

# Computing Uniform Interpolants of $\mathcal{ALCH}$ -Ontologies with Background Knowledge

Patrick Koopmann and Renate A. Schmidt

University of Manchester, UK  
`{koopmannp, schmidt}@cs.man.ac.uk`

**Abstract:** Uniform interpolation is a technique to restrict the concept and roles symbols used in an ontology to a specified subset, while preserving all logical entailments that can be expressed using that subset. We propose the notion of relative uniform interpolation, where knowledge from a second ontology is taken into account, and present a method for the description logic  $\mathcal{ALCH}$ . Relative uniform interpolation of ontologies corresponds to strongest necessary conditions of formulae studied in classical logics.

## 1 Introduction

Uniform interpolation deals with the problem of restricting the symbols used in an ontology to a given set in such a way, that all entailments in that signature are preserved. It has applications in a range of areas covering ontology analysis, hiding confidential information, ontology reuse and ontology evolution, and methods have been developed for various description logics [5, 3, 2].

A topic that has only gained little attention in the past is computing uniform interpolants in the presence of an ontology with background knowledge. For example, it is possible that an application requires only the computation of the uniform interpolant of a subset of the ontology, whereas information about the terms to be forgotten is present in other parts of the ontology. We call these uniform interpolants *relative*, since they interpolate the ontology relative to a second ontology. Possible applications are analysing how certain concepts are related in a subset of the ontology, removing confidential information from ontologies to be shared, or approximating communication between agents that only share a limited set of common vocabulary.

Relative uniform interpolants have been investigated for classical logics under the name strongest necessary conditions (see for example [1]). We believe “relative uniform interpolant” is better suited for the description logic case, since it is unusual to speak of ontologies as sets of conditions.

We give a formal definition of relative uniform interpolants. Let  $\text{sig}_c(E)$  denote the concept symbols occurring in  $E$ , where  $E$  is an ontology or a concept inclusion.

**Definition 1.** Given two ontologies  $\mathcal{O}$  and  $\mathcal{O}_b$  and a set of concept symbols  $S$ , an ontology  $\mathcal{O}_{\text{rui}}$  is a uniform interpolant of  $\mathcal{O}$  for  $S$  relative to  $\mathcal{O}_b$ , iff the following conditions are satisfied:

1.  $\text{sig}_c(\mathcal{O}_{\text{rui}}) \subseteq S$
2. For every axiom  $\alpha$  with  $\text{sig}_c(\alpha) \subseteq S$ ,  $\mathcal{O}_b \cup \mathcal{O}_{\text{rui}} \models \alpha$  iff  $\mathcal{O}_b \cup \mathcal{O} \models \alpha$ .

If  $\mathcal{O}_b$  is empty,  $\mathcal{O}_{\text{rui}}$  is a uniform interpolant of  $\mathcal{O}$  for  $S$ . We call  $\mathcal{O}$  the input ontology and  $\mathcal{O}_b$  the background ontology.

### Resolution:

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

provided  $C_1 \sqcup C_2$  contains at most one negative definer literal.

### Role Propagation:

$$\frac{C_1 \sqcup \forall s.D_1 \quad C_2 \sqcup Qr.D_2}{C_1 \sqcup C_2 \sqcup Qr.D_3} \quad \mathcal{O} \models r \sqsubseteq s$$

where  $Q \in \{\exists, \forall\}$  and  $D_3$  is a (possibly new) definer symbol representing  $D_1 \sqcap D_2$ , provided  $C_1 \sqcup C_2$  contains at most one negative definer literal.

Figure 1: The calculus.

## 2 Computing Relative Uniform Interpolants

The method assumes that both the input ontology  $\mathcal{O}$  and the background ontology  $\mathcal{O}_b$  are normalised into a specific normal form, which is defined as follows. Let  $N_D$  be a set of special concept symbols that do not occur in the input ontology, called definer symbols.

**Definition 2.** An  $\mathcal{ALCH}$ -literal is a concept description of the form  $A$ ,  $\neg A$ ,  $\forall r.D$  or  $\exists r.D$ , where  $A$  is a concept symbol,  $r$  a role symbol and  $D \in N_D$ . A literal of the form  $\neg D$ , where  $D \in N_D$ , is called negative definer literal. A TBox is in  $\mathcal{ALCH}$ -clausal form if every axiom is of the form  $\top \sqsubseteq L_1 \sqcup \dots \sqcup L_n$ , where each  $L_i$  is an  $\mathcal{ALCH}$ -literal and at most one  $L_i$  is a negative definer literal. The right part of such a concept inclusion is called  $\mathcal{ALCH}$ -clause. We assume that  $\mathcal{ALCH}$ -clauses are represented as sets of literals.

Each ontology can be polynomially transformed into  $\mathcal{ALCH}$ -normal form by using standard flattening and conjunctive normal form transformations.

We first present the method for computing uniform interpolants relative to an empty background ontology (standard uniform interpolation) as described in [3].

In order to compute a uniform interpolant of an ontology  $\mathcal{O}$  for  $S$ , we forget each symbol  $B \in \text{sig}(\mathcal{O}) \setminus S$  one

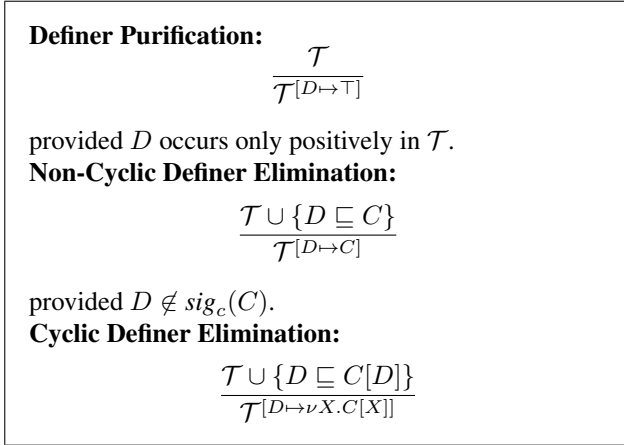


Figure 2: Definer elimination.

after another using the rules in Figure 1.

The role propagation rule may involve the introduction of a new definer symbol  $D_{12}$  representing the conjunction  $D_1 \sqcap D_2$ , which is performed by adding new clauses  $\neg D_{12} \sqcup D_1$  and  $\neg D_{12} \sqcup D_2$ . By introducing only one definer symbol per pair, the number of introduced definer symbols is limited by a finite bound. This way, the number of symbols used in the derived clause set is always finite, which also gives a finite bound on the number of clauses and ensures termination.

In order to forget a symbol  $B$ , we only apply resolution on  $B$  or definer symbols, and only apply the role propagation rule if that enables us to do further resolution steps. Afterwards all clauses containing  $B$  are filtered out, as well as clauses of the form  $\neg D \sqcup D'$ , which were added when a new definer symbol was introduced.

After all symbols have been forgotten, the definers symbols are eliminated. For this we replace for each definer symbol  $D$  the clauses of the form  $\neg D \sqcup C_i$  by a single concept inclusion  $D \sqsubseteq \prod_i C_i$ , and then apply the rules in Figure 2 exhaustively. The cyclic definer elimination rule introduces fixpoint operators to the result, which is in general unavoidable if we want to represent the uniform interpolant finitely. It is however possible to represent the result without fixpoints by approximation or using helper concepts [3]. The method can be further optimised by using redundancy elimination techniques as described in [3].

In order to compute uniform interpolants relative to non-empty background ontologies, we follow a set of support strategy. The background ontology  $\mathcal{O}_b$  is transformed into a set of clauses  $\mathbf{N}_b$  and the input ontology  $\mathcal{O}$  into a set of clauses  $\mathbf{N}_k$ . Both sets will be updated during the computation, where  $\mathbf{N}_b$  serves as set of support, and  $\mathbf{N}_k$  contains the clausal form of the relative uniform interpolant in the end.

When forgetting a concept symbol  $B$ , the rules are applied with the additional requirement that at most one premise is taken from the set  $\mathbf{N}_b$ , and derived clauses are added to  $\mathbf{N}_k$ . If a clause of the form  $\neg D \sqcup C$ ,  $D \in N_D$ , is added to  $\mathbf{N}_k$ , all clauses from  $\mathbf{N}_b$  that contain  $D$  are moved to  $\mathbf{N}_k$ . This step is necessary to ensure that no informa-

tion is lost when eliminating the definer symbols in the last step. After having forgotten a concept symbol  $B$  this way, all clauses containing  $B$  are moved to  $\mathbf{N}_b$ .

Through resolution with clauses from the set  $\mathbf{N}_b$ , it is possible that previously forgotten symbols get reintroduced to  $\mathbf{N}_k$ . For this reason, the process has to be repeated until no clause in  $\mathbf{N}_k$  contains a symbol that is neither a definer symbol nor in  $\mathcal{S}$ , where we take care that we do not add any clauses to  $\mathbf{N}_k$  that have previously been moved to  $\mathbf{N}_b$ . Since there is a finite bound on the set of clauses that can be derived using our calculus, this process always terminates. From the final clause set  $\mathbf{N}_k$ , the relative uniform interpolant  $\mathcal{O}_{\text{rui}}$  is computed by eliminating all definer symbols using the rules in Figure 2.

### 3 Outlook

We implemented a prototype of the presented method based on our implementation for computing uniform interpolants [4] and did some first experiments for small subsets of ontologies, the remaining part of the ontology was taken as background ontology. Sometimes a lot of information was transferred from the background ontology into the uniform interpolant, causing many iterations for each symbol to be forgotten, but usually we could compute relatively small relative uniform interpolants.

While the presented method extends our method for computing uniform interpolants in  $\mathcal{ALCH}$ , the key idea can be used with similar resolution based approaches, as for example for computing uniform interpolants of  $\mathcal{SHQ}$ -ontologies [5] or ontologies with ABoxes.

Open questions are how the method would perform in a deeper evaluation and whether the computed relative uniform interpolants are minimal in the sense that only as much information as needed is used from the background ontology.

### References

- [1] Patrick Doherty, Witold Lukaszewicz, and Andrzej Szalas. Computing strongest necessary and weakest sufficient conditions of first-order formulas. In *Proc. IJCAI'01*, pages 145–154. Springer, 2001.
- [2] Boris Konev, Dirk Walther, and Frank Wolter. Forgetting and Uniform Interpolation in Large-Scale Description Logic Terminologies. In *Proc. IJCAI '09*, pages 830–835. AAAI Press, 2009.
- [3] Patrick Koopmann and Renate A Schmidt. Forgetting concept and role symbols in  $\mathcal{ALCH}$ -ontologies. In *Proc. LPAR'13*, pages 552–567. Springer, 2013.
- [4] Patrick Koopmann and Renate A. Schmidt. Implementation and evaluation of forgetting in  $\mathcal{ALC}$ -ontologies. In *Proc. WoMO'13*. CEUR-WS.org, 2013.
- [5] Patrick Koopmann and Renate A. Schmidt. Count and forget: Uniform interpolation of  $\mathcal{SHQ}$ -ontologies. In *Proc. IJCAR'14*. Springer, 2014. To appear.