Forgetting and Uniform Interpolation in Large-Scale Description Logic Terminologies[*]

Boris Konev, Dirk Walther, Frank Wolter
Department of Computer Science
University of Liverpool, UK
{konev, dwalther, wolter}@liverpool.ac.uk

Abstract

We develop a framework for forgetting concepts and roles (aka uniform interpolation) in terminologies in the lightweight description logic $\mathcal{EL}$ extended with role inclusions and domain and range restrictions. Three different notions of forgetting, preserving, respectively, concept inclusions, concept instances, and answers to conjunctive queries, with corresponding languages for uniform interpolants are investigated. Experiments based on SNOMED CT (Systematised Nomenclature of Medicine Clinical Terms) and NCI (National Cancer Institute Ontology) demonstrate that forgetting is often feasible in practice for large-scale terminologies.

1 Introduction

The main application of ontologies in computer science is to fix the vocabulary of an application domain and to provide a formal theory that defines the meaning of terms built from the vocabulary and their relationships. Current applications lead to the development of very large and comprehensive ontologies such as the medical ontology SNOMED CT (Systematised Nomenclature of Medicine Clinical Terms) [Spackman, 2000] containing about 380 000 concept definitions and the National Cancer Institute ontology (NCI) [Sioutos *et al.*, 2006] containing more than 60 000 axioms. For ontologies $\mathcal{T}$ of this size, it is often of interest to *forget* a subvocabulary Σ of the vocabulary of $\mathcal{T}$; i.e., to transform $\mathcal{T}$ into a new ontology $\mathcal{T}_\Sigma$ (called a Σ-*interpolant of* $\mathcal{T}$) that contains no symbols from Σ and that is indistinguishable from $\mathcal{T}$ regarding its consequences that do not use Σ. In AI, this problem has been studied under a variety of names such as *forgetting* and *variable elimination* [Reiter and Lin, 1994; Eiter and Wang, 2008; Lang *et al.*, 2003]. In mathematical logic, this problem has been investigated as the *uniform interpolation problem* [Visser, 1996]. Computing Σ-interpolants of ontologies has a number of potential applications, e.g.,

Re-use of ontologies: when using ontologies such as SNOMED CT in an application, often only a very small fraction of its vocabulary is of interest. In this case, one could use a Σ-interpolant instead of the whole ontology, where Σ is the vocabulary not of interest for the application.

Predicate hiding: an ontology developer might not want to publish an ontology completely because a certain part of its vocabulary is not intended for public use. Again, publishing Σ-interpolants, where Σ is the vocabulary to be hidden, appears to be a solution to this problem.

Exhibiting hidden relations between terms: large ontologies are difficult to maintain as small changes to its axioms can have drastic and damaging effects. To analyze possibly unwanted consequences over a certain part Γ of the vocabulary, an ontology developer can automatically generate a complete axiomatization of the relations between terms over Γ by computing a Σ-interpolant, where Σ is the complement of Γ.

Ontology versioning: to check whether two versions of an ontology have the same consequences over their common vocabulary (or a subset thereof), one can first compute their interpolants by forgetting the vocabulary not shared by the two versions and then check whether the two interpolants are logically equivalent (i.e., have the same models).

In the description of Σ-interpolants given above, we have neither specified a language in which they are axiomatized nor did we specify the language wrt. which Σ-interpolants should be indistinguishable from the original ontology. The choice of the latter language depends on the application: for example, if one is interested in *inclusions between concepts*, then a Σ-interpolant should imply the same concept inclusions using no symbols from Σ as the original ontology. On the other hand, if the ontology is used to query instance data using *conjunctive queries*, then a Σ-interpolant together with any instance data using no symbols from Σ should imply the same certain answers to conjunctive queries using no symbols from Σ as the original ontology.

Regarding the language $\mathcal{L}$ in which Σ-interpolants should be axiomatized, one has to find a compromise between the following three conflicting goals:

(R) Standard reasoning problems (e.g., logical equivalence) in $\mathcal{L}$ should not be more complex than reasoning in the language underlying the ontology.

(I) Σ-interpolants in $\mathcal{L}$ should be uniquely determined up to logical equivalence: if $\mathcal{T}_1'$ and $\mathcal{T}_2'$ are Σ-interpolants in $\mathcal{L}$ of ontologies $\mathcal{T}_1$ and $\mathcal{T}_2$ that have the same consequences not using Σ, then $\mathcal{T}_1'$ and $\mathcal{T}_2'$ should be logically equivalent.

(E) The language $\mathcal{L}$ should be powerful enough to admit *fi-*

[*] The research was supported by EPSRC grant EP/E065279/1.

nite and *succinct* (ideally, polynomial size) axiomatizations of Σ-interpolants, and it should be possible to compute Σ-interpolants efficiently (ideally, in polynomial time).

For ontologies given in standard description logics (DLs) such as $\mathcal{EL}$ and any language between $\mathcal{ALC}$ and $\mathcal{SHIQO}$, there do not exist languages $\mathcal{L}$ achieving all these goals simultaneously.[1] To illustrate this point, let $\mathcal{L}$ be second-order logic. Then $\mathcal{L}$ trivially satisfies (E) but fails to satisfy (R) and (I), for ontologies in any standard DL.

In this paper, we consider forgetting in the lightweight description logic $\mathcal{EL}$ underlying the designated OWL2-EL profile of the upcoming OWL Version 2 extended with role inclusions and domain and range restrictions [Baader *et al.*, 2008]. This choice it motivated by the fact that forgetting appears to be of particular interest for large-scale and comprehensive ontologies and that many such ontologies are given in this language. We introduce three DLs for axiomatizing Σ-interpolants satisfying criteria (R) and (I) and preserving, respectively, inclusions between concepts, concept instances, and answers to conjunctive queries. These DLs do not satisfy (E), as Σ-interpolants sometimes to not exist or are of exponential size. We demonstrate that, nevertheless, Σ-interpolants typically exist and can be computed in practice for large-scale terminologies such as SNOMED CT and appropriate versions of NCI. Detailed proofs and additional experiments are available in a technical report [Konev *et al.*, 2009].

2 Preliminaries

Let $\mathsf{N_C}$, $\mathsf{N_R}$, and $\mathsf{N_I}$ be countably infinite and mutually disjoint sets of concept names, role names, and individual names. $\mathcal{EL}$-concepts C are built according to the rule

$$C := \quad A \quad | \quad \top \quad | \quad C \sqcap D \quad | \quad \exists r.C,$$

where $A \in \mathsf{N_C}$, $r \in \mathsf{N_R}$, and C, D range over $\mathcal{EL}$-concepts. The set of $\mathcal{ELH}^r$-inclusions consists of *concept inclusions* $C \sqsubseteq D$ and *concept equations* $C \equiv D$, *domain restrictions* $\mathsf{dom}(r) \sqsubseteq C$, *range restrictions* $\mathsf{ran}(r) \sqsubseteq C$ and *role inclusions* $r \sqsubseteq s$, where C, D are $\mathcal{EL}$-concepts and $r, s \in \mathsf{N_R}$. An $\mathcal{ELH}^r$-TBox $\mathcal{T}$ is a finite set of $\mathcal{ELH}^r$-inclusions. An $\mathcal{ELH}^r$-TBox is called $\mathcal{ELH}^r$*-terminology* if all its concept inclusions and equations are of the form $A \sqsubseteq C$ and $A \equiv C$ and no concept name occurs more than once on the left hand side. In what follows we use $A \bowtie C$ to denote expressions of the form $A \sqsubseteq C$ and $A \equiv C$.

Assertions of the form $A(a)$ and $r(a, b)$, where $a, b \in \mathsf{N_I}$, $A \in \mathsf{N_C}$, and $r \in \mathsf{N_R}$, are called ABox-assertions. An ABox is a finite set of ABox-assertions. By $\mathsf{obj}(\mathcal{A})$ we denote the set of individual names in $\mathcal{A}$. A knowledge base (KB) is a pair $(\mathcal{T}, \mathcal{A})$ consisting of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$. Assertions of the form $C(a)$ and $r(a, b)$, where $a, b \in \mathsf{N_I}$, C a $\mathcal{EL}$-concept, and $r \in \mathsf{N_R}$, are called instance assertions. To define the semantics of DLs considered in this paper we make use of the fact that DL-expressions can be regarded as formulas in FO, where FO denotes the set of first-order predicate logic formulas with equality using unary predicates in $\mathsf{N_C}$, binary predicates in $\mathsf{N_R}$, and constants from $\mathsf{N_I}$; see Figure 1 (in which the DL-constructors not considered so far are defined later). In what follows, we will not distinguish between DL-expressions and their translation into FO and regard TBoxes, ABoxes and KBs as finite subsets of FO. Thus, we use $\mathcal{T} \models \varphi$ to denote that φ follows from $\mathcal{T}$ in first-order logic even if φ is an $\mathcal{ELH}^r$-inclusion and $\mathcal{T}$ a subset of FO and similar conventions apply to DLs introduced later in this paper. FO (and, therefore, $\mathcal{ELH}^r$) is interpreted in models $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where the *domain* $\Delta^\mathcal{I}$ is a non-empty set, and $\cdot^\mathcal{I}$ is a function mapping each concept name A to a subset $A^\mathcal{I}$ of $\Delta^\mathcal{I}$, each role name r to a binary relation $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$, and each individual name a to an element $a^\mathcal{I} \in \Delta^\mathcal{I}$.

Concept C	Translation $C^\sharp$
$\top$	$x = x$
A	$A(x)$
$C \sqcap D$	$C^\sharp(x) \wedge D^\sharp(x)$
$\exists r.C$	$\exists y\,(r(x, y) \wedge C^\sharp(y))$
$\mathsf{dom}(r)$	$\exists y\,(r(x, y))$
$\mathsf{ran}(r)$	$\exists y\,(r(y, x))$
$\exists u.C$	$(x = x) \wedge \exists y\, C^\sharp(y)$
$\exists r_1 \sqcap \cdots \sqcap r_n.C$	$\exists y(r_1(x, y) \wedge \cdots \wedge r_n(x, y) \wedge C^\sharp(y))$

Inclusion α	Translation $\alpha^\sharp$
$C \sqsubseteq D$	$\forall x\,(C^\sharp(x) \rightarrow D^\sharp(x))$
$C \equiv D$	$\forall x\,(C^\sharp(x) \leftrightarrow D^\sharp(x))$
$r \sqsubseteq s$	$\forall xy\,(r(x, y) \rightarrow s(x, y))$

Figure 1: Standard translation

The most important ways of querying $\mathcal{ELH}^r$-TBoxes and KBs are subsumption (check whether $\mathcal{T} \models \alpha$ for an $\mathcal{ELH}^r$-inclusion α), instance checking (check whether $(\mathcal{T}, \mathcal{A}) \models \alpha$ for an instance assertion α), and conjunctive query answering. To define the latter, call a first-order formula $q(\vec{x})$ a *conjunctive query* if it is of the form $\exists \vec{y} \psi(\vec{x}, \vec{y})$, where ψ is a conjunction of expressions $A(t)$ and $r(t_1, t_2)$ with t, t_1, t_2 drawn from $\mathsf{N_I}$ and sequences of variables $\vec{x}$ and $\vec{y}$. If $\vec{x}$ has length k, then a sequence $\vec{a}$ of elements of $\mathsf{obj}(\mathcal{A})$ of length k is called a *certain answer* to $q(\vec{x})$ of a KB $(\mathcal{T}, \mathcal{A})$ if $(\mathcal{T}, \mathcal{A}) \models q(\vec{a})$.

3 Forgetting

A signature Σ is a subset of $\mathsf{N_C} \cup \mathsf{N_R}$[2]. Given a signature Σ, we set $\overline{\Sigma} = (\mathsf{N_C} \cup \mathsf{N_R}) \setminus \Sigma$. Given a concept, role, concept inclusion, TBox, ABox, FO-sentence, set of FO-sentences E, we denote by $\mathsf{sig}(E)$ the signature of E, that is, the set of concept and role names occurring in it. We use the term $\mathcal{ELH}^r_\Sigma$-inclusion (Σ-ABox, Σ-query, $\mathcal{L}_\Sigma$-sentence, etc.) to denote $\mathcal{ELH}^r$-inclusions (ABoxes, queries, $\mathcal{L}$-sentences, etc.) whose signature is contained in Σ.

To define forgetting, we first formalize the notion of inseparability between TBoxes wrt. a signature. Intuitively, two TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ are inseparable wrt. a signature Σ if they have the same Σ-consequences, where the set of Σ-consequences considered can either reflect subsumption queries, instance queries, or conjunctive queries, depending

[1] This follows from the fact that deciding whether TBoxes in these DLs imply the same concept inclusions over a signature is by at least one exponential harder than deciding logical equivalence [Lutz and Wolter, 2009; Lutz *et al.*, 2007].

[2] We investigate forgetting for TBoxes for DLs without nominals; thus we do not include individual names into the signature.

on the application. We give the definitions for sets of FO-sentences because we later require these notions for a variety of DLs.

Definition 1. Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be sets of FO-sentences and Σ a signature.
–$\mathcal{T}_1$ and $\mathcal{T}_2$ are concept Σ-inseparable, in symbols $\mathcal{T}_1 \equiv^C_\Sigma \mathcal{T}_2$, if for all $\mathcal{ELH}^r_\Sigma$-inclusions α: $\mathcal{T}_1 \models \alpha \Leftrightarrow \mathcal{T}_2 \models \alpha$.
–$\mathcal{T}_1$ and $\mathcal{T}_2$ are instance Σ-inseparable, in symbols $\mathcal{T}_1 \equiv^i_\Sigma \mathcal{T}_2$, if for all Σ-ABoxes $\mathcal{A}$ and Σ-instance assertions α using individual names from $\mathsf{obj}(\mathcal{A})$: $(\mathcal{T}_1, \mathcal{A}) \models \alpha \Leftrightarrow (\mathcal{T}_2, \mathcal{A}) \models \alpha$.
–$\mathcal{T}_1$ and $\mathcal{T}_2$ are query Σ-inseparable, in symbols $\mathcal{T}_1 \equiv^q_\Sigma \mathcal{T}_2$, if for all Σ-ABoxes $\mathcal{A}$, conjunctive Σ-queries $q(\vec{x})$, and vectors $\vec{a}$ of the same length as $\vec{x}$ of individual names in $\mathsf{obj}(\mathcal{A})$: $(\mathcal{T}_1, \mathcal{A}) \models q(\vec{a}) \Leftrightarrow (\mathcal{T}_2, \mathcal{A}) \models q(\vec{a})$.

The definition of forgetting (Σ-interpolants) is now straightforward.

Definition 2 (Σ-interpolant). Let $\mathcal{T}$ be an $\mathcal{ELH}^r$-TBox, Σ a finite signature, and $\mathcal{L}$ a set of FO-sentences. If $\mathcal{T}_\Sigma$ is a finite set of $\mathcal{L}_{\overline{\Sigma}}$-sentences such that $\mathcal{T} \models \varphi$ for all $\varphi \in \mathcal{T}_\Sigma$, then $\mathcal{T}_\Sigma$ is

- a concept Σ-interpolant of $\mathcal{T}$ in $\mathcal{L}$ if $\mathcal{T} \equiv^C_{\overline{\Sigma}} \mathcal{T}_\Sigma$;
- an instance Σ-interpolant of $\mathcal{T}$ in $\mathcal{L}$ if $\mathcal{T} \equiv^i_{\overline{\Sigma}} \mathcal{T}_\Sigma$;
- a query Σ-interpolant of $\mathcal{T}$ in $\mathcal{L}$ if $\mathcal{T} \equiv^q_{\overline{\Sigma}} \mathcal{T}_\Sigma$.

One can show that every query Σ-interpolant is an instance Σ-interpolant and every instance Σ-interpolant is a concept Σ-interpolant. The converse implications do not hold, even for $\mathcal{ELH}^r$-terminologies:

Example 3. Let $\mathcal{T} = \{\mathsf{ran}(r) \sqsubseteq A_1, \mathsf{ran}(s) \sqsubseteq A_2, B \equiv A_1 \sqcap A_2\}$ and $\Sigma = \{A_1, A_2\}$. One can show that the empty TBox is a concept Σ-interpolant of $\mathcal{T}$. However, the empty TBox is not an instance Σ-interpolant of $\mathcal{T}$. To show this, consider the $\overline{\Sigma}$-ABox $\mathcal{A} = \{r(a_0, b), s(a_1, b)\}$. Then $(\mathcal{T}, \mathcal{A}) \models B(b)$ but $(\emptyset, \mathcal{A}) \not\models B(b)$. Observe that *no* $\mathcal{ELH}^r$-TBox (and even no $\mathcal{SHQ}$-TBox) is an instance Σ-interpolant of $\mathcal{T}$ because it is impossible to capture the ABox $\mathcal{A}$ in a DL in which one cannot refer to the range of distinct roles in one concept. On the other hand, the TBox $\mathcal{T}' = \{\mathsf{ran}(r) \sqcap \mathsf{ran}(s) \sqsubseteq B\}$ given in an extension of $\mathcal{ELH}^r$ is an instance Σ-interpolant of $\mathcal{T}$.

Example 4. Let $\mathcal{T} = \{A \sqsubseteq \exists s.\top, s \sqsubseteq r_1, s \sqsubseteq r_2\}$ and $\Sigma = \{s\}$. Then $\mathcal{T}' = \{A \sqsubseteq \exists r_1.\top \sqcap \exists r_2.\top\}$ is an instance Σ-interpolant of $\mathcal{T}$, but $\mathcal{T}'$ is not a query Σ-interpolant of $\mathcal{T}$. To show the latter, let $\mathcal{A} = \{A(a)\}$ and let $q = \exists x(r_1(a, x) \wedge r_2(a, x))$. Then $(\mathcal{T}, \mathcal{A}) \models q$ but $(\mathcal{T}', \mathcal{A}) \not\models q$. Again, no $\mathcal{ELH}^r$-TBox (and even no TBox in $\mathcal{SHIQ}$) is a query Σ-interpolant of $\mathcal{T}$. On the other hand, the TBox $\mathcal{T}'' = \{A \sqsubseteq \exists r_1 \sqcap r_2.\top\}$ given in an extension of $\mathcal{ELH}^r$ with conjunctions of roles names is a query Σ-interpolant of $\mathcal{T}$.

Besides of exhibiting examples where concept, instance, and query Σ-interpolants are distinct, Example 3 and 4 also show that even in extremely simple cases $\mathcal{ELH}^r$ and a variety of more expressive DLs are not sufficiently powerful to express instance and query Σ-interpolants of $\mathcal{ELH}^r$-terminologies. Rather surprisingly, there also exist simple examples in which $\mathcal{ELH}^r$-TBoxes are not sufficiently expressive to axiomatize concept Σ-interpolants of $\mathcal{ELH}^r$-terminologies.

Example 5. Let $\Sigma = \{\mathsf{Research\_Inst}, \mathsf{Education\_Inst}\}$ and $\mathcal{T}$ be

$$\begin{array}{rcl} \mathsf{University} & \equiv & \mathsf{Research\_Inst} \sqcap \mathsf{Education\_Inst} \\ \mathsf{School} & \sqsubseteq & \mathsf{Education\_Inst} \\ \mathsf{ran}(\mathsf{PhD\_from}) & \sqsubseteq & \mathsf{Research\_Inst} \end{array}$$

Then there does not exist an $\mathcal{ELH}^r$-TBox that is a concept Σ-interpolant of $\mathcal{T}$. Intuitively, the reason is that there is no $\mathcal{ELH}^r_{\overline{\Sigma}}$-TBox which follows from $\mathcal{T}$ and has the following infinite set of $\overline{\Sigma}$-consequences (which are consequences of $\mathcal{T}$):
$\exists \mathsf{PhD\_from}.(\mathsf{School} \sqcap A) \sqsubseteq \exists \mathsf{PhD\_from}.(\mathsf{University} \sqcap A)$, where $A \in \overline{\Sigma}$. On the other hand, the TBox $\mathcal{T}' = \{\mathsf{ran}(\mathsf{PhD\_from}) \sqcap \mathsf{School} \sqsubseteq \mathsf{University}\}$ given in an extension of $\mathcal{ELH}^r$ is a concept Σ-interpolant of $\mathcal{T}$.

We now introduce three extensions of $\mathcal{ELH}^r$ which we propose to axiomatize concept, instance, and query Σ-interpolants.

Definition 6 ($\mathcal{EL}^{\mathsf{ran},0}$, $\mathcal{EL}^{\mathsf{ran}}$, $\mathcal{EL}^{\mathsf{ran},\sqcap,u}$). $\mathcal{C}^{\mathsf{ran},0}$-concepts are constructed using the following syntax rule

$$C := \quad D \quad | \quad \mathsf{ran}(r) \quad | \quad \mathsf{ran}(r) \sqcap D,$$

where D ranges over $\mathcal{EL}$-concepts and $r \in \mathsf{N_R}$. The set of $\mathcal{EL}^{\mathsf{ran},0}$-inclusions consists of concept inclusions $C \sqsubseteq D$ and role inclusions $r \sqsubseteq s$, where C is a $\mathcal{C}^{\mathsf{ran},0}$-concept, D an $\mathcal{EL}$-concept, and $r, s \in \mathsf{N_R}$.

$\mathcal{C}^{\mathsf{ran}}$-concepts are constructed using the following syntax rule

$$C := \quad A \quad | \quad \mathsf{ran}(r) \quad | \quad C \sqcap D \quad | \quad \exists r.C,$$

where $A \in \mathsf{N_C}$, C, D range over $\mathcal{C}^{\mathsf{ran}}$-concepts and $r \in \mathsf{N_R}$. The set of $\mathcal{EL}^{\mathsf{ran}}$-inclusions consists of all concept inclusions $C \sqsubseteq D$ and role inclusions $r \sqsubseteq s$, where C is a $\mathcal{C}^{\mathsf{ran}}$-concept, D an $\mathcal{EL}$-concept, and $r, s \in \mathsf{N_R}$.

Let u (the universal role) be a fresh logical symbol. $\mathcal{C}^{\sqcap,u}$-concepts are constructed using the following syntax rule

$$C := \quad A \quad | \quad C \sqcap D \quad | \quad \exists R.C \quad | \quad \exists u.C,$$

where $A \in \mathsf{N_C}$, C, D range over $\mathcal{C}^{\sqcap,u}$-concepts and $R = r_1 \sqcap \cdots \sqcap r_n$ with $r_1, \ldots, r_n \in \mathsf{N_R}$. The set of $\mathcal{EL}^{\mathsf{ran},\sqcap,u}$-inclusions consists of concept inclusions $C \sqsubseteq D$ and role inclusions $r \sqsubseteq s$, where C is a $\mathcal{C}^{\mathsf{ran}}$-concept, D a $\mathcal{C}^{\sqcap,u}$-concept, and $r, s \in \mathsf{N_R}$.

An X-TBox is a finite set of X-inclusions, where X ranges over $\mathcal{EL}^{\mathsf{ran}}$, $\mathcal{EL}^{\mathsf{ran},0}$, and $\mathcal{EL}^{\mathsf{ran},\sqcap,u}$.

We have the following inclusions:

$$\mathcal{ELH}^r \quad \lhd \quad \mathcal{EL}^{\mathsf{ran},0} \quad \lhd \quad \mathcal{EL}^{\mathsf{ran}} \quad \lhd \quad \mathcal{EL}^{\mathsf{ran},\sqcap,u}$$

where $\mathcal{L}_1 \lhd \mathcal{L}_2$ means that every $\mathcal{L}_1$-TBox is logically equivalent to some $\mathcal{L}_2$-TBox. The semantics of the additional constructors is straightforward and given in Figure 1. We regard the universal role u as a logical symbol (i.e., $u \notin \mathsf{N_R}$). This interpretation reflects the fact that the signature of the first-order translation of $\exists u.C$ coincides with the signature of C. Observe that the TBox given as a concept Σ-interpolant in Example 5 is an $\mathcal{EL}^{\mathsf{ran},0}$-TBox; the instance Σ-interpolant given in Example 3 is an $\mathcal{EL}^{\mathsf{ran}}$-TBox, and the query Σ-interpolant in Example 4 is an $\mathcal{EL}^{\mathsf{ran},\sqcap,u}$-TBox. For an example illustrating the need of the universal role for query Σ-interpolants, see [Lutz and Wolter, 2009].

We show that the languages introduced in Definition 6 satisfy criteria (R) and (I) from the introduction. (R) is a consequence of the following result.

Theorem 7. *The following problems are PTIME-complete for $\mathcal{EL}^{\mathsf{ran},\sqcap,u}$-TBoxes $\mathcal{T}$ and ABoxes $\mathcal{A}$: decide whether*

- $\mathcal{T} \models C \sqsubseteq D$, *for* $C \sqsubseteq D$ *an* $\mathcal{EL}^{\mathsf{ran},\sqcap,u}$*-inclusion;*
- $(\mathcal{T},\mathcal{A}) \models C(a)$*, where* C *is an* $\mathcal{EL}$*-concept.*

Deciding whether $(\mathcal{T},\mathcal{A}) \models q(\vec{a})$*, where* q *is a conjunctive query, is NP-complete, and deciding this problem for fixed* $q(\vec{a})$ *(knowledge base complexity) is PTIME-complete.*

It follows, in particular, that logical equivalence of $\mathcal{EL}^{\mathsf{ran},\sqcap,u}$-TBoxes is decidable in PTime. These complexity results are exactly the same as for $\mathcal{ELH}$-TBoxes [Rosati, 2007]. For (I) and the computation of Σ-interpolants below, we first investigate the relationship between the distinct inseparability notions introduced in Definition 1 and inseparability wrt. the new languages. (For $\mathcal{EL}$ this relationship is characterized in [Lutz and Wolter, 2009].) Let X range over the superscripts $\mathsf{ran}, 0$ and ran and $\mathsf{ran}, \sqcap, u$. Say that two finite sets of FO-sentences $\mathcal{T}_1$ and $\mathcal{T}_2$ are X-inseparable wrt. Σ, in symbols $\mathcal{T}_1 \equiv^X_\Sigma \mathcal{T}_2$, if $\mathcal{T}_1 \models \alpha \Leftrightarrow \mathcal{T}_2 \models \alpha$, for all $\mathcal{EL}^X_\Sigma$-inclusions α.

Theorem 8. *Let* $\mathcal{T}_1$ *and* $\mathcal{T}_2$ *be* $\mathcal{EL}^{\mathsf{ran},\sqcap,u}$*-TBoxes and* Σ *an infinite signature. Then the following holds:*

- $\mathcal{T}_1 \equiv^C_\Sigma \mathcal{T}_2$ *iff* $\mathcal{T}_1 \equiv^{\mathsf{ran},0}_\Sigma \mathcal{T}_2$*;*
- $\mathcal{T}_1 \equiv^i_\Sigma \mathcal{T}_2$ *iff* $\mathcal{T}_1 \equiv^{\mathsf{ran}}_\Sigma \mathcal{T}_2$*;*
- $\mathcal{T}_1 \equiv^q_\Sigma \mathcal{T}_2$ *iff* $\mathcal{T}_1 \equiv^{\mathsf{ran},\sqcap,u}_\Sigma \mathcal{T}_2$.

The condition that Σ is infinite is required only for the implication from right to left in Point 1. As we forget *finite* signatures, their complement is always infinite.

From Theorem 8 we immediately obtain that (I) is met for the three notions of Σ-interpolants. For example, assume that $\mathcal{T}_1$ and $\mathcal{T}_2$ are $\mathcal{ELH}^r$-TBoxes such that $\mathcal{T}_1 \equiv^q_{\overline{\Sigma}} \mathcal{T}_2$ and let $\mathcal{T}_1'$ and $\mathcal{T}_2'$ be query Σ-interpolants in $\mathcal{EL}^{\mathsf{ran},\sqcap,u}$ of $\mathcal{T}_1$ and $\mathcal{T}_2$, respectively. By Theorem 8, $\mathcal{T}_1' \equiv^{\mathsf{ran},\sqcap,u}_{\overline{\Sigma}} \mathcal{T}_2'$. But then $\mathcal{T}_1'$ and $\mathcal{T}_2'$ are logically equivalent: we have $\mathcal{T}_1' \models \alpha$ for all $\alpha \in \mathcal{T}_2'$ because all such α are $\mathcal{EL}^{\mathsf{ran},\sqcap,u}_{\overline{\Sigma}}$-inclusions and $\mathcal{T}_2' \models \alpha$. The converse direction holds for the same reason.

4 Computing Σ-interpolants

We give a recursive algorithm computing instance Σ-interpolants for $\mathcal{ELH}^r$-terminologies satisfying certain acyclicity conditions (similar algorithms computing concept and query Σ-interpolants are given in the technical report [Konev *et al.*, 2009]). In this section we assume wlog. that terminologies are *normalized* $\mathcal{ELH}^r$ terminologies; i.e., $\mathcal{ELH}^r$-terminologies $\mathcal{T}$ consisting of role inclusions and axioms of the form (here, and in what follows, we write $r \sqsubseteq_\mathcal{T} s$ if $\mathcal{T} \models r \sqsubseteq s$)

- $A \bowtie \exists r.B$, where $B \in \mathsf{N_C} \cup \{\top\}$;
- $A \bowtie B_1 \sqcap \cdots B_n$, where $B_1, \dots, B_n \in \mathsf{N_C}$;
- $\mathsf{dom}(s) \sqsubseteq A$ and $\mathsf{ran}(s) \sqsubseteq A$, where $A \in \mathsf{N_C}$

such that $\mathsf{dom}(s) \sqsubseteq A \in \mathcal{T}$ and $r \sqsubseteq_\mathcal{T} s$ imply $\mathsf{dom}(r) \sqsubseteq A \in \mathcal{T}$; $\mathsf{ran}(s) \sqsubseteq A \in \mathcal{T}$ and $r \sqsubseteq_\mathcal{T} s$ imply $\mathsf{ran}(r) \sqsubseteq A \in \mathcal{T}$; and $r \sqsubseteq_\mathcal{T} s$ and $s \sqsubseteq_\mathcal{T} r$ implies $r = s$. We give the acyclicity conditions required for the algorithms to terminate. The $\overline{\Sigma}$-cover $\mathcal{C}^{\overline{\Sigma}}_\mathcal{T}(r)$ of a role r wrt. a terminology $\mathcal{T}$ consists of all $s \in \overline{\Sigma}$ such that $r \sqsubseteq_\mathcal{T} s$ and there does not exist $r' \in \overline{\Sigma}$ with $r' \neq s$ and $r \sqsubseteq_\mathcal{T} r' \sqsubseteq_\mathcal{T} s$.

Definition 9 (Σ-loop). Let $\mathcal{T}$ be a normalized $\mathcal{ELH}^r$-terminology and Σ a signature. Define a relation $\prec_\Sigma \subseteq (\mathsf{N_C} \cap \Sigma) \times (\mathsf{N_C} \cap \Sigma)$ as follows: $A \prec_\Sigma B$ if $A, B \in \Sigma$ and
(a) $A \bowtie C \in \mathcal{T}$ for some C such that B occurs in C, or
(b) $A \bowtie \exists r.A' \in \mathcal{T}$ for some $A' \in \mathsf{N_C} \cup \{\top\}$ and $r \in \Sigma$ such that $\mathsf{dom}(r) \sqsubseteq B \in \mathcal{T}$, or
(c) $A \bowtie \exists r.A' \in \mathcal{T}$ for some $A' \in \mathsf{N_C} \cup \{\top\}$ and r such that there exists $s \in \mathcal{C}^{\overline{\Sigma}}_\mathcal{T}(r)$ with $\mathsf{ran}(r) \sqsubseteq B \in \mathcal{T}$, $\mathsf{ran}(s) \sqsubseteq B \notin \mathcal{T}$.
We say that $\mathcal{T}$ contains a Σ-loop if $\prec_\Sigma$ contains a cycle.

The following example illustrates this definition and shows that the existence of Σ-loops typically entails the non-existence of Σ-interpolants, even in FO.

Example 10. Consider the set of inclusions

$$\begin{array}{rcll}
\mathsf{Elephant} & \sqsubseteq & \mathsf{Mammal} & (1)\\
\mathsf{Mammal} & \sqsubseteq & \exists \mathsf{has\_mother.Mammal} & (2)\\
\mathsf{Mammal} & \sqsubseteq & \exists \mathsf{has\_mam'l\_father}.\top & (3)\\
\mathsf{dom(has\_mam'l\_father)} & \sqsubseteq & \exists \mathsf{has\_mother.Mammal} & (4)\\
\mathsf{ran(has\_mam'l\_father)} & \sqsubseteq & \mathsf{Mammal} & (5)\\
\mathsf{has\_mam'l\_father} & \sqsubseteq & \mathsf{has\_mother} & (6)
\end{array}$$

and define $\mathcal{ELH}^r$-terminologies $\mathcal{T}_1 = \{(1),(2)\}$, $\mathcal{T}_2 = \{(1),(3),(4)\}$, and $\mathcal{T}_3 = \{(1),(3),(5),(6)\}$, and let $\Sigma_i = \mathsf{sig}(\mathcal{T}_i) \setminus \{\mathsf{Elephant}, \mathsf{has\_mother}\}$, for $i = 1,2,3$. Even in FO, there exists no concept/instance/query Σ_i-interpolant of $\mathcal{T}_i$. To see this observe that in all three cases an *infinite* axiomatization of such a Σ-interpolant is given by the inclusions

$$\{\mathsf{Elephant} \sqsubseteq \overbrace{\exists \mathsf{has\_mother}. \cdots \exists \mathsf{has\_mother}}^{n}.\top \mid n \geq 1\}.$$

This theory cannot be finitely axiomatized in FO without additional predicates. Observe that $\mathcal{T}_1$ contains a Σ_1-loop as axiom (2) implies $\mathsf{Mammal} \prec_{\Sigma_1} \mathsf{Mammal}$ by clause (a) of Definition 9 for Σ_1-loops; $\mathcal{T}_2$ contains a Σ_2'-loop as axioms (3) and (4) imply $\mathsf{Mammal} \prec_{\Sigma_2'} A \prec_{\Sigma_2'} \mathsf{Mammal}$ by clauses (a) and (b), where the fresh concept name A is due to normalization introducing

$\mathsf{dom(has\_mam'l\_father)} \sqsubseteq A$, $A \sqsubseteq \exists \mathsf{has\_mother.Mammal}$;

and $\mathcal{T}_3$ contains a Σ_3-loop as axioms (3), (5), and (6) imply $\mathsf{Mammal} \prec_{\Sigma_3} \mathsf{Mammal}$ by clause (c).

Call a concept name A primitive (pseudo-primitive) in a terminology $\mathcal{T}$ if A does not occur on the left hand side of any axiom in $\mathcal{T}$ (does not occur in the form $A \equiv C$ in $\mathcal{T}$).

The intuition behind the following algorithm for Σ-interpolants is as follows: first, one can show using Theorem 8 and a sequent-style proof system for $\mathcal{ELH}^r$ that under the conditions of Theorem 11 there exists an instance Σ-interpolant consisting of (in addition to role inclusions and domain and range restrictions) concept inclusions of the form $C \sqsubseteq A$ and $A \sqsubseteq C$. In Figure 2, we compute the set $P_\Sigma(A)$ of Cs such that $C \sqsubseteq A$ is in the interpolant by making a case distinction: in Point 1 A is pseudo-primitive; in Point 2 it is defined by a conjunction; in Point 3 it is defined as $\exists r.A'$. Points 2 and 3 are recursive as they require the sets $P_\Sigma(B)$

For $A \in \mathsf{sig}(\mathcal{T})$, let $\mathsf{Pre}_\Sigma(A)$ consist of all $D = \mathsf{ran}(r)$, $D = \exists r.\top$, and $D \in \mathsf{N_C}$ such that $\mathcal{T} \models D \sqsubseteq A$ and $\mathsf{sig}(D) \subseteq \mathsf{sig}(\mathcal{T}) \cap \overline{\Sigma}$; construct $P_\Sigma(A)$ as follows:

- for A pseudo-primitive in $\mathcal{T}$, $P_\Sigma(A) = \mathsf{Pre}_\Sigma(A)$;
- if $A \equiv B_1 \sqcap \cdots \sqcap B_n \in \mathcal{T}$, then
$$P_\Sigma(A) \;=\; \{C_{B_1} \sqcap \cdots \sqcap C_{B_n} \mid (B_i \in \overline{\Sigma} \text{ and } C_{B_i} = B_i) \text{ or } (B_i \in \Sigma \text{ and } C_{B_i} \in P_\Sigma(B_i))\};$$
- if $A \equiv \exists r.A' \in \mathcal{T}$, then $P_\Sigma(A)$ is the union of $\mathsf{Pre}_\Sigma(A)$ and
  - if $A' \in \Sigma$: $\{\exists s.A' \mid s \sqsubseteq_\mathcal{T} r, s \in \overline{\Sigma}\}$;
  - if $A' \notin \Sigma$: $\{\exists s.D \mid s \sqsubseteq_\mathcal{T} r, s \in \overline{\Sigma}, D' \in P_\Sigma(A')\}$.

Figure 2: Computing $P_\Sigma(A)$

For $A \in \mathsf{sig}(\mathcal{T})$, let $\mathsf{Post}_\Sigma(A) = \{B \in \overline{\Sigma} \cap \mathsf{sig}(\mathcal{T}) \mid \mathcal{T} \models A \sqsubseteq B\}$ and construct $Q_\Sigma(A)$ as follows:

- for A primitive in $\mathcal{T}$, $Q_\Sigma(A) = \mathsf{Post}_\Sigma(A)$;
- if $A \bowtie B_1 \sqcap \cdots \sqcap B_n \in \mathcal{T}$, then
$$Q_\Sigma(A) = \mathsf{Post}_\Sigma(A) \;\cup \bigcup_{1 \leq i \leq n, B_i \in \Sigma} Q_\Sigma(B_i);$$
- if $A \bowtie \exists r.A' \in \mathcal{T}$, then $Q_\Sigma(A)$ is the union of $\mathsf{Post}_\Sigma(A)$,
$$\bigcup_{r \sqsubseteq_\mathcal{T} s} \{Q_\Sigma(B) \mid \mathsf{dom}(s) \sqsubseteq B \in \mathcal{T}, s \in \Sigma, B \in \Sigma\},$$
and
$$\{\exists s.E_s \mid s \in \mathcal{C}^\Sigma_\mathcal{T}(r)\},$$
where
$$E_s = \bigsqcap_{\substack{B \in \Sigma,\, \mathsf{ran}(r) \sqsubseteq B \in \mathcal{T} \\ \mathsf{ran}(s) \sqsubseteq B \notin \mathcal{T},\, D \in Q_\Sigma(B)}} D \;\sqcap \bigsqcap_{\substack{D \in Q_\Sigma(A') \\ A' \in \Sigma}} D \;\sqcap \bigsqcap_{\substack{B \in \overline{\Sigma} \\ \mathcal{T} \models \mathsf{ran}(r) \sqcap A' \sqsubseteq B}} B.$$

Figure 3: Computing $Q_\Sigma(A)$

when B is used in the definition of A. Σ-loops describe exactly the situation in which the recursion does not terminate. In Figure 3 we compute, in a similar way, the set $Q_\Sigma(A)$ of Cs such that $A \sqsubseteq C$ is in the interpolant. A sample run of the algorithm is provided in [Konev *et al.*, 2009].

Theorem 11. *Let Σ be a finite signature and $\mathcal{T}$ a normalized $\mathcal{ELH}^r$-terminology without Σ-loops. Then the algorithms computing $P_\Sigma(A)$ and $Q_\Sigma(A)$ in Figures 2 and 3 terminate for all $A \in \mathsf{sig}(\mathcal{T})$.*

Let $\mathcal{T}_\Sigma$ consist of the following inclusions, where A, r, and s range over $\mathsf{sig}(\mathcal{T}) \cap \overline{\Sigma}$:

- *$r \sqsubseteq s$, for $r \sqsubseteq_\mathcal{T} s$;*
- *$D \sqsubseteq A$, for all $D \in P_\Sigma(A)$;*
- *$A \sqsubseteq D$, for all $D \in Q_\Sigma(A)$;*
- *$\mathsf{ran}(r) \sqsubseteq D$, for all $D \in Q_\Sigma(B)$ such that $\mathsf{ran}(r) \sqsubseteq B \in \mathcal{T}$ and $B \in \Sigma$;*
- *$\mathsf{dom}(r) \sqsubseteq D$, for all $D \in Q_\Sigma(B)$ such that $\mathsf{dom}(r) \sqsubseteq B \in \mathcal{T}$ and $B \in \Sigma$;*

Then $\mathcal{T}_\Sigma$ is an instance Σ-interpolant of $\mathcal{T}$.

$P_\Sigma(A)$ and $Q_\Sigma(A)$ are both of exponential size, in the worst case. For $P_\Sigma(A)$, this is clear from Point 2 of the construction: let $\mathcal{T}$ consist of $A \equiv B_1 \sqcap \cdots \sqcap B_n$ and $A_i^j \sqsubseteq B_i$ ($1 \leq i, j \leq n$) and let $\Sigma = \{B_i \mid 1 \leq i \leq n\}$. Then $P_\Sigma(A)$ is of size n^n, and one can show that there does not exist a shorter Σ-interpolant in $\mathcal{EL}^{\mathsf{ran},\sqcap,u}$. For $Q_\Sigma(A)$ this follows from the fact that one might have to construct a complete unfolding of the terminology.

If we admit disjunctions in C in axioms $C \sqsubseteq D$ of Σ-interpolants, then we can replace, in Point 2, $P_\Sigma(A)$ for $A \equiv B_1 \sqcap \cdots \sqcap B_n \in \mathcal{T}$ by the singleton set consisting of

$$\bigsqcap_{1 \leq i \leq n, B_i \in \overline{\Sigma}} B_i \;\sqcap \bigsqcap_{1 \leq i \leq n, B_i \in \Sigma} \;\bigsqcup_{C_{B_i} \in P_\Sigma(B_i)} C_{B_i}.$$

We will see below that in practice this construction leads to much smaller Σ-interpolants. However, this improvement does not come for free. Consider the language $\mathcal{EL}^{\mathsf{ran},\sqcup}$, where the only difference to $\mathcal{EL}^{\mathsf{ran}}$ is that $\mathcal{C}^{\mathsf{ran}}$-concepts now admit '$\sqcup$' as a binary concept constructor. Every $\mathcal{EL}^{\mathsf{ran},\sqcup}$-TBox is logically equivalent to an (exponentially larger) $\mathcal{EL}^{\mathsf{ran}}$-TBox, and so $\mathcal{EL}^{\mathsf{ran},\sqcup}$ inherits many desirable properties from $\mathcal{EL}^{\mathsf{ran}}$. However, one can show that, in contrast to $\mathcal{EL}^{\mathsf{ran}}$, logical equivalence between $\mathcal{EL}^{\mathsf{ran},\sqcup}$-TBoxes is coNP-hard.

5 Experiments

We have implemented a prototype called NUI that computes instance Σ-interpolants as presented in Theorem 11. We have applied NUI to a version of SNOMED CT dated 09 February 2005 (without two left-identities) and the $\mathcal{ELH}^r$-fragment of the release 08.08d of NCI. The first terminology has approx. 380K axioms, almost the same number of concept names, and 56 role names. The $\mathcal{ELH}^r$-fragment of NCI has approx. 63K axioms, approx. 65K concept names, and 123 role names. We note that the algorithms given above compute (for ease of exposition) a large number of redundant axioms and NUI implements a variety of straightforward optimizations.

First observe that neither SNOMED CT nor NCI contain any Σ-loops, for any signature Σ. Thus, Σ-interpolants always exist and can, in principle, be computed using our algorithm.

In our experiments, we focus on the case of forgetting a large signature Σ (and keeping a "small" signature $\overline{\Sigma} \cap \mathsf{sig}(\cdot)$), as this corresponds to many application scenarios. The experiments have been performed on a standard PC with 2.13 GHz and 3 GB of RAM.

Success rate: Table 1 shows the rate at which NUI succeeds to compute instance Σ-interpolants of SNOMED CT and NCI wrt. various signatures. All failed cases are due to memory overflow after several hours. For each table entry, 100 samples have been used. The signatures contain concept and role names randomly selected from the full signature of SNOMED CT (we never forget the role 'roleGroup' as this would make forgetting trivial) and NCI, respectively. $\overline{\Sigma} \cap \mathsf{sig}(\cdot)$ always contains 20 role names. For NCI and signatures of size $\leq 4\,000$, NUI had a 100% success rate.

Size: We compare the size of instance Σ-interpolants of SNOMED CT and NCI computed by NUI with the size of extracted $\overline{\Sigma} \cap \mathsf{sig}(\cdot)$-modules; i.e., minimal *subsets* $\mathcal{M}$ of the respective terminologies which preserve, e.g., inclusions between $\overline{\Sigma} \cap \mathsf{sig}(\cdot)$-concepts. We use MEX-modules [Konev *et*

$\lvert\overline{\Sigma} \cap \mathsf{sig}(\cdot)\rvert$	SNOMED CT	$\lvert\overline{\Sigma} \cap \mathsf{sig}(\cdot)\rvert$	NCI
2 000	93.0%	5 000	97.0%
3 000	84.5%	10 000	81.1%
4 000	67.0%	15 000	72.0%
5 000	59.5%	20 000	59.2%

Table 1: Success rate of NUI

al., 2008a] of SNOMED CT and $\top$-local modules [Cuenca Grau *et al.*, 2007] of NCI. The size of Σ-interpolants, terminologies, and modules is measured as number of symbols rather than number of axioms as Σ-interpolants can contain large axioms.

For SNOMED CT, we computed instance Σ-interpolants and $\overline{\Sigma}\cap\mathsf{sig}(\cdot)$-modules wrt. 100 random signatures with 3 000 concept names and 20 role names in $\overline{\Sigma} \cap \mathsf{sig}(\cdot)$. 48.19% of instance Σ-interpolants are smaller than the corresponding modules, whose sizes lie between 2.94% and 3.21% of SNOMED CT. However, the largest instance Σ-interpolant is more than 11 times larger than SNOMED CT.

For NCI, we computed instance Σ-interpolants and $\overline{\Sigma} \cap \mathsf{sig}(\cdot)$-modules wrt. 100 random signatures with 7 000 concept names and 20 role names in $\overline{\Sigma} \cap \mathsf{sig}(\cdot)$. 74.47% of the instance Σ-interpolants are smaller than the corresponding modules, whose sizes lie between 21.62% and 23.17% of NCI. NUI computes each of those interpolants within 25 min. However, the largest instance Σ-interpolant is more than 12 times larger than NCI.

Forgetting with disjunction: All failures in Table 1 are due to the fact that $P_\Sigma(A)$ is too large. Indeed, if we admit disjunction and consider $\mathcal{EL}^{\mathsf{ran},\sqcup}$, then NUI succeeds to compute *all* Σ-interpolants from Table 1, each within 15 min. Moreover, for NCI, no signature for which NUI fails has been detected. For SNOMED CT, however, NUI still typically fails for $\lvert\overline{\Sigma} \cap \mathsf{sig}(\cdot)\rvert \geq 30\,000$.

6 Discussion

The notion of forgetting in DL ontologies has recently been investigated in a number of research papers. [Kontchakov *et al.*, 2008; Wang *et al.*, 2008] consider forgetting in DL-Lite and [Eiter *et al.*, 2006] investigate in how far forgetting in DLs can be reduced to forgetting in logic programs. [Konev *et al.*, 2008b], on which this paper is based, proposes forgetting for acyclic $\mathcal{EL}$-terminologies and concept inclusions.

The main novel contributions of this paper are (i) the first algorithms with experimental results indicating the practical feasibility of forgetting in DL-terminologies and (ii) the first systematic analysis of the distinct languages required to axiomatize Σ-interpolants for distinct queries languages. Many open problems remain; e.g., we conjuncture that Σ-interpolants of $\mathcal{ELH}^r$-terminologies (and possibly even TBoxes) exist in the languages introduced whenever they exist in FO. Such a result would provide further justification for those languages. Secondly, it would be of interest to prove decidability (and complexity) of the decision problem whether there exists a Σ-interpolant for a given $\mathcal{ELH}^r$-terminology (TBox). Note that our acyclicity conditions are sufficient but not necessary for the existence of Σ-interpolants.

References

[Baader *et al.*, 2008] F. Baader, S. Brandt, and C. Lutz. Pushing the EL-envelope further. In *Proc. of OWLED DC*, 2008.

[Cuenca Grau *et al.*, 2007] B. Cuenca Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Just the right amount: Extracting modules from ontologies. In *Proc. of WWW'07*, 2007.

[Eiter and Wang, 2008] T. Eiter and K. Wang. Semantic forgetting in answer set programming. *Artificial Intelligence*, 172(14):1644–1672, 2008.

[Eiter *et al.*, 2006] T. Eiter, G. Ianni, R. Schindlauer, H. Tompits, and K. Wang. Forgetting in managing rules and ontologies. In *Proc. of Web Intelligence*, 2006.

[Konev *et al.*, 2008a] B. Konev, C. Lutz, D. Walther, and F. Wolter. Semantic modularity and module extraction in description logic. In *Proc. of ECAI'08*, 2008.

[Konev *et al.*, 2008b] B. Konev, D. Walther, and F. Wolter. The logical difference problem for description logic terminologies. In *Proc. of IJCAR'08*, 2008.

[Konev *et al.*, 2009] B. Konev, D. Walther, and F. Wolter. Forgetting and uniform interpolation in large-scale description logic terminologies. Technical Report ULCS-09-006, University of Liverpool, 2009.

[Kontchakov *et al.*, 2008] R. Kontchakov, F. Wolter, and M. Zakharyaschev. Can you tell the difference between DL-lite ontologies. In *Proc. of KR'08*, 2008.

[Lang *et al.*, 2003] J. Lang, P. Liberatore, and P. Marquis. Propositional independence: formula-variable independence and forgetting. *Journal of Artificial Intelligence Research*, 18:391–443, 2003.

[Lutz and Wolter, 2009] C. Lutz and F. Wolter. Deciding inseparability and conservative extensions in the description logic EL. To appear in *J. of Symbolic Computation*, 2009.

[Lutz *et al.*, 2007] C. Lutz, D. Walther, and F. Wolter. Conservative extensions in expressive description logics. In *Proc. of IJCAI'07*, 2007.

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

[Rosati, 2007] R. Rosati. On conjunctive query answering in $\mathcal{EL}$. In *Proc. of DL'07*, 2007.

[Sioutos *et al.*, 2006] N. Sioutos, S. de Coronado, M.W. Haber, F.W. Hartel, W.L. Shaiu, and L.W. Wright. NCI thesaurus: a semantic model integrating cancer-related clinical and molecular information. *Journal of Biomedical Informatics*, 40(1):30–43, 2006.

[Spackman, 2000] K.A. Spackman. Managing clinical terminology hierarchies using algorithmic calculation of subsumption: Experience with SNOMED-RT. 2000.

[Visser, 1996] A. Visser. Uniform interpolation and layered bisimulation. In *Gödel '96 (Brno, 1996)*, volume 6 of *Lecture Notes Logic*. Springer Verlag, 1996.

[Wang *et al.*, 2008] Z. Wang, K. Wang, R. Topor, and J.Z. Pan. Forgetting in DL-Lite. In *Proc. of ESWC'08*, 2008.