Modularity in DL-Lite

Roman Kontchakov[1], Frank Wolter[2], and Michael Zakharyaschev[1]

[1] School of Computer Science and Information Systems, Birkbeck College, London,
{roman,michael}@dcs.bbk.ac.uk
[2] Department of Computer Science, University of Liverpool, U.K.,
frank@csc.liv.ac.uk

Abstract. We develop a formal framework for modular ontologies by analysing four notions of conservative extensions and their applications in refining, re-using, merging, and segmenting ontologies. For two members of the *DL-Lite* family of description logics, we prove important meta-properties of these notions such as robustness under joins, vocabulary extensions, and iterated import of ontologies. The computational complexity of the corresponding reasoning tasks is investigated.

1 Introduction

In computer science and related areas, ontologies are used to define the meaning of vocabularies designed to speak about some domains of interest. In ontology languages based on description logics (DLs), such an ontology typically consists of a TBox stating which inclusions hold between complex concepts built over the vocabulary. An increasingly important application of ontologies is management of large amounts of data, where the ontology is used to provide flexible and efficient access to repositories consisting of data sets of instances of concepts and relations of the vocabulary. In DLs, such repositories are typically modelled as ABoxes.

Developing ontologies for this and other purposes is a difficult task. When dealing with DLs, the ontology designer is supported by efficient reasoning tools for classification, instance checking and some other reasoning problems. However, it is generally recognised that this support is not sufficient when ontologies are not developed as 'monolithic entities' but rather result from importing, merging, combining, re-using, refining and extending existing ontologies. In all those cases, reasoning support for analysing the impact of the respective operation on the ontology would be highly desirable. Typical reasoning tasks in this case may include the following:

- If we add some new concepts, relations and axioms to our ontology, can new assertions over the vocabulary of the original TBox be derived from the extended TBox?
- When importing an ontology, do we change the meaning of its vocabulary?
- When looking for a definition of some concepts, what part of the existing ontology defining them should be used?

Recently, the notion of conservative extension has been identified as fundamental for dealing with problems of this kind [1–5]. Parameterising this notion by a language $\mathcal{L}$, we say that a TBox $\mathcal{T}$ is a *conservative extension* of a TBox $\mathcal{T}'$ w.r.t. $\mathcal{L}$ if $\mathcal{T} \models \alpha$ implies $\mathcal{T}' \models \alpha$, for every α from $\mathcal{L}$ which only uses the vocabulary of $\mathcal{T}'$. In these papers, the main emphasis has been on languages $\mathcal{L}$ consisting of TBox axioms over some description logic (such as $\mathcal{ALC}$) and the much stronger notion of *model conservativity* which corresponds to the assumption that α can be taken from any language with standard Tarski semantics (e.g., second-order logic). Considering TBox axioms is motivated by the fact that ontologies are developed and represented via such axioms. They are the syntactic objects an ontology designer is working with, and a possibility to derive some new axioms appears therefore to be a good indicator as to whether the meaning of symbols has changed in any relevant sense. The notion of model conservativity is motivated by its flexibility: whatever language $\mathcal{L}$ is chosen, no new consequences in $\mathcal{L}$ will be derivable [5, 4]. A third option (which lies between the two above as far as expressivity is concerned) is as follows: if the main application of the ontologies $\mathcal{T}$ and $\mathcal{T}'$ is to provide a vocabulary and its meaning for posing queries to ABoxes, then it appears to be of interest to regard $\mathcal{T}$ as a conservative extension of $\mathcal{T}'$ if, for every ABox $\mathcal{A}$ and every (say, positive existential) query q in the vocabulary of $\mathcal{T}'$, any answer to q given by $(\mathcal{T}, \mathcal{A})$ is given by $(\mathcal{T}', \mathcal{A})$ as well. It can thus be seen that there is a variety of notions of conservativity which can be used to formally define modularity in ontologies. The choice of the appropriate one depends on what the ontologies are supposed to be used for, the computational complexity of the corresponding reasoning tasks, and the relevant meta-properties and 'robustness' of the notion of conservativity.

Here we investigate these and related notions of conservative extensions for the *DL-Lite* family of description logics [6–8]. *DL-Lite* and its variants are weak descriptions logics that have been designed in order to facilitate efficient query-answering over large data sets. We introduce four different notions of conservativity for two languages within this family, motivate their relevance for modularity and re-use of ontologies, study their meta-properties, and determine the computational complexity of the corresponding reasoning tasks. All the proofs can be found in the Appendix available at `http://www.csc.liv.ac.uk/~frank`.

2 The *DL-Lite* Family

The *DL-Lite* family of DLs has been introduced and investigated in [6–8] with the aim of establishing maximal subsets of DL constructors for which the data complexity of query answering stays within LogSpace. The 'covering' DL of the *DL-Lite* family is known as $\textit{DL-Lite}_{bool}$ [8]. As $\textit{DL-Lite}_{bool}$ itself contains classical propositional logic, query answering in it is coNP-hard, but by taking the Horn-fragment $\textit{DL-Lite}_{horn}$ of $\textit{DL-Lite}_{bool}$, one obtains a language for which query answering is within LogSpace [8] (precise formulations of these results are given below).

The language of *DL-Lite*$_{bool}$ has *object names* $a_1, a_2, \ldots$, *concept names* $A_1, A_2, \ldots$, and *role names* $P_1, P_2, \ldots$. Complex *roles* R and *DL-Lite*$_{bool}$ *concepts* C are defined as follows:

$$\begin{array}{lllllllll} R & ::= & P_i & \mid & P_i^-, & & & & \\ B & ::= & \bot & \mid & \top & \mid & A_i & \mid & \geq q\, R, \\ C & ::= & B & \mid & \neg C & \mid & C_1 \sqcap C_2, & & \end{array}$$

where $q \geq 1$. The concepts of the form B above are called *basic*. A *DL-Lite*$_{bool}$ *concept inclusion* is of the form $C_1 \sqsubseteq C_2$, where C_1 and C_2 are *DL-Lite*$_{bool}$ concepts. A *DL-Lite*$_{bool}$ *TBox* is a finite set of *DL-Lite*$_{bool}$ concept inclusions. (Other concept constructs like $\exists R$, $\leq q\, R$ and $C_1 \sqcup C_2$ will be used as standard abbreviations.)

As mentioned above, we also consider the *Horn* fragment *DL-Lite*$_{horn}$ of *DL-Lite*$_{bool}$: a *DL-Lite*$_{horn}$ *concept inclusion* is of the form

$$\bigsqcap_k B_k \sqsubseteq B,$$

where B and the B_k are basic concepts. In this context, basic concepts will also be called *DL-Lite*$_{horn}$ *concepts*. Note that the axioms $\bigsqcap_k B_k \sqsubseteq \bot$ and $\top \sqsubseteq B$ are legal in *DL-Lite*$_{horn}$. A *DL-Lite*$_{horn}$ *TBox* is a finite set of *DL-Lite*$_{horn}$ concept inclusions. For other fragments of *DL-Lite*$_{bool}$ we refer the reader to [6–8]. It is worth noting that in *DL-Lite*$_{horn}$ we can express both *global functionality* of a role and *local functionality* (i.e., functionality restricted to a (basic) concept B) by means of the axioms $\geq 2\, R \sqsubseteq \bot$ and $B \sqcap \geq 2\, R \sqsubseteq \bot$, respectively.

Let $\mathcal{L}$ be either *DL-Lite*$_{bool}$ or *DL-Lite*$_{horn}$. An $\mathcal{L}$-*ABox* is a set of assertions of the form $C(a_i)$, $R(a_i, a_j)$, where each C is an $\mathcal{L}$-concept, R a role, and a_i, a_j are object names. An $\mathcal{L}$ *knowledge base* ($\mathcal{L}$-*KB*) is a pair $(\mathcal{T}, \mathcal{A})$ consisting of an $\mathcal{L}$-TBox $\mathcal{T}$ and an $\mathcal{L}$-ABox $\mathcal{A}$.

An *interpretation* $\mathcal{I}$ is a structure of the form $(\Delta^{\mathcal{I}}, A_1^{\mathcal{I}}, \ldots, P_1^{\mathcal{I}}, \ldots a_1^{\mathcal{I}}, \ldots)$, where $\Delta^{\mathcal{I}}$ is non-empty, $A_i^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, $P_i^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ and $a_i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ such that $a_i^{\mathcal{I}} \neq a_j^{\mathcal{I}}$, for $a_i \neq a_j$ (i.e., we adopt the unique name assumption). The *extension* $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ of a concept C is defined as usual. A concept inclusion $C_1 \sqsubseteq C_2$ is *satisfied* in $\mathcal{I}$ if $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$; in this case we write $\mathcal{I} \models C_1 \sqsubseteq C_2$. $\mathcal{I}$ is a *model for* a TBox $\mathcal{T}$ if all concept inclusions from $\mathcal{T}$ are satisfied in $\mathcal{I}$. A concept inclusions $C_1 \sqsubseteq C_2$ *follows from* $\mathcal{T}$, $\mathcal{T} \models C_1 \sqsubseteq C_2$ in symbols, if every model for $\mathcal{T}$ satisfies $C_1 \sqsubseteq C_2$. A concept C is $\mathcal{T}$-*satisfiable* if there exists a model $\mathcal{I}$ for $\mathcal{T}$ with $C^{\mathcal{I}} \neq \emptyset$. We say that $\mathcal{I}$ is a *model* for an $\mathcal{L}$-KB $(\mathcal{T}, \mathcal{A})$ if $\mathcal{I}$ is a model for $\mathcal{T}$ and every assertion of $\mathcal{A}$ is satisfied in $\mathcal{I}$.

An (*essentially positive existential*) $\mathcal{L}$-*query* $q(x_1, \ldots, x_n)$ is a formula

$$\exists y_1 \cdots \exists y_m \varphi(x_1, \ldots, x_n, y_1, \ldots, y_m),$$

where φ is constructed, using only $\wedge$ and $\vee$, from atoms of the form $C(t)$ and $P(t_1, t_2)$, with C being an $\mathcal{L}$-concept, P a role, and t_i being either a variable from the list $x_1, \ldots, x_n, y_1, \ldots, y_m$ or an object name. Given an $\mathcal{L}$-KB $(\mathcal{T}, \mathcal{A})$ and an

$\mathcal{L}$-query $q(\boldsymbol{x})$, with $\boldsymbol{x} = x_1, \ldots, x_n$, we say that an n-tuple $\boldsymbol{a}$ of object names is an *answer* to $q(\boldsymbol{x})$ w.r.t. $(\mathcal{T}, \mathcal{A})$ and write $(\mathcal{T}, \mathcal{A}) \models q(\boldsymbol{a})$ if, for every model $\mathcal{I}$ for $(\mathcal{T}, \mathcal{A})$, we have $\mathcal{I} \models q(\boldsymbol{a})$. The data complexity of the query answering problem for *DL-Lite*$_{horn}$ knowledge bases is in LogSpace, while for *DL-Lite*$_{bool}$ it is coNP-complete [8].

3 Types of Conservativity and Modularity

In this section, we introduce four different notions of conservative extension for *DL-Lite*$_{bool}$ and *DL-Lite*$_{horn}$, discuss their applications, investigate their meta-properties, and determine the computational complexity of the corresponding reasoning tasks. By a *signature* we understand here a finite set Σ of concept names and role names.[3] Given a concept, role, TBox, ABox, or query E, we denote by $\mathsf{sig}(E)$ the *signature* of E, that is, the set of concept and role names that occur in E. It is worth noting that the symbols $\bot$ and $\top$ are regarded as logical symbols. Thus, $\mathsf{sig}(\bot) = \mathsf{sig}(\top) = \emptyset$. A concept (role, TBox, ABox, query) E is called a *Σ-concept* (*role*, *TBox*, *ABox*, *query*, respectively) if $\mathsf{sig}(E) \subseteq \Sigma$. Thus, P^- is a Σ-role iff $P \in \Sigma$. As before, we will use $\mathcal{L}$ as a generic name for *DL-Lite*$_{bool}$ and *DL-Lite*$_{horn}$.

Definition 1 (deductive conservative extension). Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be $\mathcal{L}$-TBoxes and Σ a signature. We call $\mathcal{T}_1 \cup \mathcal{T}_2$ a (*deductive*) *conservative extension of* $\mathcal{T}_1$ *in* $\mathcal{L}$ *w.r.t.* Σ if, for every $\mathcal{L}$-concept inclusion $C_1 \sqsubseteq C_2$ with $\mathsf{sig}(C_1 \sqsubseteq C_2) \subseteq \Sigma$, we have $\mathcal{T}_1 \models C_1 \sqsubseteq C_2$ whenever $\mathcal{T}_1 \cup \mathcal{T}_2 \models C_1 \sqsubseteq C_2$.

This notion of deductive conservative extension is appropriate in the following situations; see also [2]. (i) Suppose that $\mathcal{T}_1$ is a TBox which does not cover part of its domain in sufficient detail. An ontology engineer, say Eve, decides to expand it by axioms $\mathcal{T}_2$, but wants to be sure that by doing this she does not interfere with the derivable inclusions between Σ-concepts. Then she should check whether $\mathcal{T}_1 \cup \mathcal{T}_2$ is a conservative extension of $\mathcal{T}_1$ in $\mathcal{L}$ w.r.t. to Σ. (ii) If the designer of an ontology $\mathcal{T}_2$ *imports* an ontology $\mathcal{T}_1$ and wants to ensure that no extra inclusions between $\mathsf{sig}(\mathcal{T}_1)$-concepts are derivable after importing the ontology, then again she should check whether $\mathcal{T}_1 \cup \mathcal{T}_2$ is a conservative extension of $\mathcal{T}_1$ in $\mathcal{L}$ w.r.t. $\mathsf{sig}(\mathcal{T}_1)$. Observe that in *DL-Lite*$_{bool}$, $\mathcal{T}_1 \cup \mathcal{T}_2$ is a deductive conservative extension of $\mathcal{T}_1$ iff every $\mathcal{T}_1$-satisfiable *DL-Lite*$_{bool}$ concept C with $\mathsf{sig}(C) \subseteq \Sigma$ is also $\mathcal{T}_1 \cup \mathcal{T}_2$-satisfiable.

Theorem 1. *For any DL-Lite*$_{horn}$ *TBoxes* $\mathcal{T}_1$, $\mathcal{T}_2$ *and any signature* Σ, *the following two conditions are equivalent:*

- $\mathcal{T}_1 \cup \mathcal{T}_2$ *is a conservative extension of* $\mathcal{T}_1$ *in DL-Lite*$_{bool}$ *w.r.t* Σ;
- $\mathcal{T}_1 \cup \mathcal{T}_2$ *is a conservative extension of* $\mathcal{T}_1$ *in DL-Lite*$_{horn}$ *w.r.t.* Σ.

[3] In the languages we consider, object names do not occur in TBoxes. Therefore, in this paper, we assume that signatures do not contain object names. When considering languages with nominals one would have to allow for object names in signatures.

For DL-Lite$_{horn}$ TBoxes, the problem of deciding whether $\mathcal{T}_1 \cup \mathcal{T}_2$ is a conservative extension of $\mathcal{T}_1$ in DL-Lite$_{horn}$ w.r.t Σ is coNP*-complete. For DL-Lite$_{bool}$ TBoxes, this problem is Π_2^p-complete.*

Observe that the complexity lower bounds follow immediately from the same lower bounds for the corresponding reasoning problems in classical propositional (Horn) logic. The upper bounds are proved in the Appendix. We remind the reader that conservativity is much harder for most DLs: it is ExpTime-complete for $\mathcal{EL}$ [9], 2ExpTime-complete for $\mathcal{ALC}$ and $\mathcal{ALCQI}$, and undecidable for $\mathcal{ALCQIO}$ [2, 4]. To explain, at a very high level, the reason for these results we consider the notion of a conservative extension in *DL-Lite$_{bool}$*: let $\mathcal{T}_1$ and $\mathcal{T}_2$ be TBoxes and Σ a signature with $\Sigma \subseteq \mathsf{sig}(\mathcal{T}_1)$. $\mathcal{T}_1 \cup \mathcal{T}_2$ is not a conservative extension of $\mathcal{T}_1$ in *DL-Lite$_{bool}$* w.r.t. Σ if, and only if, there exists a concept C with $\mathsf{sig}(C) \subseteq \Sigma$ such that C is satisfiable relative to $\mathcal{T}_1$ but not relative to $\mathcal{T}_1 \cup \mathcal{T}_2$. We call such a concept C a *witness-concept*. Thus, a decision procedure for conservativity can be regarded as a systematic search for such a witness-concept. In standard description logics such as *DL-Lite$_{bool}$*, $\mathcal{EL}$, $\mathcal{ALC}$, etc. the space of all possible witnesses is infinite. (This observation implies that from the decidability of the problem whether a concept is satisfiable w.r.t. a TBox it does not necessarily follow that conservativity is decidable.) Now, we prove in the Appendix that for *DL-Lite$_{bool}$* the existence of some witness concept implies the existence of a witness concept of size *polynomial* in the size of $\mathcal{T}_1$ and $\mathcal{T}_2$ and which uses only the numeral parameters which occur in number restrictions from $\mathcal{T}_1 \cup \mathcal{T}_2$. In contrast, in $\mathcal{EL}$ one can construct examples in which minimal witnesses for non-conservativity are of double exponential size in the size of $\mathcal{T}_1$ and $\mathcal{T}_2$ [9]. In $\mathcal{ALC}$, one can even enforce minimal witness concepts of triple exponential size [2]. The reason for this difference is the availability of *qualified* quantification in those language, and its absence in *DL-Lite$_{bool}$*. The result on the size of witness concepts for *DL-Lite$_{bool}$* is easily converted into a decision procedure for non-conservativity which is in Π_2^p: just (non-deterministically) guess a concept C of polynomial size in the size of $\mathcal{T}_1$ and $\mathcal{T}_2$ and with $\mathsf{sig}(C) \subseteq \Sigma$ and check, by calling an NP-oracle, whether (i) C is satisfiable w.r.t. $\mathcal{T}_1$ and (ii) not satisfiable w.r.t. $\mathcal{T}_1 \cup \mathcal{T}_2$. Because of the larger size of minimal witnesses, no such procedure exists for $\mathcal{EL}$ or $\mathcal{ALC}$.

Consider now the situation when the ontology designer is not only interested in preserving derivable concept inclusions, but also in preserving answers to queries, for both *DL-Lite$_{bool}$* and *DL-Lite$_{horn}$* TBoxes.

Definition 2 (query conservative extension). Let $\mathcal{T}_1$, $\mathcal{T}_2$ be $\mathcal{L}$-TBoxes and Σ a signature. We call $\mathcal{T}_1 \cup \mathcal{T}_2$ a *query conservative extension of* $\mathcal{T}_1$ *in* $\mathcal{L}$ *w.r.t.* Σ if, for every $\mathcal{L}$-ABox $\mathcal{A}$ with $\mathsf{sig}(\mathcal{A}) \subseteq \Sigma$, every $\mathcal{L}$-query q with $\mathsf{sig}(q) \subseteq \Sigma$, and every tuple $\boldsymbol{a}$ of object names from $\mathcal{A}$, we have $(\mathcal{T}_1, \mathcal{A}) \models q(\boldsymbol{a})$ whenever $(\mathcal{T}_1 \cup \mathcal{T}_2, \mathcal{A}) \models q(\boldsymbol{a})$.

It is easy to see that query conservativity implies deductive conservativity for both logics *DL-Lite$_{bool}$* and *DL-Lite$_{horn}$*. Indeed, let $\mathcal{L}$ be one of *DL-Lite$_{bool}$* and *DL-Lite$_{horn}$*. Suppose that we have $\mathcal{T}_1 \not\models C_1 \sqsubseteq C_2$ but $\mathcal{T}_1 \cup \mathcal{T}_2 \models C_1 \sqsubseteq C_2$, for some $\mathcal{L}$-concept inclusion $C_1 \sqsubseteq C_2$ with $\mathsf{sig}(C_1 \sqsubseteq C_2) \subseteq \Sigma$. Consider the ABox $\mathcal{A} = \{C_1(a)\}$ and the query $q = C_2(a)$. Then clearly $(\mathcal{T}_1 \cup \mathcal{T}_2, \mathcal{A}) \models q$, while $(\mathcal{T}_1, \mathcal{A}) \not\models q$. Note that in *DL-Lite*$_{horn}$, $C_1 = B_1 \sqcap \cdots \sqcap B_k$ and $C_2 = B$, where $B, B_1, \ldots, B_k$ are basic concepts.

The following example shows, in particular, that the converse implication does not hold.

Example 1. (1) To see that there are deductive conservative extensions which are not query conservative, take $\mathcal{T}_1 = \emptyset$, $\mathcal{T}_2 = \{A \sqsubseteq \exists P, \exists P^- \sqsubseteq B\}$ and $\Sigma = \{A, B\}$. Then $\mathcal{T}_2$ is a deductive conservative extension of $\mathcal{T}_1$ (in both *DL-Lite*$_{bool}$ and *DL-Lite*$_{horn}$) w.r.t. Σ. However, it is not a query conservative extension: let $\mathcal{A} = \{A(a)\}$ and $q = \exists y\, B(y)$; then $(\mathcal{T}_1, \mathcal{A}) \not\models q$ but $(\mathcal{T}_2, \mathcal{A}) \models q$.

(2) Note also that query conservativity in *DL-Lite*$_{horn}$ does not imply query conservativity in *DL-Lite*$_{bool}$. Indeed, let $\mathcal{T}_1 = \emptyset$, $\mathcal{T}_2 = \{A \sqsubseteq \exists P, A \sqcap \exists P^- \sqsubseteq \bot\}$ and $\Sigma = \{A\}$. Then $\mathcal{T}_2$ is not a query conservative extension of $\mathcal{T}_1$ in *DL-Lite*$_{bool}$ w.r.t. Σ: just take $\mathcal{A}$ as before and $q = \exists y\, \neg A(y)$. But it is a query conservative extension in *DL-Lite*$_{horn}$.

In the definition of essentially positive existential queries for *DL-Lite*$_{bool}$ above, we have allowed negated concepts in queries and ABoxes. An alternative approach would be to allow only *positive* concepts. These two types of queries give rise to different notions of query conservativity: under the second definition, the TBox $\mathcal{T}_2$ from Example 1 (2) is a query conservative extension of $\mathcal{T}_1 = \emptyset$ w.r.t. $\{A\}$, even in *DL-Lite*$_{bool}$. We argue, however, that it is the *essentially positive* queries that should be considered in the context of this investigation. The reason is that, with positive queries, the addition of the *definition* $B \equiv \neg A$ to $\mathcal{T}_2$ and B to Σ would result in a TBox which is not a query conservative extension in *DL-Lite*$_{bool}$ of $\mathcal{T}_1$ any longer. This kind of non-robust behaviour of the notion of conservativity is clearly undesirable. Obviously, the definitions we gave are robust under the addition of such definitions. Moreover, two extra robustness conditions hold true.

Definition 3 (robustness). Let Σ be a signature and cons_Σ a 'conservativity' relation between TBoxes w.r.t. Σ. (For example, $\mathsf{cons}_\Sigma(\mathcal{T}_1, \mathcal{T}_1 \cup \mathcal{T}_2)$ can be defined as '$\mathcal{T}_1 \cup \mathcal{T}_2$ is a deductive conservative extension of $\mathcal{T}_1$ in *DL-Lite*$_{bool}$ w.r.t. Σ').

- We say that cons_Σ is *robust under joins* if $(\mathcal{T}_0, \mathcal{T}_0 \cup \mathcal{T}_1), (\mathcal{T}_0, \mathcal{T}_0 \cup \mathcal{T}_2) \in \mathsf{cons}_\Sigma$ and $\mathsf{sig}(\mathcal{T}_1) \cap \mathsf{sig}(\mathcal{T}_2) \subseteq \Sigma$ imply $(\mathcal{T}_0, \mathcal{T}_0 \cup \mathcal{T}_1 \cup \mathcal{T}_2) \in \mathsf{cons}_\Sigma$;
- cons_Σ is *robust under vocabulary extensions* if $(\mathcal{T}_1, \mathcal{T}_1 \cup \mathcal{T}_2) \in \mathsf{cons}_\Sigma$ implies $(\mathcal{T}_1, \mathcal{T}_1 \cup \mathcal{T}_2) \in \mathsf{cons}_{\Sigma'}$, for all signatures Σ' with $\mathsf{sig}(\mathcal{T}_1 \cup \mathcal{T}_2) \cap \Sigma' \subseteq \Sigma$.

Roughly speaking, robustness under joins means that an ontology can be safely imported into joins of independent ontologies if each of them safely imports the ontology: if the shared symbols of $\mathcal{T}_1$ and $\mathcal{T}_2$ are from Σ, and both $\mathcal{T}_1 \cup \mathcal{T}_0$ and $\mathcal{T}_2 \cup \mathcal{T}_0$ are conservative extensions of $\mathcal{T}_0$ w.r.t. Σ, then the join $\mathcal{T}_1 \cup \mathcal{T}_2 \cup \mathcal{T}_0$ is a conservative extension of $\mathcal{T}_0$ w.r.t. Σ. In practice, this property supports collaborative ontology development in the following sense: it implies that if two (or more) ontology developers extend a given ontology $\mathcal{T}_0$ independently and do not use common symbols with the exception of those in a certain signature Σ then they can safely form the union of $\mathcal{T}_0$ and all their additional axioms provided that their individual extensions are safe for Σ (in the sense of deductive or, respectively, query conservativity). This property is closely related to the well-known *Robinson consistency lemma* and *interpolation* (see e.g., [10]) and has been investigated in the context of modular software specification [11] as well. We refer the reader to the Appendix for a more detailed discussion.

Robustness under vocabulary extensions is even closer to interpolation: it states that once we know conservativity w.r.t. Σ, we also know conservativity with respect to any signature with extra fresh symbols. The practical relevance of this property is as follows: when specifying the signature Σ for which an ontology developer wants to check conservativity, the developer only has to decide which symbols from $\mathcal{T}_1$ and $\mathcal{T}_2$ she wants to include into Σ. The answer to the query does not depend on whether Σ contains fresh symbols or not.

Theorem 2. *Both deductive and query conservativity in both DL-Lite$_{bool}$ and DL-Lite$_{horn}$ are robust under joins and vocabulary extensions.*

Actually, in *DL-Lite$_{horn}$* and *DL-Lite$_{bool}$* we even have a much stronger form of interpolation which is known as the *uniform interpolation property* [12, 13]. Let $\mathcal{T}$ be a TBox and Σ a signature. A TBox $\mathcal{T}'$ is called a *uniform interpolant* for $\mathcal{T}$ w.r.t. Σ in $\mathcal{L}$ if $\mathcal{T}'$ is an $\mathcal{L}$-TBox, $\mathsf{sig}(\mathcal{T}') \subseteq \Sigma$, $\mathcal{T} \models \mathcal{T}'$, and for all $\mathcal{L}$-concept inclusions $C_1 \sqsubseteq C_2$ with $\mathcal{T} \models C_1 \sqsubseteq C_2$ and $\mathsf{sig}(C_1, C_2) \cap \mathsf{sig}(\mathcal{T}) \subseteq \Sigma$, we have $\mathcal{T}' \models C_1 \sqsubseteq C_2$.

Intuitively, a uniform interpolant for $\mathcal{T}$ w.r.t. Σ contains exactly the same information about Σ in terms of concept inclusions as $\mathcal{T}$ *without using additional* symbols. For most DLs, such as $\mathcal{ALC}$, uniform interpolants do not necessarily exist [2].

Theorem 3. *Let $\mathcal{L}$ be DL-Lite$_{horn}$ or DL-Lite$_{bool}$. Then for every $\mathcal{L}$-TBox $\mathcal{T}$ and every signature Σ there exists a uniform interpolant for $\mathcal{T}$ w.r.t. Σ in $\mathcal{L}$.*

We note that one has to be very careful when interpreting the meaning of uniform interpolants. Consider, for instance, $\mathcal{T} = \{A \sqsubseteq \exists P,\ A \sqcap \exists P^- \sqsubseteq \bot\}$ and $\Sigma = \{A\}$. The TBox $\mathcal{T}' = \emptyset$ is a uniform interpolant of $\mathcal{T}$ w.r.t. Σ in *DL-Lite$_{bool}$*. However, as we saw in Example 1, the TBoxes $\mathcal{T}$ and $\mathcal{T}'$ behave differently with respect to queries in Σ: $(\mathcal{T}, \{A(a)\}) \models \exists x\, \neg A(x)$ but $(\mathcal{T}', \{A(a)\}) \not\models \exists x\, \neg A(x)$.

As sketched above, one application of deductive and query conservativity is to ensure that, when importing an ontology $\mathcal{T}$, one does not change the meaning of its vocabulary (in terms of concept inclusions or answers to queries). We now consider the situation where the ontology $\mathcal{T}$ to be imported is not known because, for example, it is still under development or because different ontologies can be chosen. In this case, $\mathcal{T}$ should be regarded as a 'black box' which supplies information about a signature Σ. To ensure that the meaning of the symbols in Σ as defined by this black box is not changed by importing it into $\mathcal{T}_1$, one has to check the following condition:

Definition 4 (safety). Let Σ be a signature and $\mathcal{T}_1$ an $\mathcal{L}$-TBox. We say that $\mathcal{T}_1$ is *safe for* Σ *w.r.t. deductive* (*or query*) *conservativity in* $\mathcal{L}$ if, for all $\mathcal{L}$-TBoxes $\mathcal{T}$ with $\mathsf{sig}(\mathcal{T}) \cap \mathsf{sig}(\mathcal{T}_1) \subseteq \Sigma$, $\mathcal{T}_1 \cup \mathcal{T}$ is a deductive (respectively, query) conservative extension of $\mathcal{T}$ in $\mathcal{L}$ w.r.t. Σ.

This notion has been introduced in [3] where the reader can find further discussion. A natural generalisation of safety, considered in [5], is the following property:

Definition 5 (strong deductive/query conservative extension). Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be $\mathcal{L}$-TBoxes and Σ a signature. We call $\mathcal{T}_1 \cup \mathcal{T}_2$ a *strong deductive* (*query*) *conservative extension* of $\mathcal{T}_1$ in $\mathcal{L}$ w.r.t. Σ if $\mathcal{T}_1 \cup \mathcal{T}_2 \cup \mathcal{T}$ is a deductive (respectively, query) conservative extension of $\mathcal{T}_1 \cup \mathcal{T}$ in $\mathcal{L}$ w.r.t. Σ, for every $\mathcal{L}$-TBox $\mathcal{T}$ with $\mathsf{sig}(\mathcal{T}) \cap \mathsf{sig}(\mathcal{T}_1 \cup \mathcal{T}_2) \subseteq \Sigma$.

Observe that safety is indeed a special case of strong conservativity: it covers exactly the case where the TBox $\mathcal{T}_1$ in the definition of strong conservativity is empty. A typical application of strong conservativity for ontology re-use is as follows (see [5]). Suppose that there is a large ontology $\mathcal{O}$ and a subset Σ of its signature. Assume also that the ontology designer wants to use what $\mathcal{O}$ says about Σ in her own ontology $\mathcal{T}$ she is developing at the moment. Then instead of importing $\mathcal{O}$ as a whole, it would be preferable to find a small subset $\mathcal{T}_1$ of $\mathcal{O}$, which says precisely the same about Σ as $\mathcal{O}$ does, and import only this small $\mathcal{T}_1$ rather than the large $\mathcal{O}$. But then what are the conditions we should impose on $\mathcal{T}_1$? An obvious minimal requirement is that by importing $\mathcal{T}_1$ into $\mathcal{T}$ we obtain the same consequences for subsumptions/queries over Σ as by importing $\mathcal{O}$ into $\mathcal{T}$. Depending on whether concept inclusions or answers to queries are of interest, one therefore wants $\mathcal{O} = \mathcal{T}_1 \cup \mathcal{T}_2$ to be a strong deductive or query conservative extension of $\mathcal{T}_1$ w.r.t Σ. We refer the reader to [5] for further discussion and algorithms for extracting such TBoxes from a given TBox.

Example 2. (1) Let us see first that strong deductive conservativity is indeed a stronger notion than deductive conservativity, for *DL-Lite*$_{bool}$ and *DL-Lite*$_{horn}$. Consider again the TBoxes $\mathcal{T}_1 = \emptyset$, $\mathcal{T}_2 = \{A \sqsubseteq \exists R, \quad A \sqcap \exists R^- \sqsubseteq \bot\}$, and $\Sigma = \{A\}$. Then $\mathcal{T}_1 \cup \mathcal{T}_2$ is a deductive conservative extension of $\mathcal{T}_1$ w.r.t. Σ. However, $\mathcal{T}_1 \cup \mathcal{T}_2$ is *not* a strong deductive conservative extension of $\mathcal{T}_1$ w.r.t. Σ. Let $\mathcal{T} = \{\top \sqsubseteq A\}$. Then we have $\mathcal{T}_1 \cup \mathcal{T}_2 \cup \mathcal{T} \models \top \sqsubseteq \bot$ but $\mathcal{T}_1 \cup \mathcal{T} \not\models \top \sqsubseteq \bot$.

(2) We show now that an analogue of Theorem 1 does not hold for strong deductive conservativity. Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be the following *DL-Lite*$_{horn}$ TBoxes

$$\begin{aligned}
\mathcal{T}_1 &= \big\{A \sqcap B \sqsubseteq \bot, \quad \top \sqsubseteq \exists P_1, \quad \top \sqsubseteq \exists P_2, \quad \exists P_1^- \sqsubseteq A, \quad \exists P_2^- \sqsubseteq B\big\},\\
\mathcal{T}_2 &= \big\{\top \sqsubseteq \exists R, \quad A \sqcap \exists R^- \sqsubseteq \bot, \quad B \sqcap \exists R^- \sqsubseteq \bot\big\},
\end{aligned}$$

and let $\Sigma = \{A, B, P_1, P_2\}$. $\mathcal{T}_2$ says that $\top \not\sqsubseteq A \sqcup B$. Now, in *DL-Lite*$_{bool}$, $\mathcal{T}_1 \cup \mathcal{T}_2$ is not a strong deductive conservative extension of $\mathcal{T}_1$ w.r.t. Σ: just take $\mathcal{T} = \{\top \sqsubseteq A \sqcup B\}$. However, $\mathcal{T}_1 \cup \mathcal{T}_2$ is a strong deductive conservative extension of $\mathcal{T}_1$ in *DL-Lite*$_{horn}$.

Obviously, the robustness conditions introduced above are of importance for the strong versions of conservativity as well.

Theorem 4. *Both strong deductive and strong query conservativity are robust under joins and vocabulary extensions for DL-Lite$_{bool}$ and DL-Lite$_{horn}$.*

In addition to these types of robustness, the following condition, which is dual to robustness under joins, is of crucial importance for iterated applications of the notion of safety for a signature. Suppose that $\mathcal{T}$ is safe for $\Sigma_1 \cup \Sigma_2$ under some notion of conservativity and $\Sigma_1 \cap \Sigma_2 = \emptyset$. Then, for any $\mathcal{T}_1$ with $\mathsf{sig}(\mathcal{T}_1) \cap (\mathsf{sig}(\mathcal{T}) \cup \Sigma_2) \subseteq \Sigma_1$, the TBox $\mathcal{T} \cup \mathcal{T}_1$ should be safe for Σ_2 for the same notion of conservativity. Without this property, one might have the situation that a TBox is safe for a signature $\Sigma_1 \cup \Sigma_2$, but after importing a TBox for Σ_1 the resulting TBox is not safe for Σ_2 any longer, which is clearly undesirable.

Theorem 5 (robustness under joins of signatures). *Let $\mathcal{L}$ be either DL-Lite$_{bool}$ or DL-Lite$_{horn}$. If an $\mathcal{L}$-TBox $\mathcal{T}$ is safe for a signature $\Sigma_1 \cup \Sigma_2$ w.r.t. deductive/query conservativity in $\mathcal{L}$, $\Sigma_1 \cap \Sigma_2 = \emptyset$ and $\mathcal{T}_1$ is a satisfiable $\mathcal{L}$-TBox with $\mathsf{sig}(\mathcal{T}_1) \cap (\mathsf{sig}(\mathcal{T}) \cup \Sigma_2) \subseteq \Sigma_1$, then $\mathcal{T} \cup \mathcal{T}_1$ is safe for Σ_2 w.r.t. deductive/query-conservativity in $\mathcal{L}$.*

This result follows immediately from the fact that any two satisfiable $\mathcal{L}$-TBoxes in disjoint signatures are strong query conservative extensions of each other. This property fails for a number of stronger notions of conservativity, for example, model conservativity.

The next theorem shows that in all those cases where we have not provided counterexamples the introduced notions of conservativity are equivalent:

DL-Lite$_{horn}$	deductive	$\lneqq$	query	$\lneqq$	strong deductive	$\equiv$	strong query
DL-Lite$_{bool}$	deductive	$\lneqq$	query	$\equiv$	strong deductive	$\equiv$	strong query

It also establishes the complexity of the corresponding decision problems.

Theorem 6. *Let $\mathcal{L}$ be either DL-Lite$_{bool}$ or DL-Lite$_{horn}$, $\mathcal{T}_1$ and $\mathcal{T}_2$ $\mathcal{L}$-TBoxes, and Σ a signature.*

For $\mathcal{L}$ = DL-Lite$_{bool}$, the following conditions are equivalent:

(1) *$\mathcal{T}_1 \cup \mathcal{T}_2$ is a query conservative extension of $\mathcal{T}_1$ in $\mathcal{L}$ w.r.t. Σ;*
(2) *$\mathcal{T}_1 \cup \mathcal{T}_2$ is a strong deductive conservative extension of $\mathcal{T}_1$ in $\mathcal{L}$ w.r.t. Σ;*
(3) *$\mathcal{T}_1 \cup \mathcal{T}_2$ is a strong query conservative extension of $\mathcal{T}_1$ in $\mathcal{L}$ w.r.t. Σ.*

For $\mathcal{L}$ = DL-Lite$_{horn}$, conditions (2) *and* (3) *are equivalent, while* (1) *is strictly weaker than each of them.*

For DL-Lite$_{horn}$, the decision problems corresponding to conditions (1)–(3) *are all* CONP*-complete. For DL-Lite$_{bool}$, these problems are Π_2^p-complete.*

We believe that the equivalences stated in Theorem 6 are somewhat surprising. For example, it can be easily seen that for $\mathcal{ALC}$ none of those equivalences holds true.

4 Model-Theoretic Characterisations of Conservativity

All the results discussed above are proved with the help of the model-theoretic characterisations of our notions of conservativity formulated below.

Let Σ be a signature and Q a set of positive natural numbers containing 1. By a *ΣQ-concept* we mean any concept of the form $\bot$, $\top$, A_i, $\geq q\, R$, or its negation for some $A_i \in \Sigma$, Σ-role R and $q \in Q$. A *ΣQ-type* is a set $\boldsymbol{t}$ of ΣQ-concepts containing $\top$ such that the following conditions hold:

- for every ΣQ-concept C, either $C \in \boldsymbol{t}$ or $\neg C \in \boldsymbol{t}$,
- if $q < q'$ are both in Q and $\geq q'\, R \in \boldsymbol{t}$ then $\geq q\, R \in \boldsymbol{t}$,
- if $q < q'$ are both in Q and $\neg(\geq q\, R) \in \boldsymbol{t}$ then $\neg(\geq q'\, R) \in \boldsymbol{t}$.

It should be clear that, for each ΣQ-type $\boldsymbol{t}$ with $\bot \notin \boldsymbol{t}$, there is an interpretation $\mathcal{I}$ and a point x in it such that, for every $C \in \boldsymbol{t}$, we have $x \in C^{\mathcal{I}}$. In this case we say that $\boldsymbol{t}$ is *realised at x in $\mathcal{I}$*, or that $\boldsymbol{t}$ is the *ΣQ-type of x in $\mathcal{I}$*.

Definition 6. A set Ξ of ΣQ-types is said to be *$\mathcal{T}$-realisable* if there is a model for $\mathcal{T}$ realising *all* types from Ξ. We also say that Ξ is *precisely $\mathcal{T}$-realisable* if there is a model $\mathcal{I}$ for $\mathcal{T}$ such that $\mathcal{I}$ realises all types in Ξ and every ΣQ-type realised in $\mathcal{I}$ is in Ξ.

Given a signature Σ, we say that interpretations $\mathcal{I}$ and $\mathcal{J}$ are *Σ-isomorphic* and write $\mathcal{I} \sim_\Sigma \mathcal{J}$ if there is a bijection $f\colon \Delta^{\mathcal{I}} \to \Delta^{\mathcal{J}}$ such that $f(a^{\mathcal{I}}) = a^{\mathcal{J}}$, for every object name a, $x \in A^{\mathcal{I}}$ iff $f(x) \in A^{\mathcal{J}}$, for every concept name $A \in \Sigma$, and $(x, y) \in P^{\mathcal{I}}$ iff $(f(x), f(y)) \in P^{\mathcal{J}}$, for every role name $P \in \Sigma$. Clearly, Σ-isomorphic interpretations cannot be distinguished by TBoxes, ABoxes, or queries *over* Σ.

Given a set $\mathcal{I}_i$, $i \in I$, of interpretations with $1 \in I$, define the interpretation (the *disjoint union* of the $\mathcal{I}_i$) $\mathcal{J} = \bigoplus_{i \in I} \mathcal{I}_i$, where $\Delta^{\mathcal{J}} = \{(i, w) \mid i \in I, w \in \Delta_i\}$, $a^{\mathcal{J}} = (1, a^{\mathcal{I}_1})$, for every object name a, $A^{\mathcal{J}} = \{(i, w) \mid w \in A^{\mathcal{I}_i}\}$, for every concept name A, and $P^{\mathcal{J}} = \{((i, w_1), (i, w_2)) \mid (w_1, w_2) \in P^{\mathcal{I}_i}\}$, for every role name P. Given an interpretation $\mathcal{I}$, we set $\mathcal{I}^\omega = \bigoplus_{i \in \omega} \mathcal{I}_i$, where $\mathcal{I}_i = \mathcal{I}$ for $i \in \omega$. Again, it should be clear that TBoxes, ABoxes or queries (over any signature) cannot distinguish between $\mathcal{I}$ and $\mathcal{I}^\omega$.

The following lemma provides an important model-theoretic property of *DL-Lite*$_{bool}$ which is used to establish model-theoretic characterisations of various notions of conservativity.

Lemma 1. *Let $\mathcal{J}$ be an (at most countable) model for $\mathcal{T}_1$ and Σ a signature with $\Sigma \subseteq \mathsf{sig}(\mathcal{T}_1)$. Suppose that there is a model for $\mathcal{T}_1 \cup \mathcal{T}_2$ realising precisely the same $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types as $\mathcal{J}$, where $Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$ is the set of all numerical parameters occurring in $\mathcal{T}_1 \cup \mathcal{T}_2$ together with 1. Then there is a model $\mathcal{I}^*$ for $\mathcal{T}_1 \cup \mathcal{T}_2$ such that $\mathcal{I}^* \sim_\Sigma \mathcal{J}^\omega$.*

In particular, $\mathcal{I}^* \models \mathcal{A}$ iff $\mathcal{J} \models \mathcal{A}$, for all ABoxes $\mathcal{A}$ over Σ, $\mathcal{I}^* \models \mathcal{T}$ iff $\mathcal{J} \models \mathcal{T}$, for all TBoxes $\mathcal{T}$ over Σ, and $\mathcal{I}^* \models q(\boldsymbol{a})$ iff $\mathcal{J} \models q(\boldsymbol{a})$, for all queries $q(\boldsymbol{a})$ over Σ.

In the case of *DL-Lite$_{horn}$* we need some extra definitions. By a *ΣQ^h-concept* we mean any concept of the form $\bot$, A_i or $\geq q\, R$, for some $A_i \in \Sigma$, Σ-role R and $q \in Q$. Given a ΣQ-type $\boldsymbol{t}$, let $\boldsymbol{t}^+ = \{B \in \boldsymbol{t} \mid B \text{ a } \Sigma Q^h\text{-concept}\}$. Say that a ΣQ-type $\boldsymbol{t}_1$ is *h-contained* in a ΣQ-type $\boldsymbol{t}_2$ if $\boldsymbol{t}_1^+ \subseteq \boldsymbol{t}_2^+$. The following two notions characterise conservativity for *DL-Lite$_{horn}$*:

Definition 7. A set Ξ of ΣQ-types is said to be *sub-precisely $\mathcal{T}$-realisable* if there is a model $\mathcal{I}$ for $\mathcal{T}$ such that $\mathcal{I}$ realises all types from Ξ, and every ΣQ-type realised in $\mathcal{I}$ is h-contained in some type from Ξ. We also say that Ξ is *join-precisely $\mathcal{T}$-realisable* if there is a model $\mathcal{I}$ for $\mathcal{T}$ such that, for every ΣQ-type $\boldsymbol{t}$ realised in $\mathcal{I}$, $\Xi_{\boldsymbol{t}} = \{\boldsymbol{t}_i \in \Xi \mid \boldsymbol{t}^+ \subseteq \boldsymbol{t}_i^+\} \neq \emptyset$ and $\boldsymbol{t}^+ = \bigcap_{\boldsymbol{t}_i \in \Xi_{\boldsymbol{t}}} \boldsymbol{t}_i^+$. (It follows that $\boldsymbol{t}^+ \subseteq \boldsymbol{t}_i^+$, for all $\boldsymbol{t}_i \in \Xi_{\boldsymbol{t}}$, and thus, Ξ is sub-precisely $\mathcal{T}$-realisable.)

The table below gives characterisations of our four notions of conservativity in the following form: let Σ be a signature and $\mathcal{L}$ be either *DL-Lite$_{bool}$* or *DL-Lite$_{horn}$*; then $\mathcal{T}_1 \cup \mathcal{T}_2$ is an α conservative extension of $\mathcal{T}_1$ w.r.t. Σ in $\mathcal{L}$ iff every precisely $\mathcal{T}_1$-realisable set Ξ of $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$ types is '... $\mathcal{T}_1 \cup \mathcal{T}_2$-realisable' ($Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$ is the set of numerical parameters occurring in $\mathcal{T}_1 \cup \mathcal{T}_2$ together with 1).

<table>
<tr><th rowspan="2">conservativity α</th><th colspan="2">language $\mathcal{L}$</th></tr>
<tr><th>DL-Lite$_{bool}$</th><th>DL-Lite$_{horn}$</th></tr>
<tr><td>deductive</td><td>$\mathcal{T}_1 \cup \mathcal{T}_2$-realisable</td><td>$\mathcal{T}_1 \cup \mathcal{T}_2$-realisable</td></tr>
<tr><td>query</td><td rowspan="3">precisely
$\mathcal{T}_1 \cup \mathcal{T}_2$-realisable</td><td>sub-precisely
$\mathcal{T}_1 \cup \mathcal{T}_2$-realisable</td></tr>
<tr><td>strong deductive</td><td rowspan="2">join-precisely
$\mathcal{T}_1 \cup \mathcal{T}_2$-realisable</td></tr>
<tr><td>strong query</td></tr>
</table>

These characterisations are proved in the Appendix, where it is also shown that in each case it suffices to consider only those sets Ξ the size of which is bounded by a polynomial function in the size of the TBoxes. Then, for *DL-Lite$_{bool}$* TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$, one can decide whether $\mathcal{T}_1 \cup \mathcal{T}_2$ is a *not* a strong deductive conservative extension by (non-deterministically) guessing a polynomial set Ξ of $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types and checking that it is precisely $\mathcal{T}_1$-realisable and *not* precisely $\mathcal{T}_1 \cup \mathcal{T}$-realisable. The Appendix provides a polynomial *non-deterministic* algorithm deciding whether a given set of ΣQ-types is precisely $\mathcal{T}$-realisable, which yields a Π_2^p upper bound for strong deductive conservativity in *DL-Lite$_{bool}$*. Similarly, for *DL-Lite$_{horn}$* TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$, one can decide whether $\mathcal{T}_1 \cup \mathcal{T}_2$ is a query (or strong deductive) conservative extension of $\mathcal{T}_1$ by guessing Ξ and checking that it is precisely $\mathcal{T}_1$-realisable and *not* sub-precisely (respectively, join-precisely) $\mathcal{T}_1 \cup \mathcal{T}_2$-realisable. The Appendix provides polynomial *deterministic* algorithms deciding whether a given set of ΣQ-types is precisely, sub-precisely and join-precisely $\mathcal{T}$-realisable, for a *DL-Lite$_{horn}$* TBox $\mathcal{T}$, which give CONP upper bounds for query and strong deductive conservativity. The lower bounds follow immediately from the corresponding lower bounds for propositional logic.

5 Conclusion

We have analysed the relation between different notions of conservative extension in description logics $DL\text{-}Lite_{bool}$ and $DL\text{-}Lite_{horn}$, and proved that the corresponding reasoning problems are not harder than the same problems in propositional logic. Moreover, we have also shown that important meta-properties for modular ontology engineering, such as robustness under joins, vocabulary extensions, and iterated import of ontologies, hold true for these notions of conservativity.

Acknowledgements. This work was partially supported by U.K. EPSRC grant GR/S63175.

References

1. Antoniou, G., Kehagias, A.: A note on the refinement of ontologies. Int. J. of Intelligent Systems **15**(7) (2000) 623–632
2. Ghilardi, S., Lutz, C., Wolter, F.: Did I damage my ontology? A case for conservative extensions in description logic. In: Proc. of KR 2006, AAAI Press (2006) 187–197
3. Grau, B.C., Horrocks, I., Kazakov, Y., Sattler, U.: A logical framework for modularity of ontologies. In: Proc. of IJCAI 2007. (2007) 298–303
4. Lutz, C., Walther, D., Wolter, F.: Conservative extensions in expressive description logics. In: Proc. of IJCAI 2007. (2007) 453–458
5. Grau, B.C., Horrocks, I., Kazakov, Y., Sattler, U.: Just the right amount: Extracting modules from ontologies. In: Proc. of the 16th International World Wide Web Conference (WWW-2007). (2007)
6. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: DL-Lite: Tractable description logics for ontologies. In: Proc. of AAAI 2005. (2005) 602–607
7. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Data complexity of query answering in description logics. In: Proc. of KR 2006. (2006) 260–270
8. Artale, A., Calvanese, D., Kontchakov, R., Zakharyaschev, M.: DL-Lite in the light of first-order logic. In: Proc. of AAAI 2007. (2007)
9. Lutz, C., Wolter, F.: Conservative extensions in the lightweight description logic EL. To appear in the Proceedings of 21st Conference on Automated Deduction (CADE-21) (2007)
10. Chang, C., Keisler, H.: Model Theory. Elsevier (1990)
11. Diaconescu, R., Goguen, J., Stefaneas, P.: Logical support for modularisation. In Huet, G., Plotkin, G., eds.: Logical Environments, Cambridge University Press, New York (1993) 83–130
12. Pitts, A.: On an interpretation of second-order quantification in first-order intuitionistic propositional logic. J. Symbolic Logic **57**(1) (1992) 33–52
13. Visser, A.: Uniform interpolation and layered bisimulation. In Hájek, P., ed.: Gödel '96 (Brno, 1996). Volume 6 of Lecture Notes in Logic. Springer Verlag (1996) 139–164
14. Papadimitriou, C.: Computational Complexity. Addison Wesley Publ. Co. (1994)