. Rest.
SOFTWARE	60	5.0	43	7
PEOPLE	89	5.9	71	3
ADOLENA	244	3.8	157	4
CDAO	359	4.4	211	51
TOK	361	4.1	278	35
NEOMARK	365	3.3	351	26
AMINO ACID	688	4.4	52	5
PORIFERA	874	3.9	634	60
All	380.0	4.4	224.6	23.9

Table 1. Input ontologies used for the experiments.

involving a different role s in one premise where $s \sqsubseteq r$ or $r \sqsubseteq s$. But these inferences are preserved by the monotonicity rules. If r has a super-role, the $\geq$-monotonicity rule also preserves any possible applications of the $\geq$-resolution and $\geq$-elimination rule. If r has no super-role, these rules $\mathbf{N}^{-r}$ is also saturated using this rules. We can prove in a similar way as for Lemma 10 that all necessary definers are introduced as well, that is, in whatever context r might occur when we saturate a set $\mathbf{N} \cup \mathbf{M}$, $r \notin \mathbf{M}$, if we can infer the empty set from $\mathbf{N} \cup \mathbf{M}$ then we can do so in $\mathbf{N}^{-r} \cup \mathbf{M}$. We obtain that for all entailments α with $r \notin sig(\alpha)$, we have $\mathbf{N}^{-r} \models \alpha$ iff $\mathbf{N} \models \alpha$. The next step after computing $\mathbf{N}^{-r}$ is the elimination of definer symbols. As for Theorem 3, this can be shown by adaptions of the proofs in [1, 16]. $\square$

C Evaluation

In order to evaluate the practicality of our algorithm, we implemented a prototype for concept forgetting in Scala[3] with the OWL-API[4] using some but not all of the optimisations presented in [10]. The implementation can be found online under `http://www.cs.man.ac.uk/~koopmanp`. The prototype was evaluated on a set of smaller ontologies.

Not all ontologies we used are completely in $\mathcal{SHQ}$. For this reason, we applied the prototype on the $\mathcal{SHQ}$-fragments of the ontologies. For this, we removed all TBox axioms not in $\mathcal{SHQ}$, where number restrictions of the form $=nr.C$ were replaced by corresponding conjunctions $(\leq nr.C \sqcap \geq nr.C)$, domain and range restriction axioms $\mathsf{dom}(r) \sqsubseteq C$, $\mathsf{range}(r) \sqsubseteq C$ by corresponding $\mathcal{SHQ}$-concept inclusions $\exists r.\top \sqsubseteq C$ and $\top \sqsubseteq \forall r.C$, and functional role axioms $\mathsf{func}(r)$ by corresponding $\mathcal{SHQ}$-concept inclusions $\top \sqsubseteq \leq 1r.\top$.

An overview of the ontologies used can be seen in Table 1. `SOFTWARE` and `PEOPLE` where taken from the TONES repository, the remaining ontologies where

[3] `http://www.scala-lang.org`

[4] `http://owlapi.sourceforge.net`

Ontology	Timeouts	Duration	Axioms	Axiom Size	Helper Concepts	Num. Rest.
SOFTWARE	0.0%	0.1 sec.	25.7	5.4	0.0%	3.49
PEOPLE	0.0%	0.3 sec.	138.5	11.5	53.1%	5.2
ADOLENA	2.8%	37.5 sec.	452.3	63.2	29.1%	2,318.72
CDAO	36.5%	19.5 sec.	363.4	8.4	54.4%	187.5
TOK	0.00%	0.6 sec.	358.7	7.9	98.9%	31.3
NEOMARK	4.8%	35.0 sec.	364.1	24.0	68.8%	205.7
AMINO ACID	43.7%	355.7 sec.	998.2	22.9	0.00%	67.84
PORIFERA	50.4%	127.5 sec.	1,591.5	150.8	68.9%	10,952.48
All	16.9%	51.4 sec.	434.0	30.3	48.3%	1,191.81

Table 2. Average values of the uniform interpolants computed for each ontology.

taken from the NCBO BioPortal. Table 1 shows the number of axioms, average axiom size, number of occurrences of number restrictions and the signature size of the $\mathcal{SHQ}$-fragments of the ontologies. The size of an axiom is computed using the function *size*, which is recursively defined as follows: $size(A) = 1$, where A is a concept symbol, $size(\neg C) = size(C)+1$, $size(C \sqcup D) = size(C \sqcap D) = size(C) + size(D) + 1$, $size(\geq nr.C) = size(\leq nr.C) = size(C) + 2 + n$, and $size(C \sqsubseteq D) = size(C \equiv D) = size(C) + size(D) + 1$.

For each ontology, we generated 350 random signatures, for which we computed the uniform interpolant. Each signature contained all role symbols and each concept symbol from the input ontology with a probability of 50%. Our implementation does some incomplete syntactical tests to detect tautological fixpoint expressions and replaces these by $\top$ (see [10] for details). In case the uniform interpolant would still use fixpoint operators, we computed a representation in $\mathcal{SHQ}$ using helper concepts, so that the ontologies could in theory be exported to OWL. The timeout for each computation was set to 10 minutes.

The results are shown in Table 2. For each ontology, the table shows the percentage of signatures that caused a timeout, and for the uniform interpolants that could be computed within 10 minutes, the average values of the duration of the computation, the number of axioms, the average axiom size, the percentage of ontologies using helper concepts and the number of occurrences of number restrictions in the $\mathcal{SHQ}$-representations of the uniform interpolants.

Most uniform interpolants could be computed in short amounts of time, which is mostly due to the small size of the input ontologies. The higher number of helper concepts in the uniform interpolants compared to earlier experiments [10, 9] is explained by the fact that we forgot about half of the concept symbols present in the input ontology, but kept all role symbols. This way, most cyclic relations between concept definitions had to be maintained in the output, but could only be represented using fixpoint operators or helper concepts.

Most uniform interpolants had numbers of axioms comparable to the respective input ontologies, but contained more complex axioms. This was especially the case for `PORIFERA`, where the average axiom had a size of 150.8. But Table 2 also shows that the ontologies behaved very differently, with `AMINO ACID`, despite being acylic, being the most difficult ontology for our prototype in terms of average duration, and `TOK`, being a cyclic ontology and not the the smallest of the set, which allowed in general computations of uniform interpolants in less than a second. When examining the individual ontologies, it turned out that all values also varied a lot between different signatures. In particular, a lot of ontologies contain a small number of concept symbols that are harder to forget, whereas the majority can be forgotten in few steps.