Can you tell the difference between *DL-Lite* ontologies?

Roman Kontchakov,[1] Frank Wolter[2] and Michael Zakharyaschev[1]
[1]School of Computer Science and Information Systems, Birkbeck College London
[2]Department of Computer Science, University of Liverpool

Abstract

We develop a formal framework for comparing different versions of *DL-Lite* ontologies. Four distinct notions of difference and entailment between ontologies are introduced and their applications in ontology development and maintenance discussed. These notions are obtained by distinguishing between differences that can be observed among concept inclusions, answers to queries over ABoxes, and by taking into account additional context ontologies. We compare these notions, study their meta-properties, and determine the computational complexity of the corresponding reasoning tasks. Moreover, we show that checking difference and entailment can be automated by means of encoding into QBF satisfiability and using off-the-shelf QBF solvers.

Introduction

In computer science, ontologies are used to provide a common vocabulary for a domain of interest, together with a description of the relationships between terms built from the vocabulary. Ontology languages based on description logics (DLs) represent ontologies as TBoxes (terminological boxes) containing inclusions between complex concepts over the vocabulary. An increasingly important application of ontologies is management of large amounts of data, where ontologies are used to provide flexible and efficient access to repositories consisting of data sets of instances of concepts and relations. In DLs, such repositories are typically modelled as ABoxes (sets of assertions).

Developing and maintaining 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 tasks. 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 already existing ontologies. In all those cases, reasoning support for analysing the impact of the respective operation on the ontology would be highly desirable. Typical examples of such 'unorthodox' reasoning services include the following:

- *Comparing versions of ontologies.* The standard `diff` utility is an indispensable tool for comparing files. However, such a purely syntactic operation is of little value if the files contain different versions of ontologies (Noy & Musen 2002) because our concern now is not the syntactic form of their axioms, but the differences between relationships between terms over their common vocabulary Σ these ontologies *imply*. The reasoning service we need in this case is to compare the logical consequences of different versions of ontologies over Σ.
- *Ontology refinement.* When refining an ontology by adding new axioms, one usually wants to preserve the relationships between terms of a certain part Σ of its vocabulary. The reasoning service required in such a case is to check whether the refined ontology has precisely the same logical consequences over Σ as the original one.
- *Ontology re-use.* When importing an ontology, one wants to *use* its vocabulary Σ as originally defined. However, relationships between terms over Σ may change due to some axioms in the importing ontology. So, again, we need a reasoning service capable of checking whether new logical consequences over Σ are derivable (this service has been termed *safety checking* in (Grau *et al.* 2007b)).

In all these and many other cases, we are interested in comparing the relationships between terms over some vocabulary (or *signature*) Σ two ontologies imply. This gives rise to the two main notions we investigate in this paper: Σ-*difference* and Σ-*entailment*. Roughly, the Σ-difference between two ontologies is the set of 'formulas' over Σ that are derivable from one ontology but not from the other; and one ontology Σ-entails another if all Σ-formulas derivable from the latter are also derivable from the former.

A very important special case of Σ-entailment, namely various versions of the notion of *conservative extension*, has been intensively investigated in the past few years (Antoniou & Kehagias 2000; Ghilardi, Lutz, & Wolter 2006; Grau *et al.* 2007a; 2007b; Lutz, Walther, & Wolter 2007). In this case one ontology is included in the other and Σ is the vocabulary of the smaller one. The Σ-formulas considered in these papers were concept inclusions $C_1 \sqsubseteq C_2$, and a number of complexity and decidability results were obtained.[1] Also, model conservativity (Lutz, Walther, & Wolter

[1]The complexity and decidability results for conservative extensions obtained in (Ghilardi, Lutz, & Wolter 2006; Lutz, Walther, & Wolter 2007) can easily be generalised to Σ-entailment.

2007) and sufficient syntactic conditions of conservativity, e.g., locality (Grau *et al.* 2007b), have been considered.

In this paper we also deal with concept inclusions, but more importantly, we analyse Σ-difference and entailment with respect to existential Σ-*queries*, where the reasoning task is to decide whether two ontologies give precisely the same answers to Σ-queries for *any* database (= ABox) over Σ, and perhaps *any* additional context ontology over Σ. The corresponding notions of Σ-query difference and entailment are of interest for any DL, but they are of particular importance to those DLs that were specifically designed in order to facilitate efficient query-answering over large data sets.

The idea of using ontologies as a conceptual view over data repositories goes back to (Borgida *et al.* 1989) and has recently been developed to a quite practical level (Acciarri *et al.* 2005; Calvanese *et al.* 2007) with promising applications in such areas as data integration and P2P data management. The *DL-Lite* family of description logics has been largely designed with this application in mind (Calvanese *et al.* 2005; 2006). The data complexity of query answering is within LOGSPACE for most members of the family, and moreover, queries over *DL-Lite* ontologies can be rewritten as SQL queries so that standard database query engines can be used. *DL-Lite* is part of the OWL 1.1 Web Ontology Language which is a W3C Member Submission.

In this paper, we investigate four notions of Σ-difference and Σ-entailment for two members of the *DL-Lite* family: *DL-Lite*$_{bool}$, the most expressive language of the family, basically covering all others, and *DL-Lite*$_{horn}$, the Horn subset of *DL-Lite*$_{bool}$. The four notions of Σ-difference and entailment are obtained by distinguishing between differences visible among concept inclusions, answers to queries over ABoxes, and by taking into account additional context ontologies. We compare these notions, study their meta-properties, and determine the computational complexity of the corresponding reasoning tasks. Moreover, we show that the reasoning services discussed above can be implemented by means of encoding into satisfiability of quantified Boolean formulas (QBF). We report on our first experiments with general purpose off-the-shelf QBF solvers for deciding Σ-entailment between 'typical' *DL-Lite* ontologies.

The *DL-Lite* Family

We remind the reader of the syntax and semantics of the DLs *DL-Lite*$_{bool}$ and *DL-Lite*$_{horn}$ introduced and investigated in (Calvanese *et al.* 2005; 2006; Artale *et al.* 2007). The language of *DL-Lite*$_{bool}$ has *object names* $a_1, a_2, \dots$, *concept names* $A_1, A_2, \dots$, and *role names* $P_1, P_2, \dots$. 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 *concept inclusion* in *DL-Lite*$_{bool}$ is of the form $C_1 \sqsubseteq C_2$, where C_1 and C_2 are *DL-Lite*$_{bool}$ concepts. (Other concept constructs like $\exists R$, $\leq q\, R$ and $C_1 \sqcup C_2$ will be used as standard abbreviations.) A *TBox in DL-Lite*$_{bool}$, $\mathcal{T}$, is a finite set of concept inclusions in *DL-Lite*$_{bool}$.

In the *Horn* fragment *DL-Lite*$_{horn}$ of *DL-Lite*$_{bool}$, *concept inclusions* are restricted to 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 inclusions $\bigsqcap_k B_k \sqsubseteq \bot$ 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. 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 $B \sqcap \geq 2\, R \sqsubseteq \bot$.

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

An *interpretation* $\mathcal{I}$ is a structure of the form $(\Delta^{\mathcal{I}}, A_1^{\mathcal{I}}, \dots, P_1^{\mathcal{I}}, \dots, a_1^{\mathcal{I}}, \dots)$, where $\Delta^{\mathcal{I}}$ is nonempty, $A_i^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, $P_i^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ and $a_i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ with $a_i^{\mathcal{I}} \neq a_j^{\mathcal{I}}$, 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, e.g.,

$$d \in (\geq q\, R)^{\mathcal{I}} \quad \text{iff} \quad |\{d' \in \Delta^{\mathcal{I}} \mid (d, d') \in R^{\mathcal{I}}\}| \geq q.$$

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 $\mathcal{I}$. An ABox assertion $C(a)$ ($R(a_i, a_j)$) is satisfied in $\mathcal{I}$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$ ($(a_i^{\mathcal{I}}, a_j^{\mathcal{I}}) \in R^{\mathcal{I}}$). A concept inclusion $C_1 \sqsubseteq C_2$ *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 a 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 query in* $\mathcal{L}$ (or simply a *query*, if $\mathcal{L}$ is understood) is a first-order formula

$$q(x_1, \dots, x_n) \;=\; \exists y_1 \dots \exists y_m \varphi(x_1, \dots, x_n, y_1, \dots, y_m),$$

where φ is constructed, using only $\wedge$ and $\vee$, from atoms of the form $C(t)$ and $R(t_1, t_2)$, with C being an $\mathcal{L}$-concept, R a role, and t_i being either an object name or a variable from the list $x_1, \dots, x_n, y_1, \dots, y_m$. Given a KB $\mathcal{K}$ and a query $q(\boldsymbol{x})$, $\boldsymbol{x} = x_1, \dots, x_n$, we say that an n-tuple $\boldsymbol{a}$ of object names is a *certain answer* to $q(\boldsymbol{x})$ w.r.t. $\mathcal{K}$ and write $\mathcal{K} \models q(\boldsymbol{a})$ if, for every model $\mathcal{I}$ for $\mathcal{K}$, we have $\mathcal{I} \models q(\boldsymbol{a})$. The subsumption problem '$\mathcal{T} \models C_1 \sqsubseteq C_2$?' is coNP-complete in *DL-Lite*$_{bool}$ and P-complete in *DL-Lite*$_{horn}$; the data complexity of the query answering problem for *DL-Lite*$_{horn}$ KBs is in LOGSPACE, while for *DL-Lite*$_{bool}$ it is CONP-complete (Artale *et al.* 2007).

What is the Difference?

As we saw in the introduction, the notions of difference and entailment between ontologies are restricted to some *signature*, i.e., a finite set of concept and role names.[2] Given a concept, role, concept inclusion, TBox, ABox, or query E, we denote by *sig*(E) the *signature* of E, that is, the

[2] As *DL-Lite*$_{bool}$ TBoxes do not contain object names, we do not have to include them to signatures (unlike DLs with nominals).

set of concept and role names that occur in E. It is to be noted that $\bot$ and $\top$ are regarded as logical symbols, and so $sig(\bot) = sig(\top) = \emptyset$. A concept (role, concept inclusion, TBox, ABox, query) E is called a Σ-*concept* (*role*, *concept inclusion*, *TBox*, *ABox*, *query*, respectively) if $sig(E) \subseteq \Sigma$. Thus, P^- is a Σ-role iff $P \in \Sigma$.

Definition 1 Let $\mathcal{L} \in \{\textit{DL-Lite}_{bool}, \textit{DL-Lite}_{horn}\}$ and let $\mathcal{T}_1$ and $\mathcal{T}_2$ be TBoxes in $\mathcal{L}$ and Σ a signature.

- The Σ-*concept difference* between $\mathcal{T}_1$ and $\mathcal{T}_2$ is the set $\mathsf{cDiff}^{\mathcal{L}}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2)$ of all Σ-concept inclusions $C \sqsubseteq D$ in $\mathcal{L}$ such that $\mathcal{T}_2 \models C \sqsubseteq D$ and $\mathcal{T}_1 \not\models C \sqsubseteq D$. We say that $\mathcal{T}_1$ Σ-*concept entails* $\mathcal{T}_2$ *in* $\mathcal{L}$ if $\mathsf{cDiff}^{\mathcal{L}}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2) = \emptyset$.
- The Σ-*query difference* between $\mathcal{T}_1$ and $\mathcal{T}_2$ is the set $\mathsf{qDiff}^{\mathcal{L}}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2)$ of pairs $(\mathcal{A}, q(\boldsymbol{x}))$, where $\mathcal{A}$ is a Σ-ABox in $\mathcal{L}$ and $q(\boldsymbol{x})$ a Σ-query in $\mathcal{L}$ such that $(\mathcal{T}_1, \mathcal{A}) \not\models q(\boldsymbol{a})$ and $(\mathcal{T}_2, \mathcal{A}) \models q(\boldsymbol{a})$, for some tuple $\boldsymbol{a}$ of object names from $\mathcal{A}$. We say that $\mathcal{T}_1$ Σ-*query entails* $\mathcal{T}_2$ *in* $\mathcal{L}$ if $\mathsf{qDiff}^{\mathcal{L}}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2) = \emptyset$.
- The *strong* Σ-*concept difference* between $\mathcal{T}_1$ and $\mathcal{T}_2$ is the set $\mathsf{scDiff}^{\mathcal{L}}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2)$ of all pairs $(\mathcal{T}, C \sqsubseteq D)$ such that $\mathcal{T}$ is a Σ-TBox in $\mathcal{L}$ and $C \sqsubseteq D \in \mathsf{cDiff}^{\mathcal{L}}_{\Sigma}(\mathcal{T} \cup \mathcal{T}_1, \mathcal{T} \cup \mathcal{T}_2)$. $\mathcal{T}_1$ *strongly* Σ-*concept entails* $\mathcal{T}_2$ *in* $\mathcal{L}$ if $\mathsf{scDiff}^{\mathcal{L}}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2) = \emptyset$.
- The *strong* Σ-*query difference* between $\mathcal{T}_1$ and $\mathcal{T}_2$ is the set $\mathsf{sqDiff}^{\mathcal{L}}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2)$ of all triples $(\mathcal{T}, \mathcal{A}, q(\boldsymbol{x}))$ such that $\mathcal{T}$ is a Σ-TBox in $\mathcal{L}$ and $(\mathcal{A}, q(\boldsymbol{x})) \in \mathsf{qDiff}^{\mathcal{L}}_{\Sigma}(\mathcal{T} \cup \mathcal{T}_1, \mathcal{T} \cup \mathcal{T}_2)$. We also say that $\mathcal{T}_1$ *strongly* Σ-*query entails* $\mathcal{T}_2$ *in* $\mathcal{L}$ if $\mathsf{sqDiff}^{\mathcal{L}}_{\Sigma}(\mathcal{T}_1, \mathcal{T}_2) = \emptyset$.

As argued in the introduction, the notions of Σ-difference and Σ–entailment can play an important role in comparing ontologies, checking whether a refinement of an ontology has undesirable effects on a certain part of its signature, and in checking whether a re-used (imported) ontology changes when put into the environment of another ontology. In all those cases, Σ indicates the vocabulary over which the user wants to compare ontologies. For example, for two versions of a medical ontology, a user interested in anatomy might choose Σ to be the set of terms relevant to anatomy and then check whether the two ontologies differ w.r.t. these terms.

In the definition of Σ-query difference, we take into account *arbitrary* Σ-ABoxes in $\mathcal{L}$. The reason is that during the ontology design phase, the data repositories to which the ontology will be applied are often either completely unknown or are subject to more or less frequent changes. Thus, to assume that we have a fixed ABox is unrealistic when checking differences between ontologies, and that is why in our approach we regard ABoxes as 'black boxes.'

Observe that, in general, more differences are detected when we consider Σ-queries rather than Σ-concept inclusions. Indeed, let $\mathcal{L}$ be one of *DL-Lite*$_{bool}$ and *DL-Lite*$_{horn}$. To see that any difference detected by means of concept inclusions can also be detected by means of queries, suppose that we have $\mathcal{T}_1 \not\models C_1 \sqsubseteq C_2$ and $\mathcal{T}_2 \models C_1 \sqsubseteq C_2$, for some Σ-concept inclusion $C_1 \sqsubseteq C_2$ in $\mathcal{L}$. Consider the ABox $\mathcal{A} = \{C_1(a)\}$ and the query $q = C_2(a)$. Then $(\mathcal{T}_2, \mathcal{A}) \models q$, 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 $B, B_1, \dots, B_k$ are basic concepts.) To show that the converse does not hold, namely that queries can detect more differences than concept inclusions, we consider the following example. (Most of the claims in the examples below can be verified directly or using the criteria of Theorem 11 below.)

Example 2 Take $\Sigma = \{\mathsf{Lecturer}, \mathsf{Course}\}$, $\mathcal{T}_1 = \emptyset$, and

$$\mathcal{T}_2 = \{\mathsf{Lecturer} \sqsubseteq \exists \mathsf{teaches},\ \exists \mathsf{teaches}^- \sqsubseteq \mathsf{Course}\}.$$

Intuitively, the only consequence of $\mathcal{T}_2$ over Σ is 'if there is a lecturer, then there is a course,' but it cannot be expressed as a Σ-concept inclusion. Thus, $\mathcal{T}_1$ Σ-concept entails $\mathcal{T}_2$ (in both *DL-Lite*$_{bool}$ and *DL-Lite*$_{horn}$). However, $\mathcal{T}_1$ does not Σ-query entail $\mathcal{T}_2$. Indeed, let $\mathcal{A} = \{\mathsf{Lecturer}(a)\}$ and $q = \exists y\, \mathsf{Course}(y)$. Then $(\mathcal{T}_1, \mathcal{A}) \not\models q$ but $(\mathcal{T}_2, \mathcal{A}) \models q$.

It is also of interest to observe that Σ-query entailment in *DL-Lite*$_{horn}$ does not imply Σ-query entailment in *DL-Lite*$_{bool}$ (the converse implication follows immediately from the fact that *DL-Lite*$_{horn}$ is a fragment of *DL-Lite*$_{bool}$).

Example 3 Let $\Sigma = \{\mathsf{Lecturer}\}$, $\mathcal{T}_1 = \emptyset$, and

$$\mathcal{T}_2 = \{\mathsf{Lecturer} \sqsubseteq \exists \mathsf{teaches},\ \mathsf{Lecturer} \sqcap \exists \mathsf{teaches}^- \sqsubseteq \bot\}$$

Then $\mathcal{T}_1$ does not Σ-query entail $\mathcal{T}_2$ in *DL-Lite*$_{bool}$: just take $\mathcal{A}$ as before and $q = \exists y\, \neg\mathsf{Lecturer}(y)$. But $\mathcal{T}_1$ Σ-query entails $\mathcal{T}_2$ in *DL-Lite*$_{horn}$.

The first two notions of difference in Definition 1 do not take into account any *context ontologies* in which $\mathcal{T}_1$ or $\mathcal{T}_2$ may be used, nor do they cover the situation where $\mathcal{T}_1$ and $\mathcal{T}_2$ are changed by adding new axioms. To accommodate for this, we have introduced *strong* versions of entailment and difference. If two versions of ontologies strongly Σ-entail each other for their shared signature Σ, then they can be safely *replaced* by each other within any ontology $\mathcal{T}$ which only uses symbols from Σ; after such a replacement no differences between the sets of derivable Σ-concept inclusions (or answers to Σ-queries) can be detected. To see that the 'weak' notions of entailment do not always have this replacement property, consider the following example.

Example 4 Let $\mathcal{T}_1 = \emptyset$ and $\mathcal{T}_2$ be the TBox from Example 3 saying that every lecturer teaches and that a lecturer is not something which is taught. Let, as before, $\Sigma = \{\mathsf{Lecturer}\}$. Then $\mathcal{T}_1$ and $\mathcal{T}_2$ Σ-concept entail each other in *DL-Lite*$_{bool}$. But for $\mathcal{T} = \{\top \sqsubseteq \mathsf{Lecturer}\}$, we have $\mathcal{T}_1 \cup \mathcal{T} \not\models \top \sqsubseteq \bot$ and $\mathcal{T}_2 \cup \mathcal{T} \models \top \sqsubseteq \bot$, i.e., the former TBox is consistent while the latter is not. Thus, the difference between $\mathcal{T}_1$ and $\mathcal{T}_2$ becomes visible if we extend them with the ontology $\mathcal{T}$.

Example 4 shows also that Σ-query entailment in *DL-Lite*$_{horn}$ does not imply strong Σ-concept entailment in *DL-Lite*$_{horn}$. In the context of defining modules within ontologies, taking into account changes to ontologies and context ontologies has been strongly advocated in (Grau *et al.* 2007b), which inspired our definitions. The following example shows that strong Σ-concept entailment in *DL-Lite*$_{horn}$ does not imply strong Σ-concept entailment in *DL-Lite*$_{bool}$.

Example 5 Consider the *DL-Lite*$_{horn}$ TBoxes

$$\mathcal{T}_1 = \big\{\mathsf{Male} \sqcap \mathsf{Female} \sqsubseteq \bot,\ \top \sqsubseteq \exists \mathsf{father},\ \top \sqsubseteq \exists \mathsf{mother},\ \exists \mathsf{father}^- \sqsubseteq \mathsf{Male},\ \exists \mathsf{mother}^- \sqsubseteq \mathsf{Female}\big\},$$

$$\mathcal{T}_2 = \big\{\top \sqsubseteq \exists \mathsf{id},\ \mathsf{Male} \sqcap \exists \mathsf{id}^- \sqsubseteq \bot,\ \mathsf{Female} \sqcap \exists \mathsf{id}^- \sqsubseteq \bot\big\},$$

and let $\Sigma = \{\mathsf{Male}, \mathsf{Female}, \mathsf{father}, \mathsf{mother}\}$. $\mathcal{T}_2$ implies that $\top \not\sqsubseteq \mathsf{Male} \sqcup \mathsf{Female}$. Now, in *DL-Lite*$_{bool}$, $\mathcal{T}_1 \cup \mathcal{T}_2$ is not strongly Σ-concept entailed by $\mathcal{T}_1$: it is enough to take $\mathcal{T} = \{\top \sqsubseteq \mathsf{Male} \sqcup \mathsf{Female}\}$. However, $\mathcal{T}_1 \cup \mathcal{T}_2$ is strongly Σ-entailed by $\mathcal{T}_1$ in *DL-Lite*$_{horn}$.

Semantic Criteria of Σ-Entailment

Now we compare the notions of Σ-difference and Σ-entailment in a systematic way using model-theoretic characterisations. Our first observation generalises the well-known result from propositional logic according to which two propositional Horn theories entail the same Horn formulas if, and only if, these theories have the same consequences in the class of all propositional formulas.

Theorem 6 *For any DL-Lite*$_{horn}$ *TBoxes* $\mathcal{T}_1$, $\mathcal{T}_2$ *and any signature* Σ, *the following two conditions are equivalent*:

- $\mathcal{T}_1$ Σ*-concept entails* $\mathcal{T}_2$ *in DL-Lite*$_{bool}$;
- $\mathcal{T}_1$ Σ*-concept entails* $\mathcal{T}_2$ *in DL-Lite*$_{horn}$.

Examples 3 and 5 show that this theorem does not hold for the stronger notions of Σ-entailment. Moreover, for neither *DL-Lite*$_{horn}$ nor *DL-Lite*$_{bool}$ any of the stronger notions is equivalent to Σ-concept entailment. Our second theorem summarises the classification of the remaining notions and shows that in all those cases where we have not provided counterexamples our notions of Σ-entailment are equivalent.

Theorem 7 *Let* $\mathcal{L}$ *be DL-Lite*$_{bool}$ *or DL-Lite*$_{horn}$, $\mathcal{T}_1$ *and* $\mathcal{T}_2$ *TBoxes in* $\mathcal{L}$, *and* Σ *a signature. For* $\mathcal{L} =$ *DL-Lite*$_{bool}$, *the following conditions are equivalent*:

(1) $\mathcal{T}_1$ Σ*-query entails* $\mathcal{T}_2$ *in* $\mathcal{L}$;

(2) $\mathcal{T}_1$ *strongly* Σ*-concept entails* $\mathcal{T}_2$ *in* $\mathcal{L}$;

(3) $\mathcal{T}_1$ *strongly* Σ*-query entails* $\mathcal{T}_2$ *in* $\mathcal{L}$.

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

Thus, the full comparison table looks as follows:

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

The equivalence results of Theorem 7 follow from the model-theoretic characterisations of the notions of Σ-entailment to be presented below. In this paper, our characterisations will have a somewhat syntactic flavour in the sense that they are formulated in terms of *types*—syntactic abstractions of domain elements—realised in models, rather than in model-theoretic terms. The advantage of such characterisations is that they can be used directly for designing decision algorithms, despite the fact that the underlying models are often infinite as neither *DL-Lite*$_{bool}$ nor *DL-Lite*$_{horn}$ has the finite model property (Calvanese *et al.* 2005). Needless to say, however, that the correctness of the type-based characterisations presented below require model constructions (see Appendix).

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 in Q and $\neg(\geq q\, R) \in \boldsymbol{t}$ then $\neg(\geq q'\, R) \in \boldsymbol{t}$.

Clearly, 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 $x \in C^{\mathcal{I}}$, for all $C \in \boldsymbol{t}$. In this case we say that $\boldsymbol{t}$ is *realised* (at x) in $\mathcal{I}$.

Definition 8 For a TBox $\mathcal{T}$, a ΣQ-type $\boldsymbol{t}$ is called $\mathcal{T}$*-realisable* if $\boldsymbol{t}$ is realised in a model for $\mathcal{T}$. 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 TBox $\mathcal{T}$, let $Q_{\mathcal{T}}$ denote the set of numerical parameters occurring in $\mathcal{T}$ together with 1. The following conditions are equivalent:

- $\mathcal{T}_1$ Σ-concept entails $\mathcal{T}_2$ in *DL-Lite*$_{bool}$;
- every $\mathcal{T}_1$-realisable $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-type is $\mathcal{T}_2$-realisable.

This equivalence is trivial if one considers $\Sigma\mathbb{N}$-types instead of $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types. Thus, the message here is that it is sufficient to consider only parameters from $Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$.

For Σ-query entailment in *DL-Lite*$_{bool}$ (and the two other equivalent notions), the following conditions are equivalent:

- $\mathcal{T}_1$ Σ-query entails $\mathcal{T}_2$ in *DL-Lite*$_{bool}$;
- every precisely $\mathcal{T}_1$-realisable set Ξ of $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types is precisely $\mathcal{T}_2$-realisable.

Intuitively, while Σ-concept entailment is a 'local' form of entailment referring to one point in a model, Σ-query entailment and strong Σ-concept/query entailment are 'global' in the sense that all points of models have to be considered.

Example 9 In Example 2, to compute the $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types, we do not require numerical parameters, as Σ contains no role names. There are four $\mathcal{T}_1$-realisable Σ-types $\{(\neg)\mathsf{Lecturer}, (\neg)\mathsf{Course}\}$. All of these are $\mathcal{T}_2$-realisable as well. However, the singleton set $\{\{\mathsf{Lecturer}, \neg\mathsf{Course}\}\}$ is precisely $\mathcal{T}_1$-realisable but not precisely $\mathcal{T}_2$-realisable.

In the case of *DL-Lite*$_{horn}$ more definitions are required. Given a ΣQ-type $\boldsymbol{t}$, let $\boldsymbol{t}^+ = \{B \in \boldsymbol{t} \mid B \text{ a basic concept}\}$ (i.e., positive part of the type). 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 Σ-entailment for *DL-Lite*$_{horn}$:

Definition 10 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 a type from Ξ. We also say that Ξ is *meet-precisely* $\mathcal{T}$*-realisable* if there is a model $\mathcal{I}$ for $\mathcal{T}$ such that, for every ΣQ-type $\boldsymbol{t}$ realised in $\mathcal{I}$, $\Xi_{\boldsymbol{t}} \neq \emptyset$ and $\boldsymbol{t}^+ = \bigcap_{\boldsymbol{t}_i \in \Xi_{\boldsymbol{t}}} \boldsymbol{t}_i^+$, where $\Xi_{\boldsymbol{t}} = \{\boldsymbol{t}_i \in \Xi \mid \boldsymbol{t}^+ \subseteq \boldsymbol{t}_i^+\}$. (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.)

Theorem 11 *Let* $\mathcal{L} \in \{$*DL-Lite*$_{bool}$, *DL-Lite*$_{horn}\}$ *and* α *be one of the four notions of* Σ*-entailment. For a signature* Σ *and TBoxes* $\mathcal{T}_1$ *and* $\mathcal{T}_2$ *in* $\mathcal{L}$, *the following are equivalent*:

- $\mathcal{T}_1$ α *entails* $\mathcal{T}_2$ *in* $\mathcal{L}$;
- *every precisely* $\mathcal{T}_1$*-realisable set of* $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$ *types satisfies the corresponding property from the following table*:

entailment α	*language* $\mathcal{L}$	
	DL-Lite$_{horn}$	*DL-Lite$_{bool}$*
Σ-*concept*[3]	$\mathcal{T}_2$-*realisable*	$\mathcal{T}_2$-*realisable*
Σ-*query*	*sub-precisely* $\mathcal{T}_2$-*realisable*	*precisely* $\mathcal{T}_2$-*realisable*
strong Σ-*concept*	*meet-precisely* $\mathcal{T}_2$-*realisable*	
strong Σ-*query*		

Example 12 Consider the TBoxes from Example 3. Again, we do not require numerical parameters because Σ does not contain role names. The $\mathcal{T}_1$-realisable Σ-types are $\{\neg\mathsf{Lecturer}\}$ and $\{\mathsf{Lecturer}\}$, and both are $\mathcal{T}_2$-realisable. Hence $\mathcal{T}_1$ Σ-concept entails $\mathcal{T}_2$ in *DL-Lite$_{bool}$* (and, therefore, in *DL-Lite$_{horn}$*). The singleton set $\{\{\mathsf{Lecturer}\}\}$ is precisely $\mathcal{T}_1$-realisable, but not precisely $\mathcal{T}_2$-realisable. Hence $\mathcal{T}_1$ does not Σ-query entail $\mathcal{T}_2$ in *DL-Lite$_{bool}$*. However, $\{\{\mathsf{Lecturer}\}\}$ is sub-precisely $\mathcal{T}_1$-realisable and, therefore $\mathcal{T}_1$ Σ-query entails $\mathcal{T}_2$ in *DL-Lite$_{horn}$*. On the other hand, $\{\{\mathsf{Lecturer}\}\}$ is not meet-precisely $\mathcal{T}_2$-realisable, and so $\mathcal{T}_1$ does not strongly Σ-concept entail $\mathcal{T}_2$ in *DL-Lite$_{horn}$*.

Robustness Properties

Results regarding Σ-difference and Σ-entailment can be easily misinterpreted and are of limited use if these notions do not enjoy certain robustness properties. To start with, recall that in the definition of essentially positive existential queries for *DL-Lite$_{bool}$*, we allow *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 entailment: under the second definition, the TBox $\mathcal{T}_2$ from Example 3 is Σ-query entailed by $\mathcal{T}_1 = \emptyset$, 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 only positive queries allowed, the addition of the *definition* $B \equiv \neg\mathsf{Lecturer}$ to $\mathcal{T}_2$ and B to Σ would result in a TBox which is not Σ-query entailed by $\mathcal{T}_1$ in *DL-Lite$_{bool}$* any longer. This kind of non-robust behaviour of the notion of Σ-entailment is clearly undesirable. Obviously, the formulations we gave are robust under the addition of definitions to TBoxes. We now consider two other robustness conditions.

Theorem 13 *Let 'Σ-entails' be one of the eight notions of Σ-entailment given in Definition 1.*

- *The relation 'Σ-entails' is robust under vocabulary extensions: if $\mathcal{T}_1$ Σ-entails $\mathcal{T}_2$, then $\mathcal{T}_1$ Σ'-entails $\mathcal{T}_2$, for every Σ' such that $\Sigma' \cap sig(\mathcal{T}_2) \subseteq \Sigma$.*
- *The relation 'Σ-entails' is robust under joins: if $\mathcal{T}$ and $\mathcal{T}_i$ Σ-entail each other, for $i = 1, 2$, and $sig(\mathcal{T}_1) \cap sig(\mathcal{T}_2) \subseteq \Sigma$, then $\mathcal{T}$ Σ-entails $\mathcal{T}_1 \cup \mathcal{T}_2$.*

Robustness under vocabulary extensions is of particular importance for query Σ-entailment and the strong versions of Σ-entailment. For example, it implies that if $\mathcal{T}_1$ strongly Σ-query entails $\mathcal{T}_2$ then, for any ABox $\mathcal{A}$, TBox $\mathcal{T}$ and query q containing, besides Σ, *arbitrary* symbols not occurring in $\mathcal{T}_2$, we have $(\mathcal{T}_1 \cup \mathcal{T}, \mathcal{A}) \models q(\boldsymbol{a})$ whenever $(\mathcal{T}_2 \cup \mathcal{T}, A) \models q(\boldsymbol{a})$. This property is critical for applications, as it is hardly possible to restrict ABoxes and context ontologies to a fixed signature Σ and not permit the use of any fresh symbols.

Robustness under joins is of interest for collaborative ontology development. This property means that if two (or more) ontology developers extend a given ontology $\mathcal{T}$ 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 $\mathcal{T}$ and all their additional axioms provided that their individual extensions are safe for Σ.

Both robustness conditions are closely related to the well-known *Robinson consistency lemma* and *interpolation* (see e.g., (Chang & Keisler 1990)), which have been investigated in the context of modular software specification (Diaconescu, Goguen, & Stefaneas 1993) as well. They typically fail for description logics with nominals and/or role hierarchies (Areces & ten Cate 2006; Konev *et al.* 2007). Observe that, for robustness under joins, *mutual* Σ-entailment of $\mathcal{T}$ and $\mathcal{T}_i$ is required:

Example 14 Let $\mathcal{T}_1 = \{A \sqsubseteq \exists R, \exists R^- \sqsubseteq B\}$, $\mathcal{T}_2 = \mathcal{T}$, $\mathcal{T} = \{\top \sqsubseteq \neg B\}$, and $\Sigma = \{A, B\}$. Then $\mathcal{T}$ Σ-concept entails $\mathcal{T}_i$, $i = 1, 2$, but $\mathcal{T}_1 \cup \mathcal{T}_2 \models \top \sqsubseteq \neg A$, and so it is not Σ-concept entailed by $\mathcal{T}$.

Complexity and Algorithms

We first determine the complexity of deciding Σ-entailment and then consider the problem of computing Σ-differences.

Theorem 15 *For all notions of Σ-entailment introduced in Definition 1, deciding Σ-entailment is Π_2^p-complete in DL-Lite$_{bool}$ and* coNP*-complete in DL-Lite$_{horn}$.*

The lower bounds follow immediately from the corresponding lower bounds for propositional logic and its Horn fragment. The upper bound for Σ-concept entailment in *DL-Lite$_{bool}$* and *DL-Lite$_{horn}$* is rather straightforward: by the characterisation of Theorem 11, it is sufficient to check that every $\mathcal{T}_1$-realisable $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-type is $\mathcal{T}_2$-realisable. Thus, to check non Σ-concept entailment in *DL-Lite$_{bool}$*, the algorithm guesses a $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-type and checks, using an NP-oracle, that is is $\mathcal{T}_1$-realisable but not $\mathcal{T}_2$-realisable. For *DL-Lite$_{horn}$*, this latter check can be done in deterministic polynomial time.

Proving the upper bounds for the remaining decision problems is harder: the criteria of Theorem 11 do not make any claim regarding the cardinality of the sets of $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types one has to consider (there are exponentially many $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types). Our upper bound proof shows that it suffices to consider sets Ξ of $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types the size of which is bounded by a *linear* function in the size of the TBoxes. Then, for *DL-Lite$_{bool}$* TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$, one can decide whether $\mathcal{T}_1$ does *not* Σ-entail $\mathcal{T}_2$ by guessing a set Ξ of linearly many $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types and checking that it is precisely $\mathcal{T}_1$-realisable and *not* precisely $\mathcal{T}_2$-realisable. The Appendix

[3] Every $\mathcal{T}_1$-realisable type is always contained in a precisely $\mathcal{T}_1$-realisable set.

provides an NP algorithm deciding whether a given set of ΣQ-types is precisely $\mathcal{T}$-realisable. This gives the Π_2^p upper bound for Σ-query entailment in *DL-Lite$_{bool}$*. The procedures for *DL-Lite$_{horn}$* are similar: given *DL-Lite$_{horn}$* TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$, one can decide whether $\mathcal{T}_1$ does not (strongly) Σ-query entail $\mathcal{T}_2$ by guessing Ξ and checking that it is precisely $\mathcal{T}_1$-realisable and *not* sub-precisely (respectively, meet-precisely) $\mathcal{T}_2$-realisable. The Appendix provides polynomial *deterministic* algorithms deciding whether a set of ΣQ-types is precisely, sub-precisely and meet-precisely $\mathcal{T}$-realisable, for a *DL-Lite$_{horn}$* TBox $\mathcal{T}$. This gives CONP upper bounds for Σ-query and strong Σ-query entailment.

Observe that deciding Σ-entailment and conservativity is much harder for most DLs: it is EXPTIME-complete for $\mathcal{EL}$ (Lutz & Wolter 2007), 2EXPTIME-complete for $\mathcal{ALC}$ and $\mathcal{ALCQI}$, and undecidable for $\mathcal{ALCQIO}$ (Ghilardi, Lutz, & Wolter 2006; Lutz, Walther, & Wolter 2007).

In applications, it is not enough just to decide whether two ontologies differ w.r.t. a signature. If the ontologies are different, the ontology engineer needs an informative list of differences. Observe that the set of Σ-differences as defined in Definition 1 is either infinite or empty. Thus, only approximations of these sets can be computed. By the criteria of Theorem 11, for Σ-concept difference the ΣQ-types which are $\mathcal{T}_1$-realisable but not $\mathcal{T}_2$-realisable are obvious candidates to include in such a set. Such a type contains, for each concept name $A \in \Sigma$ and $(\geq qR)$ with $q \in Q$, $R \in \Sigma$, either the concept itself, or its negation. If there are too many Σ-differences (remember, there are exponentially many types) and the resulting list is incomprehensible, the user can step-by-step decrease the size of Σ (e.g., by removing elements X from Σ such that two types which coincide except for X are in the Σ-difference) until the set of types in the Σ-difference can be analysed. Moreover, as a second step the user might consider applying pinpointing algorithms (Schlobach & Cornet 2003) which exhibit the axioms in the ontology from which the Σ-differences are derivable. For stronger versions of Σ-difference, it appears to be unavoidable to consider precisely $\mathcal{T}_1$-realisable sets of ΣQ-types, which are (in one of the four ways described in Theorem 11) not precisely $\mathcal{T}_2$-realisable. We leave a systematic study of the problem of constructing or approximating the query and strong versions of difference for future research.

Experimental Results

To see whether the algorithms of the previous section can be used in practice, we refined the criteria for Σ-concept and Σ-query entailment in *DL-Lite$_{bool}$* and encoded them by means of $\forall\exists$ QBFs (the satisfiability problem for which is Π_2^p-complete). The reader can find the encodings in the full version available at `www.dcs.bbk.ac.uk/~roman/qbf2`. We used these QBF translations to check Σ-concept/query entailment for *DL-Lite$_{bool}$* ontologies with the help of three *off-the-shelf* QBF solvers: sKizzo (Benedetti 2005), 2clsQ (Samulowitz & Bacchus 2006) and Quaffle (Zhang & Malik 2002b; 2002a). As our benchmarks, we considered three series of instances of the form $(\mathcal{T}_1, \mathcal{T}_2, \Sigma)$. In the *NN-series*, $\mathcal{T}_1$ does not Σ-concept entails $\mathcal{T}_2$; in the *YN-series*, $\mathcal{T}_1$ Σ-concept but not Σ-query entails $\mathcal{T}_2$; and in the *YY-series*, $\mathcal{T}_1$ Σ-query entails $\mathcal{T}_2$. The sizes of the instances are uniformly distributed over the intervals given in the table below.

series	no. of instances	axioms		basic concepts		
		$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_1$	$\mathcal{T}_2$	Σ
NN	420	59–154	74–198	47–121	49–146	5–52
YN	252	56–151	77–191	44–119	58–145	6–45
YY	156	54–88	62–110	43–79	47–94	6–32

It is to be noted that our ontologies were *not* randomly generated. On the contrary, we used 'typical' *DL-Lite* ontologies available on the Web: extensions of *DL-Lite$_{bool}$* fragments of the standard 'department ontology' as well as *DL-Lite$_{bool}$* representations of the ER diagrams used in the QuOnto system (`www.dis.uniroma1.it/~quonto/`).

The next table illustrates the size of the QBF translations of our instances for both Σ-concept and Σ-query entailment.

series	Σ-concept entailment QBF		Σ-query entailment QBF	
	variables	clauses	variables	clauses
NN	1,469–11,752	2,391–18,277	1,715–15,174	5,763–163,936
YN	1,460–11,318	2,352–17,424	1,755–14,723	7,006–151,452
YY	1,526–4,146	2,200–6,079	1,510–4,946	5,121–29,120

The large difference between the size of the QBF translations for Σ-concept and Σ-query entailment (say, 18,277 v. 163,936 clauses in the same instance) reflects the difference between simple and precise realisability of sets of types (cf. Theorem 11): roughly, the QBF encoding of the latter requires *quadratical* number of clauses (in the number of role names) whereas the former needs only linearly many.

A brief summary of the tests, conducted on a 3GHz P4 machine with 2GB RAM, is given in Fig. 1, where the graphs in the upper (lower) row show the percentage of solved instances for Σ-concept (respectively, Σ-query) entailment; for details and more charts see `www.dcs.bbk.ac.uk/~roman/qbf2`.

The main conclusion of the tests is that automated checking of Σ-entailment between *DL-Lite$_{bool}$* ontologies[4] is indeed possible, even with *off-the-shelf general purpose software* (let alone dedicated reasoners). Although of the same worst-case complexity, in practice Σ-concept entailment turns our to be much easier to check than Σ-query entailment. Quaffle solved all of our 828 Σ-concept instances. However, none of the solvers could cope with all Σ-query instances, with those of the YY series being especially hard. All in all, we have solved more than 90% of the Σ-query instances. Another interesting observation is that, for Σ-concept entailment, bigger Σs usually meant harder instances, whereas the impact of the size of Σ on Σ-query entailment was rather limited. Finally, our tests showed that none of the three solvers was better than the others when checking Σ-entailment: Quaffle was the best for Σ-concept entailment, 2clsQ for Σ-query entailment with the answer 'NO,' and sKizzo for Σ-query entailment with the answer 'YES.' On the other hand, having tried various quantifier orderings in the prenex QBF translations (`www.dcs.bbk.ac.uk/~roman/qbf2`), we have identified a number of strategies that could dramatically improve performance of

[4]The main application area of the *DL-Lite* family of logics is conceptual data modelling and data integration, where typical *DL-Lite* ontologies do not contain more than a few hundred axioms.

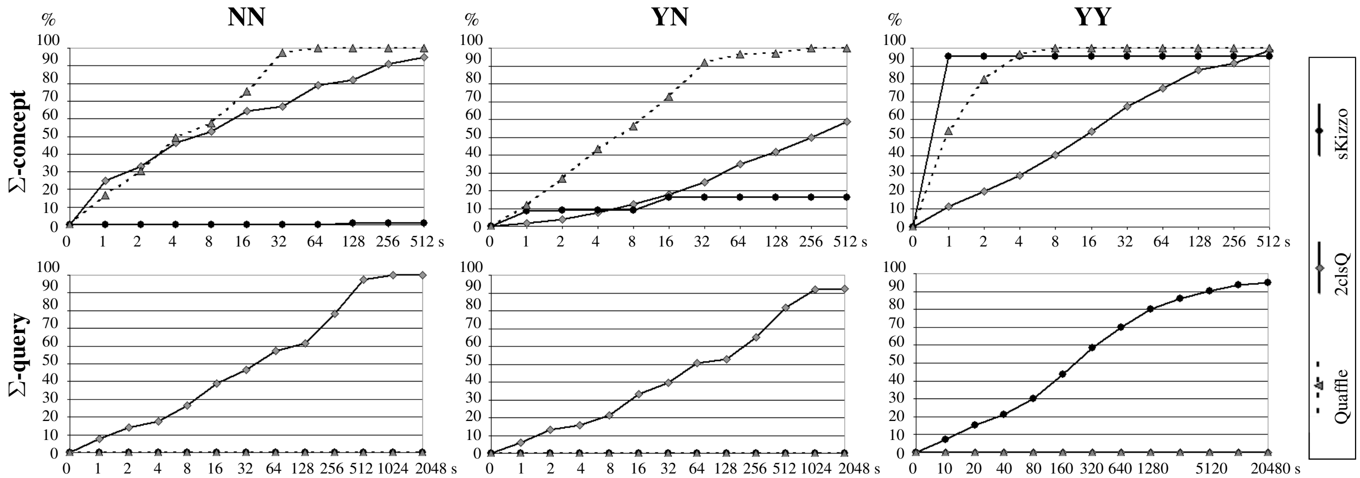

Figure 1: Percentage of instances solved (Y axis) for each timeout (X axis).

QBF solvers when checking Σ-query entailment.

Conclusion

We have analysed the relation between various notions of difference and entailment w.r.t. a signature in description logics *DL-Lite$_{bool}$* and *DL-Lite$_{horn}$*, and proved that the corresponding reasoning problems are not harder (at least theoretically) than similar problems in propositional logic. We also demonstrated that an efficient reasoning service checking entailment between *DL-Lite$_{bool}$* ontologies can be implemented, even using off-the-shelf QBF solvers. Future research problems include the following: (1) The algorithms presented for Σ-entailment provide a basis for developing *module extraction algorithms* for *DL-Lite* ontologies. Such an algorithm should output, given an ontology and a signature Σ, a minimal sub-ontology which Σ-entails the full ontology; see (Grau *et al.* 2007a) for an overview. It remains to develop the details of such a procedure for *DL-Lite*. (2) We have only provided a sketch of how an approximation of the differences between different versions of an ontology can be computed. Further experimental results are required to evaluate the feasibility of this approach.

References

Acciarri, A.; Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; Palmieri, M.; and Rosati, R. 2005. QuOnto: Querying ontologies. In *Proc. of AAAI*, 1670–1671.

Antoniou, G., and Kehagias, A. 2000. A note on the refinement of ontologies. *Int. J. of Intelligent Systems* 15(7):623–632.

Areces, C., and ten Cate, B. 2006. Hybrid logics. In Blackburn, P.; van Benthem, J.; and Wolter, F., eds., *Handbook of Modal Logic*. Elsevier.

Artale, A.; Calvanese, D.; Kontchakov, R.; and Zakharyaschev, M. 2007. DL-Lite in the light of first-order logic. In *AAAI*, 361–366.

Benedetti, M. 2005. sKizzo: A suite to evaluate and certify QBFs. In Nieuwenhuis, R., ed., *Proc. of CADE–20*, 369–376.

Borgida, A.; Brachman, R.; McGuinness, D.; and Resnick, L. A. 1989. Classic: A structural data model for objects. In *Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, 58–67.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2005. DL-Lite: Tractable description logics for ontologies. In *Proc. of AAAI*, 602–607.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2006. Data complexity of query answering in description logics. In *Proc. of KR*, 260–270.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; Poggi, A.; and Rosati, R. 2007. Mastro-I: Efficient integration of relational data through DL ontologies. In *DL*, 1670–1671.

Chang, C., and Keisler, H. 1990. *Model Theory*. Elsevier.

Diaconescu, R.; Goguen, J.; and Stefaneas, P. 1993. Logical support for modularisation. In Huet, G., and Plotkin, G., eds., *Logical Environments*, 83–130. Cambridge University Press, New York.

Ghilardi, S.; Lutz, C.; and Wolter, F. 2006. Did I damage my ontology? A case for conservative extensions in description logic. In *Proc. of KR*, 187–197.

Grau, B. C.; Horrocks, I.; Kazakov, Y.; and Sattler, U. 2007a. Just the right amount: Extracting modules from ontologies. In *Proc. of WWW*, 717–726.

Grau, B. C.; Horrocks, I.; Kazakov, Y.; and Sattler, U. 2007b. A logical framework for modularity of ontologies. In *Proc. of IJCAI*, 298–303.

Konev, B.; Lutz, C.; Walther, D.; and Wolter, F. 2007. Formal properties of modularisation. Unpublished manuscript.

Lutz, C., and Wolter, F. 2007. Conservative extensions in the lightweight description logic $\mathcal{EL}$. In *Proc. CADE*, 84–99.

Lutz, C.; Walther, D.; and Wolter, F. 2007. Conservative extensions in expressive description logics. In *Proc. of IJCAI*, 453–458.

Noy, N., and Musen, M. 2002. PromptDiff: a fixed-point algorithm for comparing ontology versions. In *AAAI*, 744–750.

Samulowitz, H., and Bacchus, F. 2006. Binary clause reasoning in QBF. In *Proc. of SAT*, volume 4121 of *LNCS*, 353–367.

Schlobach, S., and Cornet, R. 2003. Non-standard reasoning services for debugging of description logic terminologies. In *Proc. of IJCAI*, 355–362.

Zhang, L., and Malik, S. 2002a. Conflict driven learning in a quantified Boolean satisfiability solver. In *Proc. of IEEE/ACM Int. Conf. on Computer-Aided Design*, 442–449.

Zhang, L., and Malik, S. 2002b. Towards a symmetric treatment of satisfaction and conflicts in quantified Boolean formula evaluation. In *Proc. of CP*, volume 2470 of *LNCS*, 200–215.

Appendix

A Model-theoretic properties of *DL-Lite*

In this section, we prove an amalgamation property of *DL-Lite$_{bool}$* models. The construction will be used throughout the paper.

Let Σ be a signature and Q a set of positive natural numbers containing 1. Recall that a ΣQ-*concept* in *DL-Lite$_{bool}$* is 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 in Q and $\neg(\geq q\, R) \in \boldsymbol{t}$ then $\neg(\geq q'\, R) \in \boldsymbol{t}$.

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 $\mathcal{I}$* and denote it by $\boldsymbol{t}_{\mathcal{I}}^{\Sigma Q}(x)$. 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. 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. In this case we say that $\mathcal{I}$ *realises precisely* the types in Ξ.

Our aim in this section is to introduce, in Lemma A.1, an operation which allows us to amalgamate interpretations in a 'truth-preserving' way. We will need two simple definitions.

Given a signature Σ, we say that two 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 Σ, and $(x, y) \in P^{\mathcal{I}}$ iff $(f(x), f(y)) \in P^{\mathcal{J}}$, for every role name P in Σ. Clearly, Σ-isomorphic interpretations cannot be distinguished by Σ-TBoxes, Σ-ABoxes or Σ-queries.

Given a set $\mathcal{I}_i$, $i \in I$, of interpretations and $0 \in I$, define the interpretation

$$\mathcal{J} \;=\; \bigoplus_{i \in I} \mathcal{I}_i,$$

where $\Delta^{\mathcal{J}} = \{(i, w) \mid i \in I, w \in \Delta_i\}$, $a^{\mathcal{J}} = (0, a^{\mathcal{I}_1})$, for an object name a, $A^{\mathcal{J}} = \{(i, w) \mid w \in A^{\mathcal{I}_i}\}$, for a concept name A, and $P^{\mathcal{J}} = \{((i, w_1), (i, w_2)) \mid (w_1, w_2) \in P^{\mathcal{I}_i}\}$, for a role name P. $\mathcal{J}$ can be regarded as a *disjoint union* of the $\mathcal{I}_i$. Given an interpretation $\mathcal{I}$, we set

$$\mathcal{I}^\omega \;=\; \bigoplus_{i \in \omega} \mathcal{I}_i,$$

where $\mathcal{I}_i = \mathcal{I}$ for $i \in \omega$. It should be clear that Σ-TBoxes, Σ-ABoxes or Σ-queries (for *any* signature Σ) cannot distinguish between $\mathcal{I}$ and $\mathcal{I}^\omega$.

For a TBox $\mathcal{T}$, denote by $Q_{\mathcal{T}}$ the set of all numerical parameters occurring in $\mathcal{T}$ together with 1.

The following lemma provides an important model-theoretic property of *DL-Lite$_{bool}$* that will be frequently used to establish model-theoretic characterisations of various notions of Σ-entailment.

Lemma A.1 *Let $\mathcal{I}_1$ and $\mathcal{I}_2$ be* (*at most countable*) *models for TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$, respectively, and let Σ be a signature such that $sig(\mathcal{T}_1) \cap sig(\mathcal{T}_2) \subseteq \Sigma$. If interpretations $\mathcal{I}_1$ and $\mathcal{I}_2$ realise precisely the same $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types, then there is an interpretation $\mathcal{I}^*$ such that:*

- $\mathcal{I}^* \models \mathcal{T}_1 \cup \mathcal{T}_2$,
- $\mathcal{I}^* \sim_\Sigma \mathcal{I}_1^\omega$, *and*
- $\mathcal{I}^*$, *$\mathcal{I}_1$ and $\mathcal{I}_2$ realise the same set of $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types.*

Proof. Let Ξ be the set of $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types realised in $\mathcal{I}_1$ (and $\mathcal{I}_2$). We show that $\mathcal{I}_1^\omega$ can be expanded to a model $\mathcal{I}^*$ for $\mathcal{T}_1 \cup \mathcal{T}_2$. As both $\mathcal{I}_1^\omega$ and $\mathcal{I}_2^\omega$ realise each $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-type from Ξ by countably infinitely many points, there is a bijection $f\colon \Delta^{\mathcal{I}_2^\omega} \to \Delta^{\mathcal{I}_1^\omega}$ which is *invariant under* $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types. Now, we set $\Delta^{\mathcal{I}^*} = \Delta^{\mathcal{I}_1^\omega}$ and, for all object names a, concept names A, and role names P,

$$a^{\mathcal{I}^*} = a^{\mathcal{I}_1^\omega},$$

$$A^{\mathcal{I}^*} = A^{\mathcal{I}_1^\omega}, \text{ if } A \in \Sigma \cup sig(\mathcal{T}_1),$$

$$A^{\mathcal{I}^*} = \{f(x) \mid x \in A^{\mathcal{I}_2^\omega}\}, \text{ if } A \notin \Sigma \cup sig(\mathcal{T}_1),$$

$$P^{\mathcal{I}^*} = P^{\mathcal{I}_1^\omega}, \text{ if } P \in \Sigma \cup sig(\mathcal{T}_1),$$

$$P^{\mathcal{I}^*} = \{(f(x), f(y)) \mid (x, y) \in P^{\mathcal{I}_2^\omega}\}, \text{ if } P \notin \Sigma \cup sig(\mathcal{T}_1).$$

By definition, $\mathcal{I}^* \sim_{\Sigma \cup sig(\mathcal{T}_1)} \mathcal{I}_1^\omega$ and thus $\mathcal{I}^* \sim_\Sigma \mathcal{I}_1^\omega$. So $\mathcal{I}^* \models \mathcal{T}_1$ and $\mathcal{I}^*$ realises the same $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types as $\mathcal{I}_1$ and $\mathcal{I}_2$. Let us show that $\mathcal{I}^* \models \mathcal{T}_2$. By the definition, we have $x \in A^{\mathcal{I}_2^\omega}$ iff $f(x) \in A^{\mathcal{I}^*}$, for all points x in $\mathcal{I}_2^\omega$ and all concept names $A \in sig(\mathcal{T}_2)$. As $\mathcal{I}_2^\omega$ is a model for $\mathcal{T}_2$, it is enough to prove that, for every $q \in Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$ and every $sig(\mathcal{T}_2)$-role R, we have

$$x \in (\geq q\, R)^{\mathcal{I}_2^\omega} \qquad \text{iff} \qquad f(x) \in (\geq q\, R)^{\mathcal{I}^*}. \qquad (1)$$

If R is not a Σ-role then, by the definition above, the number of R-successors of x in $\mathcal{I}_2^\omega$ and $f(x)$ in $\mathcal{I}^*$ is the same. And if R is a Σ-role, then the $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-type realised by x in $\mathcal{I}_2^\omega$ coincides with the $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-type realised by $f(x)$ in $\mathcal{I}^*$, which again gives (1). Thus, $\mathcal{I}^* \models \mathcal{T}_1 \cup \mathcal{T}_2$. ❑

Our first application of Lemma A.1 is the *uniform interpolation theorem* for *DL-Lite$_{bool}$*.

For $\mathcal{L} \in \{$*DL-Lite$_{bool}$*, *DL-Lite$_{horn}$*$\}$, let $\mathcal{T}$ be a TBox in $\mathcal{L}$ and Σ a signature. A TBox $\mathcal{T}_\Sigma$ in $\mathcal{L}$ is called a *uniform interpolant* for $\mathcal{T}$ w.r.t. Σ in $\mathcal{L}$ if

- $sig(\mathcal{T}_\Sigma) \subseteq \Sigma$;
- $\mathcal{T} \models \mathcal{T}_\Sigma$;
- $\mathcal{T}_\Sigma \models C_1 \sqsubseteq C_2$ whenever $\mathcal{T} \models C_1 \sqsubseteq C_2$, for every $C_1 \sqsubseteq C_2$ in $\mathcal{L}$ with $sig(C_1 \sqsubseteq C_2) \cap sig(\mathcal{T}) \subseteq \Sigma$.

We say that $\mathcal{L}$ has *uniform interpolation* if, for every TBox $\mathcal{T}$ in $\mathcal{L}$ and every signature Σ, there exists a uniform interpolant for $\mathcal{T}$ w.r.t. Σ in $\mathcal{L}$.

Let Σ be a signature and Q a finite set of numerical parameters. As the number of nonequivalent ΣQ-concepts in $\mathcal{L}$ is obviously finite (remember that Σ is finite and there are no nested occurrences of roles in $\mathcal{L}$), for every set $\mathcal{S}$ of concept inclusions $C_1 \sqsubseteq C_2$ in $\mathcal{L}$, where C_1, C_2 are ΣQ-concepts, there is a *finite* subset $\mathcal{S}'$ of $\mathcal{S}$ such that $\mathcal{S}' \models \mathcal{S}$.

Theorem A.2 *DL-Lite$_{bool}$ has uniform interpolation. More precisely, let $\mathcal{T}$ be a TBox in DL-Lite$_{bool}$, Σ a signature, and let $\mathcal{T}'$ be a finite presentation of the set*

$$\mathcal{S} = \{C_1 \sqsubseteq C_2 \mid \mathcal{T} \models C_1 \sqsubseteq C_2, \; C_1 \sqsubseteq C_2 \text{ a } \Sigma Q_{\mathcal{T}}\text{-concept inclusion in DL-Lite}_{bool}\}.$$

Then $\mathcal{T}'$ is a uniform interpolant for $\mathcal{T}$ w.r.t. Σ in DL-Lite$_{bool}$.

Proof. It is sufficient to show that $\mathcal{S} \models C_1 \sqsubseteq C_2$ whenever $\mathcal{T} \models C_1 \sqsubseteq C_2$, for every concept inclusion $C_1 \sqsubseteq C_2$ in *DL-Lite$_{bool}$* with $sig(C_1 \sqsubseteq C_2) \cap sig(\mathcal{T}) \subseteq \Sigma$. Suppose, on the contrary, that $\mathcal{S} \not\models C_1 \sqsubseteq C_2$. Take a model $\mathcal{I}_1$ for $\mathcal{S}$ satisfying $C_1 \sqcap \neg C_2$ and realising all $\mathcal{S}$-realisable $\Sigma Q_{\mathcal{T}}$-types. Take a model $\mathcal{I}_2$ for $\mathcal{T}$ and realising all $\mathcal{T}$-realisable $\Sigma Q_{\mathcal{T}}$-types. Then $\mathcal{I}_1$ and $\mathcal{I}_2$ realise exactly the same $\Sigma Q_{\mathcal{T}}$-types. By Lemma A.1, there exists a model $\mathcal{I}^*$ for $\mathcal{T}$ satisfying $C_1 \sqcap \neg C_2$, which is a contradiction. ❑

Uniform interpolation of *DL-Lite$_{horn}$* will be considerd in Theorem B.5.

We will be using one more immediate consequence of Lemma A.1:

Lemma A.3 *Let $\mathcal{J}$ be an (at most countable) model for $\mathcal{T}_1$ and Σ a signature with $\Sigma \subseteq sig(\mathcal{T}_1)$. Suppose that there is a model for $\mathcal{T}_2$ realising exactly the same $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types as $\mathcal{J}$. Then there is a model $\mathcal{I}^*$ for $\mathcal{T}_2$ such that $\mathcal{I}^* \sim_\Sigma \mathcal{J}^\omega$.*

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

B Model-theoretic characterisations of Σ-entailment

In this section, we give model-theoretic characterisations of the notions of Σ-entailment introduced above. The equivalences stated in Theorems 6, 7 and 11 will follow from these characterisations. Moreover, they will be used for proving the complexity results of Theorem 15.

DL-Lite$_{bool}$

We start with the characterisation of Σ-concept entailment in *DL-Lite$_{bool}$*.

Lemma B.1 *The following conditions are equivalent for DL-Lite$_{bool}$ TBoxes $\mathcal{T}_1$, $\mathcal{T}_2$ and a signature Σ:*

(ce$_\mathrm{b}$) *$\mathcal{T}_1$ Σ-concept entails $\mathcal{T}_2$ in DL-Lite$_{bool}$;*

(r) *every $\mathcal{T}_1$-realisable $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-type is $\mathcal{T}_2$-realisable.*

Proof. **(ce$_\mathrm{b}$)** $\Rightarrow$ **(r)** Suppose that $\boldsymbol{t}$ is a $\mathcal{T}_1$-realisable $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-type which is not $\mathcal{T}_2$-realisable. Then $\mathcal{T}_2 \models D \sqsubseteq \bot$ but $\mathcal{T}_1 \not\models D \sqsubseteq \bot$, for $D = \bigsqcap_{C \in \boldsymbol{t}} C$, contrary to $\mathcal{T}_2$ being Σ-entailed by $\mathcal{T}_1$.

(r) $\Rightarrow$ **(ce$_\mathrm{b}$)** Suppose otherwise. Then there is $C_1 \sqsubseteq C_2$ with $sig(C_1 \sqsubseteq C_2) \subseteq \Sigma$, $\mathcal{T}_2 \models C_1 \sqsubseteq C_2$ and $\mathcal{T}_1 \not\models C_1 \sqsubseteq C_2$. Take the uniform interpolant $\mathcal{S}_2$ of $\mathcal{T}_2$ w.r.t. Σ constructed in the proof of Theorem A.2. Since $\mathcal{S}_2 \models C_1 \sqsubseteq C_2$, there exist $\Sigma Q_{\mathcal{T}_2}$-concepts C_1', C_2' such that $C_1' \sqsubseteq C_2' \in \mathcal{S}_2$ and $\mathcal{T}_1 \not\models C_1' \sqsubseteq C_2'$. Thus, there is a $\mathcal{T}_1$-realisable $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-type that is not $\mathcal{T}_2$-realisable. ❑

Let us now consider the other types of Σ-entailment for *DL-Lite$_{bool}$* TBoxes. The next lemma proves the equivalences of Theorem 7 for *DL-Lite$_{bool}$*. Moreover, it also shows that only numerical parameters from $\mathcal{T}_1 \cup \mathcal{T}_2$ need consideration.

Lemma B.2 *The following conditions are equivalent for DL-Lite$_{bool}$ TBoxes $\mathcal{T}_1$, $\mathcal{T}_2$ and a signature Σ:*

(sce$_\mathrm{b}$) *$\mathcal{T}_1$ strongly Σ-concept entails $\mathcal{T}_2$;*

(qe$_\mathrm{b}$) *$\mathcal{T}_1$ Σ-query entails $\mathcal{T}_2$;*

(sqe$_\mathrm{b}$) *$\mathcal{T}_1$ strongly Σ-query entails $\mathcal{T}_2$;*

(pr) *if a set of $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types is precisely $\mathcal{T}_1$-realisable, then it is precisely $\mathcal{T}_2$-realisable.*

Proof. Implications **(sqe$_\mathrm{b}$)** $\Rightarrow$ **(qe$_\mathrm{b}$)** and **(sqe$_\mathrm{b}$)** $\Rightarrow$ **(sce$_\mathrm{b}$)** follow immediately from definitions.

(pr) $\Rightarrow$ **(sqe$_\mathrm{b}$)** Suppose that there are a Σ-TBox $\mathcal{T}$, a Σ-ABox $\mathcal{A}$ and a Σ-query $q(\boldsymbol{x})$ such that $(\mathcal{T}_2 \cup \mathcal{T}, \mathcal{A}) \models q(\boldsymbol{a})$ but $(\mathcal{T}_1 \cup \mathcal{T}, \mathcal{A}) \not\models q(\boldsymbol{a})$, for some object names $\boldsymbol{a}$. Take a model $\mathcal{J}$ for $(\mathcal{T}_1 \cup \mathcal{T}, \mathcal{A})$ such that $\mathcal{J} \not\models q(\boldsymbol{a})$ and let Ξ be the set of $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types realised in $\mathcal{J}$. Then there is a model for $\mathcal{T}_2$ realising exactly the types in Ξ and, by Lemma A.3, there exists a model $\mathcal{I}^*$ for $\mathcal{T}_2$ such that $\mathcal{I}^* \models (\mathcal{T}, \mathcal{A})$ and $\mathcal{I}^* \not\models q(\boldsymbol{a})$, which is a contradiction.

(qe$_\mathrm{b}$) $\Rightarrow$ **(pr)** Suppose that there is a set Ξ of $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types that is precisely $\mathcal{T}_1$-realisable but not precisely $\mathcal{T}_2$-realisable. Then two cases are possible:

1. For every model $\mathcal{I}$ of $\mathcal{T}_2$, there is some $\boldsymbol{t} \in \Xi$ that is not realised in $\mathcal{I}$. Consider the query $q = \bot$ and the ABox $\mathcal{A}_\Xi = \{C(a_{\boldsymbol{t}}) \mid C \in \boldsymbol{t}, \; \boldsymbol{t} \in \Xi\}$, where $a_{\boldsymbol{t}}$ is a fresh object name for each type $\boldsymbol{t} \in \Xi$. We then have $(\mathcal{T}_2, \mathcal{A}_\Xi) \models q$ but $(\mathcal{T}_1, \mathcal{A}_\Xi) \not\models q$, which is a contradiction.
2. Suppose now that Case 1 does not hold. Consider the ABox $\mathcal{A}_\Xi = \{C(a_{\boldsymbol{t}}) \mid C \in \boldsymbol{t}, \boldsymbol{t} \in \Xi\}$, where $a_{\boldsymbol{t}}$ is a fresh object name, for each type $\boldsymbol{t} \in \Xi$. Let Θ be the set of all $\mathcal{T}_2$-realisable $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types that are not in Ξ. We have $\Theta \neq \emptyset$. Now consider the query
$$q = \exists x \bigvee_{\boldsymbol{t} \in \Theta} \bigwedge_{C \in \boldsymbol{t}} C(x).$$
Then $(\mathcal{T}_2, \mathcal{A}_\Xi) \models q$ but $(\mathcal{T}_1, \mathcal{A}_\Xi) \not\models q$, which is again a contradiction.

(sce$_\mathrm{b}$) $\Rightarrow$ **(pr)** Let Ξ be a set of precisely $\mathcal{T}_1$-realisable $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types. Consider

$$\mathcal{T}_\Xi = \{\top \sqsubseteq \bigsqcup_{\boldsymbol{t} \in \Xi} \bigsqcap_{C \in \boldsymbol{t}} C\}.$$

Clearly, $\mathcal{T}_1 \cup \mathcal{T}_\Xi \not\models \bigsqcap_{C \in \boldsymbol{t}} C \sqsubseteq \bot$, for every $\boldsymbol{t} \in \Xi$. Then, by **(sce$_\mathrm{b}$)**, $\mathcal{T}_2 \cup \mathcal{T}_\Xi \not\models \bigsqcap_{C \in \boldsymbol{t}} C \sqsubseteq \bot$ and thus, there is a model $\mathcal{I}_{\boldsymbol{t}}$ for $\mathcal{T}_2 \cup \mathcal{T}_\Xi$ realising $\boldsymbol{t}$. Take the disjoint union $\mathcal{I}$ of all these models $\mathcal{I}_{\boldsymbol{t}}$, for $\boldsymbol{t} \in \Xi$. It is easy to see that $\mathcal{I}$ is a model for $\mathcal{T}_2$ realising precisely the types in Ξ. ❑

DL-Lite$_{horn}$

Recall that a ΣQ*-concept in DL-Lite*$_{horn}$ is any concept of the form $\bot$, A_i or $\geq q\,R$, for some $A_i \in \Sigma$, Σ-role R and $q \in Q$. A ΣQ *concept inclusion in DL-Lite*$_{horn}$ is of the form $B_1 \sqcap \cdots \sqcap B_k \sqsubseteq B$, where $B_1, \dots, B_k, B$ are ΣQ-concepts in *DL-Lite*$_{horn}$. In what follows, the empty conjunction $\bigsqcap_{i\in\emptyset} B_i$ stands for $\top$.

Given a ΣQ-type $\boldsymbol{t}$, let

$$\boldsymbol{t}^+ = \{B \in \boldsymbol{t} \mid B \text{ a concept in } \textit{DL-Lite}_{horn}\},$$
$$\boldsymbol{t}^- = \{\neg B \in \boldsymbol{t} \mid B \text{ a concept in } \textit{DL-Lite}_{horn}\}.$$

Say that a ΣQ-type $\boldsymbol{t}_1$ is *h-contained* in a ΣQ-type $\boldsymbol{t}_2$ if $\boldsymbol{t}_1^+ \subseteq \boldsymbol{t}_2^+$.

Given a TBox $\mathcal{T}$ in *DL-Lite*$_{horn}$ and a ΣQ-type $\boldsymbol{t}$ with $\Sigma \subseteq sig(\mathcal{T})$ and $Q \subseteq Q_{\mathcal{T}}$, define the $\mathcal{T}$*-closure* $\mathsf{cl}_{\mathcal{T}}(\boldsymbol{t})$ of $\boldsymbol{t}$ as the $sig(\mathcal{T})Q_{\mathcal{T}}$-type $\mathsf{cl}_{\mathcal{T}}(\boldsymbol{t})$ such that $\mathsf{cl}_{\mathcal{T}}(\boldsymbol{t})^+$ consists of all $sig(\mathcal{T})Q_{\mathcal{T}}$-concepts B in *DL-Lite*$_{horn}$ with

$$\bigsqcap_{B_k \in \boldsymbol{t}^+} B_k \sqsubseteq B.$$

As we will see later in Theorem E.1, $\mathsf{cl}_{\mathcal{T}}(\boldsymbol{t})$ can be computed in polynomial time in the size of $\mathcal{T}$. Moreover, we have the following standard properties of Horn logic:

Lemma B.3 *Let $\mathcal{T}$ be a TBox in DL-Lite$_{horn}$. A ΣQ-type $\boldsymbol{t}$ is $\mathcal{T}$-realisable iff $\boldsymbol{t} = \mathsf{cl}_{\mathcal{T}}(\boldsymbol{t}) \restriction \Sigma Q$ and $\bot \notin \boldsymbol{t}$, where $\restriction \Sigma Q$ means restriction to ΣQ-types.*

Furthermore, $\mathcal{T}$ enjoys the 'disjunction property:' if $\mathcal{T} \models \bigsqcap B'_j \sqsubseteq \bigsqcup B_i$, where the B'_j and B_i are concepts in DL-Lite$_{horn}$, then there is some i such that $\mathcal{T} \models \bigsqcap B'_j \sqsubseteq B_i$.

We are now in a position to prove Theorem 6.

Lemma B.4 *For any DL-Lite$_{horn}$ TBoxes $\mathcal{T}_1$, $\mathcal{T}_2$ and any signature Σ, the following two conditions are equivalent:*

- *$\mathcal{T}_1$ Σ-concept entails $\mathcal{T}_2$ in DL-Lite$_{bool}$;*
- *$\mathcal{T}_1$ Σ-concept entails $\mathcal{T}_2$ in DL-Lite$_{horn}$.*

Proof. Suppose that $\mathcal{T}_1$ does not Σ-concept entail $\mathcal{T}_2$ in *DL-Lite*$_{bool}$ (without loss of generality, we assume that $\Sigma \subseteq sig(\mathcal{T}_1)$). By Lemma B.1, there exists a $\mathcal{T}_1$-realisable $\Sigma Q_{\mathcal{T}_1\cup\mathcal{T}_2}$-type $\boldsymbol{t}$ that is not $\mathcal{T}_2$-realisable. Consider now the $\mathcal{T}_1$ and $\mathcal{T}_2$-closures $\mathsf{cl}_{\mathcal{T}_1}(\boldsymbol{t})$ and $\mathsf{cl}_{\mathcal{T}_2}(\boldsymbol{t})$ of $\boldsymbol{t}$. Since $\boldsymbol{t}$ is $\mathcal{T}_1$-realisable, we have $\mathsf{cl}_{\mathcal{T}_1}(\boldsymbol{t}) \restriction \Sigma Q_{\mathcal{T}_1\cup\mathcal{T}_2} = \boldsymbol{t}$ by Lemma B.3. On the other hand, as $\boldsymbol{t}$ is not $\mathcal{T}_2$-realisable, $\boldsymbol{t}$ is properly h-contained in $\mathsf{cl}_{\mathcal{T}_2}(\boldsymbol{t}) \restriction \Sigma Q_{\mathcal{T}_1\cup\mathcal{T}_2}$, again by Lemma B.3. Therefore, there is $B \in (\mathsf{cl}_{\mathcal{T}_2}(\boldsymbol{t}) \setminus \mathsf{cl}_{\mathcal{T}_1}(\boldsymbol{t})) \restriction \Sigma Q_{\mathcal{T}_1\cup\mathcal{T}_2}$ such that

$$\mathcal{T}_1 \not\models \bigsqcap_{B_k \in \boldsymbol{t}^+} B_k \sqsubseteq B \quad \text{and} \quad \mathcal{T}_2 \models \bigsqcap_{B_k \in \boldsymbol{t}^+} B_k \sqsubseteq B,$$

which means that $\mathcal{T}_1$ does not Σ-concept entail $\mathcal{T}_2$ in *DL-Lite*$_{horn}$. ❑

We are now in a position to prove the uniform interpolation theorem for that *DL-Lite*$_{horn}$:

Theorem B.5 *Let $\mathcal{T}$ be a TBox in DL-Lite$_{horn}$, Σ a signature, and let $\mathcal{T}'$ be a finite presentation of the set*

$$\mathcal{S} = \{C_1 \sqsubseteq C_2 \mid \mathcal{T} \models C_1 \sqsubseteq C_2,$$
$$C_1 \sqsubseteq C_2 \text{ a } \Sigma Q_{\mathcal{T}}\text{-concept inclusion in } \textit{DL-Lite}_{horn}\}.$$

Then $\mathcal{T}'$ is a uniform interpolant for $\mathcal{T}$ w.r.t. Σ in DL-Lite$_{horn}$.

Proof. It is sufficient to show that $\mathcal{S} \models C_1 \sqsubseteq C_2$ whenever $\mathcal{T} \models C_1 \sqsubseteq C_2$, for every concept inclusion $C_1 \sqsubseteq C_2$ in *DL-Lite*$_{horn}$ with $sig(C_1 \sqsubseteq C_2) \cap sig(\mathcal{T}) \subseteq \Sigma$. Suppose, on the contrary, that $\mathcal{S} \not\models C_1 \sqsubseteq C_2$. Take a model $\mathcal{I}_1$ for $\mathcal{S}$ satisfying $C_1 \sqcap \neg C_2$ and realising all $\mathcal{S}$-realisable $\Sigma Q_{\mathcal{T}}$-types. Take a model $\mathcal{I}_2$ for $\mathcal{T}$ realising all $\mathcal{T}$-realisable $\Sigma Q_{\mathcal{T}}$-types. Then $\mathcal{I}_1$ and $\mathcal{I}_2$ realise exactly the same $\Sigma Q_{\mathcal{T}}$-types (this follows from the proof of Lemma B.4). Finally, we use Lemma A.1 according to which there exists a model $\mathcal{I}^*$ for $\mathcal{T}$ satisfying $C_1 \sqcap \neg C_2$, which is a contradiction. ❑

Let $\mathcal{I}$ and $\mathcal{I}'$ be interpretations and Σ a signature. A map f from $\Delta^{\mathcal{I}}$ into $\Delta^{\mathcal{I}'}$ is called a Σ*-homomorphism* if the following conditions hold for all $x, y \in \Delta^{\mathcal{I}}$:

- $f(a^{\mathcal{I}}) = a^{\mathcal{I}'}$, for every object name a,
- $x \in A^{\mathcal{I}}$ implies $f(x) \in A^{\mathcal{I}'}$, for every concept name A in Σ;
- $(x, y) \in P^{\mathcal{I}}$ implies $(f(x), f(y)) \in P^{\mathcal{I}'}$, for every role name P in Σ.

If there is a Σ-homomorphism from $\mathcal{I}$ to $\mathcal{I}'$, then $\mathcal{I}' \models q(\boldsymbol{a})$ follows from $\mathcal{I} \models q(\boldsymbol{a})$, for every Σ-query q in *DL-Lite*$_{horn}$ and tuple $\boldsymbol{a}$.

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. If this happens to be the case, we also say that $\mathcal{I}$ sub-precisely realises Ξ.

Lemma B.6 *For any TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ in DL-Lite$_{horn}$ and any signature Σ, the following conditions are equivalent:*

(qe$_\mathrm{h}$) *$\mathcal{T}_1$ Σ-query entails $\mathcal{T}_2$ in DL-Lite$_{horn}$;*

(spr) *if a set of $\Sigma Q_{\mathcal{T}_1\cup\mathcal{T}_2}$-types is precisely $\mathcal{T}_1$-realisable, then it is sub-precisely $\mathcal{T}_2$-realisable.*

Proof. **(spr)** $\Rightarrow$ **(qe$_\mathrm{h}$)** Let $Q_{12} = Q_{\mathcal{T}_1\cup\mathcal{T}_2}$. Assume that **(spr)** holds and $(\mathcal{T}_1, \mathcal{A}) \not\models q(\boldsymbol{a})$, for a Σ-ABox $\mathcal{A}$ and a Σ-query $q(\boldsymbol{a})$ in *DL-Lite*$_{horn}$. We show that $(\mathcal{T}_2, \mathcal{A}) \not\models q(\boldsymbol{a})$. Let $\mathcal{I}_1$ be a model for $(\mathcal{T}_1, \mathcal{A})$ with $\mathcal{I}_1 \not\models q(\boldsymbol{a})$. Let Ξ be the set of ΣQ_{12}-types realised in $\mathcal{I}_1$. Then, by **(spr)**, there exists a model $\mathcal{I}_2$ for $\mathcal{T}_2$ which realises all types in Ξ and such that every ΣQ_{12}-type realised in it is h-contained in some type from Ξ. We may assume that both $\mathcal{I}_1$ and $\mathcal{I}_2$ have a countably infinite domain.

Consider the model $\mathcal{I}_2^\omega$. We construct a model $\mathcal{I}^*$ for $(\mathcal{T}_2, \mathcal{A})$ with $\Delta^{\mathcal{I}^*} = \Delta^{\mathcal{I}_2^\omega}$ and a Σ-homomorphism $g\colon \Delta^{\mathcal{I}^*} \to \Delta^{\mathcal{I}_1}$. Then we shall have $\mathcal{I}^* \not\models q(\boldsymbol{a})$ and, therefore, $(\mathcal{T}_2, \mathcal{A}) \not\models q(\boldsymbol{a})$. Let

$$A^{\mathcal{I}^*} = A^{\mathcal{I}_2^\omega}, \quad \text{for all concept names } A,$$
$$P^{\mathcal{I}^*} = P^{\mathcal{I}_2^\omega}, \quad \text{for all } P \notin \Sigma.$$

Define a sequence

$$(g_i, \Delta_i, (P^i)_{P \in \Sigma}), \quad i \in \omega,$$

such that

- $\Delta_0 \subseteq \Delta_1 \subseteq \cdots \subseteq \Delta^{\mathcal{I}^*}$ and $\Delta^{\mathcal{I}^*} = \bigcup_{i \in \omega} \Delta_i$;
- $P^i \subseteq \Delta_i \times \Delta_i$;
- $g_i \colon \Delta_i \to \Delta^{\mathcal{I}_1}$ has the following properties:
  - if $(x, y) \in P^i$ then $(g_i(x), g_i(y)) \in P^{\mathcal{I}_1}$, for all $P \in \Sigma$ and all $x, y \in \Delta_i$;
  - for all $x \in \Delta_i$, the ΣQ_{12}-type of x in $\mathcal{I}_2^\omega$ is h-contained in the ΣQ_{12}-type of $g_i(x)$ in $\mathcal{I}_1$.

To start with, take a bijection $g_0 \colon \Delta_0 \to \Delta^{\mathcal{I}_1}$ which is invariant under ΣQ_{12}-types, where $\Delta_0 \subseteq \Delta^{\mathcal{I}_2^\omega}$ is chosen in such a way that each ΣQ_{12}-type realised in $\mathcal{I}_2^\omega$ is realised infinitely often in $\Delta^{\mathcal{I}_2^\omega} \setminus \Delta_0$. Such a bijection exists because $\mathcal{I}_2^\omega$ realises every ΣQ_{12}-type (in particular those from $\mathcal{I}_1$) infinitely often. Let

$$a^{\mathcal{I}^*} = g_0^{-1}(a^{\mathcal{I}_1}),$$

for every object name a, and

$$P^0 = \{(g_0^{-1}(x), g_0^{-1}(y)) \mid (x, y) \in P^{\mathcal{I}_1}\},$$

for every $P \in \Sigma$. Assume that an ordering $<$ of $\Delta^{\mathcal{I}_2^\omega} \setminus \Delta_0$ is isomorphic to ω, and suppose that Δ_k, g_k, and P^k, $P \in \Sigma$, have already been constructed. To construct Δ_{k+1}, g_{k+1}, and P^{k+1}, we apply one of the following three rules to $x \in \Delta^{\mathcal{I}_2^\omega} \setminus \Delta_0$, provided that none of these rules is applicable to any $y \in \Delta^{\mathcal{I}_2^\omega} \setminus \Delta_0$ with $y < x$.

- If $x \in \Delta_k$ and the ΣQ_{12}-type of x in $\mathcal{I}_2^\omega$ contains $\geq q\, P$, for $q \in Q_{12}$ and $P \in \Sigma$, such that P^k contains fewer than q pairs (x, x_i), pick a point $y \in \Delta^{\mathcal{I}^*} \setminus \Delta_k$ which has the same ΣQ_{12}-type as a point $z \in \Delta^{\mathcal{I}_1}$ with $(g_k(x), z) \in P^{\mathcal{I}_1}$ (this can be done since the ΣQ_{12}-type of x is h-contained in the ΣQ_{12}-type of $g(x)$). Then we set $\Delta_{k+1} = \Delta_k \cup \{y\}$, $g_{k+1} = g_k \cup \{(y, z)\}$, and $P^{k+1} = P^k \cup \{(x, y)\}$.
- If $x \in \Delta_k$ and the ΣQ_{12}-type of x in $\mathcal{I}_2^\omega$ contains $\geq q\, P^-$, for $q \in Q_{12}$ and $P \in \Sigma$, such that P^k contains fewer than q pairs (x_i, x), pick a point $y \in \Delta^{\mathcal{I}^*} \setminus \Delta_k$ which has the same ΣQ_{12}-type as a point $z \in \Delta^{\mathcal{I}_1}$ with $(z, g_k(x)) \in P^{\mathcal{I}_1}$. Then we set $\Delta_{k+1} = \Delta_k \cup \{y\}$, $g_{k+1} = g_k \cup \{(y, z)\}$, and $P^{k+1} = P^k \cup \{(y, x)\}$.
- If $x \in \Delta^{\mathcal{I}^*} \setminus \Delta_k$, select $z \in \Delta^{\mathcal{I}_1}$ such that the ΣQ_{12}-type of x in $\mathcal{I}_2^\omega$ is h-contained in the ΣQ_{12}-type of z in $\mathcal{I}_1$. Set $\Delta_{k+1} = \Delta_k \cup \{x\}$ and $g_{k+1} = g_k \cup \{(x, z)\}$, and $P_{k+1} = P_k$.

Clearly, for $P^{\mathcal{I}^*} = \bigcup_{i \in \omega} P^i$, $P \in \Sigma$, the function $g = \bigcup_{i \in \omega} g_i$ is a Σ-homomorphism from $\Delta^{\mathcal{I}^*}$ to $\Delta^{\mathcal{I}_1}$ and the ΣQ_{12}-type of each x is the same in $\mathcal{I}^*$ and $\mathcal{I}_2^\omega$. Hence $\mathcal{I}^*$ is a model for $\mathcal{T}_2$. Moreover, $\mathcal{I}^*$ is a model for $\mathcal{A}$ because $\mathcal{I}_1$ is a model for $\mathcal{A}$.

(qe$_\mathbf{h}$) $\Rightarrow$ (spr) Suppose that **(spr)** does not hold for Ξ. Then two cases are possible:

1. For every model $\mathcal{I}$ for $\mathcal{T}_2$, there is $\boldsymbol{t} \in \Xi$ with $(\bigsqcap_{B \in \boldsymbol{t}^+} B)^{\mathcal{I}} = \emptyset$. Consider the query $q = \bot$ and the ABox $\mathcal{A}_\Xi = \{B(a_{\boldsymbol{t}}) \mid \boldsymbol{t} \in \Xi,\ \ B \in \boldsymbol{t}^+\}$, where $a_{\boldsymbol{t}}$ is a fresh object name, for each $\boldsymbol{t} \in \Xi$. Then $(\mathcal{T}_1, \mathcal{A}_\Xi) \not\models q$, while $(\mathcal{T}_2, \mathcal{A}_\Xi) \models q$.
2. Case 1 does not hold. Consider $\mathcal{A}_\Xi = \{B(a_{\boldsymbol{t}}) \mid \boldsymbol{t} \in \Xi,\ \ B \in \boldsymbol{t}^+\}$, $a_{\boldsymbol{t}}$ an object name, for each $\boldsymbol{t} \in \Xi$. Then, for every model $\mathcal{I}'$ for $(\mathcal{T}_2, \mathcal{A}_\Xi)$, there is a $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-type $\boldsymbol{t}$ that is realised in $\mathcal{I}'$ and not h-contained in any type from Ξ. Let Θ be the set of all such $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types and consider the query
$$q \quad = \quad \exists x \bigvee_{\boldsymbol{t} \in \Theta} \bigwedge_{B \in \boldsymbol{t}^+} B(x).$$
Then $(\mathcal{T}_1, \mathcal{A}_\Xi) \not\models q$ but $(\mathcal{T}_2, \mathcal{A}_\Xi) \models q$.

This completes the proof of the lemma. ❑

The following model-theoretic property of TBoxes in *DL-Lite$_{horn}$* is standard in Horn logic; see, e.g., (Artale *et al.* 2007).

Lemma B.7 *Let $\mathcal{T}$ be a TBox in DL-Lite$_{horn}$ and $\boldsymbol{t}$ a $\mathcal{T}$-realisable ΣQ-type. Then there exists (at most countable) model $\mathcal{J}_{\mathcal{T}}(\boldsymbol{t})$ for $\mathcal{T}$ such that $\mathcal{J}_{\mathcal{T}}(\boldsymbol{t})$ realises $\boldsymbol{t}$ and, for every model $\mathcal{I}$ for $\mathcal{T}$ realising $\boldsymbol{t}$, there exists a Σ-homomorphism $h \colon \mathcal{J}_{\mathcal{T}}(\boldsymbol{t}) \to \mathcal{I}$.*

In what follows we fix some model $\mathcal{J}_{\mathcal{T}}(\boldsymbol{t})$ mentioned in the formulation of the lemma and call it the *minimal model for $\mathcal{T}$ realising $\boldsymbol{t}$*. As an immediate consequence of Lemma B.7 we obtain the following:

Lemma B.8 *Let $\mathcal{T}$ be a TBox in DL-Lite$_{horn}$ and $\boldsymbol{t}$ a $\mathcal{T}$-realisable ΣQ-type. Then there exists a countable model $\mathcal{J}_{\mathcal{T}}(\boldsymbol{t})$ for $\mathcal{T}$ such that $\mathcal{J}_{\mathcal{T}}(\boldsymbol{t})$ realises $\boldsymbol{t}$ and, for every model $\mathcal{I}$ for $\mathcal{T}$ realising $\boldsymbol{t}$, there is a Σ-homomorphism $h \colon \mathcal{I} \oplus \mathcal{J}_{\mathcal{T}}(\boldsymbol{t}) \to \mathcal{I}$. In particular,*

- *all ΣQ-types that are realised in $\mathcal{I}$ are also realised in $\mathcal{I} \oplus \mathcal{J}_{\mathcal{T}}(\boldsymbol{t})$;*
- *every ΣQ-type realised in $\mathcal{I} \oplus \mathcal{J}_{\mathcal{T}}(\boldsymbol{t})$ is h-contained in some ΣQ-type realised in $\mathcal{I}$;*
- *if $\mathcal{I} \oplus \mathcal{J}_{\mathcal{T}}(\boldsymbol{t}) \models q(\boldsymbol{a})$ then $\mathcal{I} \models q(\boldsymbol{a})$, for every Σ-query $q(\boldsymbol{a})$ in DL-Lite$_{horn}$.*

Finally, we show the equivalence of (2) and (3) in Theorem 7 for *DL-Lite$_{horn}$*. Given a realisable set Ξ of ΣQ-types, define the TBox $\mathcal{T}_\Xi$ *induced by* Ξ by taking all ΣQ-concept inclusions

$$B_1 \sqcap \cdots \sqcap B_k \sqsubseteq B,$$

where $B_1, \ldots, B_k, B$ are distinct ΣQ-concepts in *DL-Lite$_{horn}$* such that, for all $\boldsymbol{t} \in \Xi$,

$$\text{if} \quad B_1, \ldots, B_k \in \boldsymbol{t}^+ \quad \text{then} \quad B \in \boldsymbol{t}^+.$$

Note that (i) if, for distinct ΣQ-concepts $B_1, \ldots, B_k$, there is no $\boldsymbol{t} \in \Xi$ with $B_1, \ldots, B_k \in \boldsymbol{t}^+$ then $B_1 \sqcap \cdots \sqcap B_k \sqsubseteq \bot$ is in $\mathcal{T}_\Xi$, and (ii) if $B \in \boldsymbol{t}^+$, for all $\boldsymbol{t} \in \Xi$, then $\top \sqsubseteq B \in \mathcal{T}_\Xi$.

Lemma B.9 *Let Ξ be a set of ΣQ-types and $\boldsymbol{t}_0$ a ΣQ-type. Let $\Lambda_{\boldsymbol{t}_0} = \{\boldsymbol{t} \in \Xi \mid \boldsymbol{t}_0^+ \subseteq \boldsymbol{t}^+\}$. Then $\boldsymbol{t}_0$ is $\mathcal{T}_\Xi$-realisable iff $\Lambda_{\boldsymbol{t}_0} \neq \emptyset$ and $\boldsymbol{t}_0^+ = \bigcap_{\boldsymbol{t} \in \Lambda_{\boldsymbol{t}_0}} \boldsymbol{t}^+$.*

Proof. ($\Rightarrow$) Clearly, if $\Lambda_{\boldsymbol{t}_0} = \emptyset$ then $\bigsqcap_{B \in \boldsymbol{t}_0^+} B \sqsubseteq \bot$ is in $\mathcal{T}_\Xi$, and so $\boldsymbol{t}_0$ cannot be $\mathcal{T}_\Xi$-realisable. If $\Lambda_{\boldsymbol{t}_0} \neq \emptyset$ then, for every $B' \in \bigcap_{\boldsymbol{t} \in \Lambda_{\boldsymbol{t}_0}} \boldsymbol{t}^+$, we have $(\bigsqcap_{B \in \boldsymbol{t}_0^+} \sqsubseteq B') \in \mathcal{T}_\Xi$. So $\boldsymbol{t}^+ \supseteq \bigcap_{\boldsymbol{t} \in \Lambda_{\boldsymbol{t}_0}} \boldsymbol{t}^+$.

($\Leftarrow$) If there is no model for $\mathcal{T}_\Xi$ realising $\boldsymbol{t}_0$, then we have $\mathcal{T}_\Xi \models \bigsqcap_{B \in \boldsymbol{t}_0^+} B \sqsubseteq \bigsqcup_{\neg B \in \boldsymbol{t}_0^-} B$. But then, by Lemma B.3, $\mathcal{T}_\Xi \models \bigsqcap_{B \in \boldsymbol{t}_0^+} \sqsubseteq B'$, for some $\neg B' \in \boldsymbol{t}_0^-$. Therefore, $\bigsqcap_{B \in \boldsymbol{t}_0^+} B \sqsubseteq B' \in \mathcal{T}_\Xi$, and so B' must be in $\boldsymbol{t}^+$, which is impossible. ❑

A set Ξ of ΣQ-types is said to be *meet-precisely $\mathcal{T}$-realisable* if there is a model $\mathcal{I}$ for $\mathcal{T} \cup \mathcal{T}_\Xi$ such that $\mathcal{I}$ realises all types from Ξ. (It follows that every ΣQ-type realised in $\mathcal{I}$ is h-contained in a type from Ξ.) As we shall see later, given a type $\boldsymbol{t}$, the $\mathcal{T} \cup \mathcal{T}_\Xi$-closure $\mathsf{cl}_{\mathcal{T} \cup \mathcal{T}_\Xi}(\boldsymbol{t})$ can be computed in polynomial type in the size of $\mathcal{T}$, Ξ and $\boldsymbol{t}$.

Lemma B.10 *Let Ξ be the set of ΣQ-types that is precisely realised in a model $\mathcal{I}$ for a TBox $\mathcal{T}$. Then*

1. $\mathcal{I} \models \mathcal{T}_\Xi$;
2. $\mathcal{T} \models \bigsqcap_k B_k \sqsubseteq B$ *implies* $\mathcal{T}_\Xi \models \bigsqcap_k B_k \sqsubseteq B$ *(which holds iff $\bigsqcap_k B_k \sqsubseteq B \in \mathcal{T}_\Xi$), for all ΣQ-concept inclusions $\bigsqcap_k B_k \sqsubseteq B$ in DL-Lite$_{horn}$.*

Proof. The first claim is obvious. To show the second one, assume that $\mathcal{T} \models \bigsqcap_k B_k \sqsubseteq B$, but $\mathcal{T}_\Xi \not\models \bigsqcap_k B_k \sqsubseteq B$. Then $\bigsqcap_k B_k \sqsubseteq B \notin \mathcal{T}_\Xi$, and so there is a type $\boldsymbol{t} \in \Xi$ such that $B_k \in \boldsymbol{t}$, while $B \notin \boldsymbol{t}$. But this means that $\mathcal{I} \not\models \bigsqcap_k B_k \sqsubseteq B$, which is impossible, since $\mathcal{I} \models \mathcal{T}$. ❑

Lemma B.11 *For any TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ in DL-Lite$_{horn}$ and any signature Σ, the following conditions are equivalent*:

(sce$_\mathbf{h}$) *$\mathcal{T}_1$ strongly Σ-concept entails $\mathcal{T}_2$ in DL-Lite$_{horn}$*;

(sqe$_\mathbf{h}$) *$\mathcal{T}_1$ strongly Σ-query entails $\mathcal{T}_2$ in DL-Lite$_{horn}$*;

(mpr) *if a set of $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types is precisely $\mathcal{T}_1$-realisable, then it is meet-precisely $\mathcal{T}_2$-realisable.*

Proof. The implication **(sqe$_\mathbf{h}$)** $\Rightarrow$ **(sce$_\mathbf{h}$)** is trivial.

(mpr) $\Rightarrow$ **(sqe$_\mathbf{h}$)** Let $\mathcal{I}$ be a model for $\mathcal{T}_1$ and Ξ the set of $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types realised in $\mathcal{I}$. Consider

$$\mathcal{J}_\mathcal{I} \quad = \quad \mathcal{I} \quad \oplus \quad \bigoplus_{\boldsymbol{t} \in \Theta_\mathcal{I}} \mathcal{J}_{\mathcal{T}_1 \cup \mathcal{T}_\Xi}(\boldsymbol{t}),$$

where $\Theta_\mathcal{I}$ is the set of all $\mathcal{T}_1 \cup \mathcal{T}_\Xi$-realisable $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types. Observe that $\mathcal{J}_\mathcal{I}$ precisely realises $\Theta_\mathcal{I}$: indeed, $\mathcal{J}_\mathcal{I}$ realises every $\mathcal{T}_1 \cup \mathcal{T}_\Xi$-realisable $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-type and conversely, every $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-type realised in $\mathcal{J}_\mathcal{I}$, is $\mathcal{T}_1 \cup \mathcal{T}_\Xi$-realisable.

By **(mpr)**, there exists, for the set $\Theta_\mathcal{I}$ of $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types, a model $\mathcal{I}'$ for $\mathcal{T}_2$ that realises all types in $\Theta_\mathcal{I}$ and such that each $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-type realised in $\mathcal{I}'$ is $\mathcal{T}_{\Theta_\mathcal{I}}$-realisable. We claim that $\mathcal{I}'$ realises precisely the set $\Theta_\mathcal{I}$ of $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types. Indeed, $\mathcal{I}'$ realises every type from $\Theta_\mathcal{I}$; conversely, let $\boldsymbol{t}$ be a $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-type realised in $\mathcal{I}'$; as $\boldsymbol{t}$ is $\mathcal{T}_{\Theta_\mathcal{I}}$-realisable, by Lemma B.9, there are $\boldsymbol{t}_1, \dots, \boldsymbol{t}_k \in \Theta_\mathcal{I}$ with $\boldsymbol{t}^+ = \bigcap_k \boldsymbol{t}_i^+$; as the $\boldsymbol{t}_k$ are all realised in $\mathcal{J}_\mathcal{I}$, they are $\mathcal{T}_1 \cup \mathcal{T}_\Xi$-realisable and therefore, $\boldsymbol{t}$ is $\mathcal{T}_1 \cup \mathcal{T}_\Xi$-realisable and $\boldsymbol{t} \in \Theta_\mathcal{I}$.

Suppose now that **(sqe$_\mathbf{h}$)** does not hold and let $\mathcal{T}$ be a Σ-TBox, $\mathcal{A}$ a Σ-ABox and $q(\boldsymbol{x})$ a Σ-query in *DL-Lite$_{horn}$* such that $(\mathcal{T}_2 \cup \mathcal{T}, \mathcal{A}) \models q(\boldsymbol{a})$ but $(\mathcal{T}_1 \cup \mathcal{T}, \mathcal{A}) \not\models q(\boldsymbol{a})$. Let $\mathcal{I}$ be a model of $(\mathcal{T}_1 \cup \mathcal{T}, \mathcal{A})$ with $\mathcal{I} \not\models q(\boldsymbol{a})$. By Lemma B.10, $\mathcal{I} \models \mathcal{T}_\Xi$. By Lemma B.8, applied to $\mathcal{T}_1 \cup \mathcal{T}_\Xi$, $\mathcal{J}_\mathcal{I} \not\models q(\boldsymbol{a})$. But then we can apply Lemma A.3 to $\mathcal{J}_\mathcal{I}$ and $\mathcal{I}'$ and find a model $\mathcal{I}^*$ for $\mathcal{T}_2$ that is Σ-isomorphic to $\mathcal{J}_\mathcal{I}^\omega$. As $\mathcal{J}_\mathcal{I} \models \mathcal{T}$, it follows that $\mathcal{I}^*$ is a model for $(\mathcal{T}_2 \cup \mathcal{T}, \mathcal{A})$ such that $\mathcal{I}^* \not\models q(\boldsymbol{a})$, which is a contradiction.

(sce$_\mathbf{h}$) $\Rightarrow$ **(mpr)** Let Ξ be a set of precisely $\mathcal{T}_1$-realisable $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types. We claim that, for all $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$ concept inclusions $C \sqsubseteq \bigsqcup B'_k$ with $C = \bigsqcap B_k$,

$$\mathcal{T}_2 \cup \mathcal{T}_\Xi \models C \sqsubseteq \bigsqcup B'_k \qquad \text{iff} \qquad \mathcal{T}_1 \models C \sqsubseteq \bigsqcup B'_k. \quad (2)$$

Indeed, if $\mathcal{T}_2 \cup \mathcal{T}_\Xi \models C \sqsubseteq \bigsqcup B'_k$ then, by Lemma B.3, there is j with $\mathcal{T}_2 \cup \mathcal{T}_\Xi \models C \sqsubseteq B'_j$ and, by **(sce$_\mathbf{h}$)**, we have $\mathcal{T}_1 \cup \mathcal{T}_\Xi \models C \sqsubseteq B'_j$. Then, by Lemma B.10, $\mathcal{T}_1 \models C \sqsubseteq B'_j$, from which the claim follows. The converse implication is obvious.

Clearly, for each $\boldsymbol{t} \in \Xi$, we have

$$\mathcal{T}_1 \not\models \bigsqcap_{B \in \boldsymbol{t}^+} B \sqsubseteq \bigsqcup_{\neg B \in \boldsymbol{t}^-} B,$$

and in view of (2), there is a model for $\mathcal{T}_2 \cup \mathcal{T}_\Xi$ realising $\boldsymbol{t}$. Take the disjoint union $\mathcal{I}'$ of all models $\mathcal{I}_{\boldsymbol{t}}$, for $\boldsymbol{t} \in \Xi$. Clearly, $\mathcal{I}'$ realises all types in Ξ and each $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-type $\boldsymbol{t}$ realised in it is Ξ-realisable. ❑

C Robustness

Here we prove Theorem 13, which is formulated as follows:

Let 'Σ-entails' be one of the eight notions of Σ-entailment given in Definition 1.

- *The relation 'Σ-entails' is robust under vocabulary extensions: if $\mathcal{T}_1$ Σ-entails $\mathcal{T}_2$, then $\mathcal{T}_1$ Σ'-entails $\mathcal{T}_2$, for every Σ' such that $\Sigma' \cap sig(\mathcal{T}_2) \subseteq \Sigma$.*
- *The relation 'Σ'-entails' is robust under joins: if $\mathcal{T}$ and $\mathcal{T}_i$ Σ-entail each other, for $i = 1, 2$, and $sig(\mathcal{T}_1) \cap sig(\mathcal{T}_2) \subseteq \Sigma$, then $\mathcal{T}$ Σ-entails $\mathcal{T}_1 \cup \mathcal{T}_2$.*

Proof. Throughout this proof we use the obvious fact that instead of $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$ types in the semantic criteria for Σ-entailment between $\mathcal{T}_1$ and $\mathcal{T}_2$ (Theorem 11) one can take ΣQ-types, for any set $Q \supseteq Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$.

We start by considering robustness under joins.

(a) *Σ-concept entailment in DL-Lite$_{bool}$*. Suppose that $\mathcal{T}$ and $\mathcal{T}_i$ Σ-concept entail each other in *DL-Lite$_{bool}$*, for $i = 1, 2$. Consider a $\mathcal{T}$-realisable ΣQ-type $\boldsymbol{t}$ with $Q = Q_{\mathcal{T} \cup \mathcal{T}_1 \cup \mathcal{T}_2}$. By Theorem 11, it is sufficient to show that $\boldsymbol{t}$ is $\mathcal{T}_1 \cup \mathcal{T}_2$-realisable. Let Ξ be the set of all $\mathcal{T}$-realisable ΣQ-types. As $\mathcal{T}$ and $\mathcal{T}_i$ Σ-concept entail each other, Ξ is also the set of all $\mathcal{T}_i$-realisable ΣQ-types, for $i = 1, 2$. It follows that Ξ is precisely $\mathcal{T}_i$-realisable, for $i = 1, 2$. Using Lemma A.1, we obtain a model for $\mathcal{T}_1 \cup \mathcal{T}_2$ precisely realising Ξ. This model realises $\boldsymbol{t}$.

(b) *Σ-concept entailment in DL-Lite$_{horn}$*. This case follows from (a) by Theorem 6.

(c) Σ-*query entailment in DL-Lite$_{bool}$* (and, equivalently, strong Σ-concept entailment and strong Σ-query entailment). Suppose that $\mathcal{T}$ and $\mathcal{T}_i$ Σ-query entail each other in *DL-Lite$_{bool}$*, for $i = 1, 2$, and let Ξ be a precisely $\mathcal{T}$-realisable set of ΣQ-types, where $Q = Q_{\mathcal{T}\cup\mathcal{T}_1\cup\mathcal{T}_2}$. By Theorem 11, it is sufficient to show that Ξ is precisely $\mathcal{T}_1 \cup \mathcal{T}_2$-realisable. By Theorem 11, Ξ is precisely $\mathcal{T}_i$-realisable, for $i = 1, 2$. Using Lemma A.1, we obtain a model for $\mathcal{T}_1 \cup \mathcal{T}_2$ precisely realising Ξ.

(d) Σ-*query entailment in DL-Lite$_{horn}$*. Suppose that $\mathcal{T}$ and $\mathcal{T}_i$ Σ-query entail each other in *DL-Lite$_{horn}$*, for $i = 1, 2$, and let Ξ be a precisely $\mathcal{T}$-realisable set of ΣQ-types, where $Q = Q_{\mathcal{T}\cup\mathcal{T}_1\cup\mathcal{T}_2}$. By Theorem 11, it is sufficient to show that Ξ is sub-precisely $\mathcal{T}_1 \cup \mathcal{T}_2$-realisable. As $\mathcal{T}$ and $\mathcal{T}_i$ mutually Σ-query entail each other, we obtain a set $\Xi' \supseteq \Xi$ which is precisely $\mathcal{T}$-, $\mathcal{T}_1$-, and $\mathcal{T}_2$-realisable and such that each $\boldsymbol{t} \in \Xi'$ is h-contained in a type from Ξ. By Lemma A.1, we obtain a model for $\mathcal{T}_1 \cup \mathcal{T}_2$ precisely realising Ξ'. But then Ξ is sub-precisely $\mathcal{T}_1 \cup \mathcal{T}_2$-realisable.

(e) *Strong Σ-query entailment in DL-Lite$_{horn}$* (and, equivalently, strong Σ-concept entailment). Suppose that $\mathcal{T}$ and $\mathcal{T}_i$ strongly Σ-query entail each other in *DL-Lite$_{horn}$*, for $i = 1, 2$, and let Ξ be a precisely $\mathcal{T}$-realisable set of ΣQ-types, where $Q = Q_{\mathcal{T}\cup\mathcal{T}_1\cup\mathcal{T}_2}$. Let $\mathcal{I}$ be a model for $\mathcal{T}$ precisely realising Ξ. Consider the set $\Theta_{\mathcal{I}}$ of ΣQ-types constructed in the proof of Lemma B.11, **(mpr)** $\Rightarrow$ **(sce$_\mathbf{h}$)**, with $\mathcal{T}_1$ replaced by $\mathcal{T}$. The set $\Theta_{\mathcal{I}}$ is precisely $\mathcal{T}_i$-realisable, for $i = 1, 2$. Hence, by Lemma A.1, there exists a model for $\mathcal{T}_1 \cup \mathcal{T}_2$ precisely realising $\Theta_{\mathcal{I}}$. This model meet-precisely realises Ξ.

Let us consider now robustness under vocabulary extensions.

(a) Σ-*concept entailment in DL-Lite$_{bool}$*. This case follows from uniform interpolation of *DL-Lite$_{bool}$*.

(b) Σ-*concept entailment in DL-Lite$_{horn}$*. This case follows from uniform interpolation of *DL-Lite$_{horn}$*.

(c) Σ-*query entailment in DL-Lite$_{bool}$* (and, equivalently, strong Σ-concept entailment and strong Σ-query entailment). Suppose that $\mathcal{T}_1$ Σ-query entails $\mathcal{T}_2$ and Σ' is a signature with $sig(\mathcal{T}_2) \cap \Sigma' \subseteq \Sigma$. Assume that Ξ is a $\mathcal{T}_1$-precisely realisable set of $\Sigma' Q$-types, $Q = Q_{\mathcal{T}_1\cup\mathcal{T}_2}$. Let Ξ_0 be the set of restrictions of types in Ξ to ΣQ-concepts. There exists a model for $\mathcal{T}_2$ precisely realising Ξ_0, and there exists a model precisely realising Ξ. Using Lemma A.1, we then obtain a model for $\mathcal{T}_2$ precisely realising Ξ.

(d) Σ-*query entailment in DL-Lite$_{horn}$*. Suppose that $\mathcal{T}_1$ Σ-query entails $\mathcal{T}_2$ and Σ' is a signature with $sig(\mathcal{T}_2) \cap \Sigma' \subseteq \Sigma$. Assume that Ξ is a $\mathcal{T}_1$-precisely realisable set of $\Sigma' Q$-types, $Q = Q_{\mathcal{T}_1\cup\mathcal{T}_2}$. Let Ξ_0 be the set of restrictions of types in Ξ to ΣQ-concepts. There exists a model $\mathcal{I}_2$ for $\mathcal{T}_2$ sub-precisely realising Ξ_0. Let $\Xi_0' \supseteq \Xi_0$ be the set of ΣQ-types realised in $\mathcal{I}_2$. Expand every type $\boldsymbol{t} \in \Xi_0'$ to the $\Sigma' Q$-type $\boldsymbol{t}_0 \supseteq \boldsymbol{t}$ such that $\boldsymbol{t}^+ = \boldsymbol{t}_0^+$ and denote the resulting set of $\Sigma' Q$-types by Ξ_0''. Clearly, there exists a model $\mathcal{I}_1$ precisely realising $\Xi \cup \Xi_0''$. Observe that $\mathcal{I}_1$ and $\mathcal{I}_2$ realise the same ΣQ-types. Using Lemma A.1, we can construct a model for $\mathcal{T}_2$ precisely realising $\Xi \cup \Xi_0''$, and so sub-precisely realising Ξ.

(e) *Strong Σ-query entailment in DL-Lite$_{horn}$* (and, equivalently, strong Σ-concept entailment). Suppose that $\mathcal{T}_1$ strongly Σ-concept entails $\mathcal{T}_2$ and Σ' is a signature with $sig(\mathcal{T}_2) \cap \Sigma' \subseteq \Sigma$. Assume that Ξ is a $\mathcal{T}_1$-precisely realisable set of $\Sigma' Q$-types, $Q = Q_{\mathcal{T}_1\cup\mathcal{T}_2}$. Let $\mathcal{I}$ be a model for $\mathcal{T}_1$ precisely realising Ξ. Consider the sets $\Theta_{\mathcal{I},\Sigma}$ and $\Theta_{\mathcal{I},\Sigma'}$ of, respectively, ΣQ- and $\Sigma' Q$-types constructed in the proof of Lemma B.11, **(mpr)** $\Rightarrow$ **(sce$_\mathbf{h}$)**, using Σ' instead of Σ in the definition of $\Theta_{\mathcal{I},\Sigma'}$. It is not difficult to show that $\Theta_{\mathcal{I},\Sigma}$ coincides with the set of restrictions of types in $\Theta_{\mathcal{I},\Sigma'}$ to ΣQ-concepts. Hence, the set of restrictions of types in $\Theta_{\mathcal{I},\Sigma'}$ to ΣQ-concepts is precisely $\mathcal{T}_2$-realisable. Using Lemma A.1, we obtain a model for $\mathcal{T}_2$ precisely realising $\Theta_{\mathcal{I},\Sigma}$. This model meet-precisely realises Ξ. ❑

D Decision procedures and complexity for *DL-Lite$_{bool}$*

Here we prove the complexity results for *DL-Lite$_{bool}$* stated in Theorem 15. Recall the following result from (Artale *et al.* 2007):

Theorem D.1 *The problem whether $\mathcal{T} \models C_1 \sqsubseteq C_2$ holds in DL-Lite$_{bool}$ is* coNP*-complete.*

We use this result and Lemma B.1 to prove

Theorem D.2 *Σ-concept entailment for TBoxes in DL-Lite$_{bool}$ is Π_2^p-complete.*

Proof. Let $\mathcal{T}_1$, $\mathcal{T}_2$ be TBoxes in *DL-Lite$_{bool}$* and Σ a signature. Without loss of generality, we may assume that $\Sigma \subseteq sig(\mathcal{T}_1 \cup \mathcal{T}_2)$. Here is a Σ_2^p algorithm deciding whether $\mathcal{T}_1$ *does not* Σ-concept entail $\mathcal{T}_2$:

1. Guess a $\Sigma Q_{\mathcal{T}_1\cup\mathcal{T}_2}$-type $\boldsymbol{t}$. (Observe that the size of $\boldsymbol{t}$ is linear in the size of $\mathcal{T}_1 \cup \mathcal{T}_2$.)
2. Check, by calling an NP-oracle, whether (i) $\boldsymbol{t}$ is $\mathcal{T}_1$-realisable and whether (ii) $\boldsymbol{t}$ is not $\mathcal{T}_2$-realisable. Such an oracle exists by Theorem D.1.
3. Return '$\mathcal{T}_1$ does not Σ-concept entails $\mathcal{T}_2$' if the answers to (i) and (ii) are both positive.

By Lemma B.1, $\mathcal{T}_1$ does not Σ-concept entails $\mathcal{T}_2$ if, and only if, the algorithm says so. ❑

To prove the other complexity results for *DL-Lite$_{bool}$* from Theorem 15, we reduce precise realisability of a set of types (as stated in criterion **(pr)** of Lemma B.2) to a satisfiability problem in propositional logic. Let Σ be a signature and Q a set of positive natural numbers containing 1. With every basic concept B of the form A or $\geq q\, R$ we associate a fresh *propositional variable* B^*, and, for a concept C in *DL-Lite$_{bool}$*, denote by C^* the result of replacing each B in it with B^* (and $\sqcap$, $\sqcup$ with $\wedge$, $\vee$, respectively), for a ΣQ-type $\boldsymbol{t}$, denote by $\boldsymbol{t}^*$ the set $\{C^* \mid C \in \boldsymbol{t}\}$, and, for a TBox $\mathcal{T}$, denote by $\mathcal{T}^*$ the set $\{C_1^* \to C_2^* \mid C_1 \sqsubseteq C_2 \in \mathcal{T}\}$. Thus, C^* is a formula and $\boldsymbol{t}^*$, $\mathcal{T}^*$ are sets of formulas of *propositional logic*.

The following result follows immediately from (Artale *et al.* 2007):

Lemma D.3 *Let $\mathcal{T}$ be a TBox in DL-Lite$_{bool}$, $Q \supseteq Q_{\mathcal{T}}$, and Ω be a set of roles closed under inverse and containing all sig($\mathcal{T}$)-roles. Then a set Ξ of ΣQ-types is precisely $\mathcal{T}$-realisable iff there is a set $\Omega_0 \subseteq \Omega$ closed under inverse and such that:*

(t) *for each $\boldsymbol{t} \in \Xi$, $\boldsymbol{t}^* \cup \mathsf{Ax}(\mathcal{T}, \Omega_0)$ is satisfiable;*

(pw) *for each $R \in \Omega_0$, there is $\boldsymbol{t}_R \in \Xi$ such that $\boldsymbol{t}_R^* \cup \{(\geq 1\, R)^*\} \cup \mathsf{Ax}(\mathcal{T}, \Omega_0)$ is satisfiable,*

where

$$\mathsf{Ax}(\mathcal{T}, \Omega_0) \; = \; \mathcal{T}^* \;\; \cup \;\; \{\neg(\geq 1\, R)^* \mid R \in \Omega \setminus \Omega_0\} \;\; \cup \;\; \{(\geq q\, R)^* \to (\geq q'\, R)^* \mid R \in \Omega,\; q, q' \in Q,\; q > q'\}.$$

It follows, in particular, that, given $\mathcal{T}$ and a set Ξ of ΣQ-types, precise $\mathcal{T}$-realisability of Ξ is decidable in NP. To prove the complexity upper bound stated in Theorem 7, it will be sufficient to show that it is enough to consider sets Ξ of polynomial size in the size of $\mathcal{T}$.

Lemma D.4 *Suppose that a set Ξ of $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types is precisely $\mathcal{T}_1$-realisable but not precisely $\mathcal{T}_2$-realisable. Let Ω be the set of role names and their inverses that occur in $\mathcal{T}_1 \cup \mathcal{T}_2$. Then there is some $\Theta \subseteq \Xi$ with $|\Theta| \leq |\Omega| + 1$ such that Θ is precisely $\mathcal{T}_1$-realisable but not precisely $\mathcal{T}_2$-realisable.*

Proof. The proof follows from Lemmas D.3. For every $\boldsymbol{t} \in \Xi$, there is $\Omega_0 \subseteq \Omega$ such that the set $\Theta_{\boldsymbol{t}} = \{\boldsymbol{t}\} \cup \{\boldsymbol{t}_R \mid R \in \Omega_0\}$ is precisely $\mathcal{T}_1$-realisable. But then at least one of these $\Theta_{\boldsymbol{t}}$, for $\boldsymbol{t} \in \Xi$, is as required, for otherwise, if all of them turn out to be precisely $\mathcal{T}_2$-realisable, the disjoint union of models $\mathcal{I}_{\boldsymbol{t}}$ for $\mathcal{T}_2$ precisely realising $\Theta_{\boldsymbol{t}}$ would precisely realise the whole Ξ, which is impossible. ❑

Theorem D.5 *The Σ-query, strong Σ-concept and strong Σ-query entailment problems for DL-Lite$_{bool}$ are all Π_2^p-complete.*

Proof. We check criterion **(pr)** of Lemma B.2. Let Σ be a signature and Ω the set of role names and their inverses that occur in $\mathcal{T}_1 \cup \mathcal{T}_2$. We may assume that $\Sigma \subseteq sig(\mathcal{T}_1 \cup \mathcal{T}_2)$. By Lemma D.3, for both $\mathcal{T} = \mathcal{T}_1$ and $\mathcal{T} = \mathcal{T}_2$, it is decidable in NP (in $|\mathcal{T}_1 \cup \mathcal{T}_2|$) whether a set Ξ of $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types of size $\leq |\Omega| + 1$ is precisely $\mathcal{T}$-realisable. The Σ_2^p algorithm deciding whether there exists a set of $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types that is precisely $\mathcal{T}_1$-realisable but not precisely $\mathcal{T}_2$-realisable is as follows:

1. Guess a set Ξ of $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types of size $\leq |\Omega| + 1$.
2. Check, using an NP-oracle, whether (i) Ξ is precisely $\mathcal{T}_1$-realisable, and whether (ii) Ξ is not precisely $\mathcal{T}_2$-realisable.
3. Return '$\mathcal{T}_1$ Σ-query entails $\mathcal{T}_2$' if the answers to (i) and (ii) are both positive.

By Lemmas B.2 and D.4, $\mathcal{T}_1$ Σ-query entails $\mathcal{T}_2$ if, and only if, the algorithm says so. ❑

E Decision procedures and complexity for *DL-Lite$_{horn}$*

The following is proved in (Artale *et al.* 2007):

Theorem E.1 *The problem '$\mathcal{T} \models C_1 \sqsubseteq C_2$?' in DL-Lite$_{horn}$ is* P*-complete.*

Theorem E.2 *Σ-concept entailment for DL-Lite$_{horn}$ TBoxes is* coNP*-complete.*

Proof. Observe that, by Lemmas B.4 and B.1, if $\mathcal{T}_1$ does not Σ-entail $\mathcal{T}_2$ in *DL-Lite$_{horn}$* w.r.t. Σ, then there exists a $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-concept inclusion in *DL-Lite$_{horn}$* witnessing this. For the NP-upper bound for non-entailment, observe that such a witness concept inclusion is of polynomial size (in $|\mathcal{T}_1 \cup \mathcal{T}_2|$). Hence the algorithm non-deterministically guesses such a witness and then checks in polynomial time whether it is a consequence of $\mathcal{T}_2$ but not a consequence of $\mathcal{T}_1$. The coNP lower bound follows from the fact that Σ-entailment is already coNP-hard for the Horn fragment of propositional logic; see, e.g., (Flögel, Kleine Büning and Lettmann 2005). ❑

The proof of NP-completeness of Σ-query entailment for *DL-Lite$_{horn}$* TBoxes is based on criterion **(spr)** of Lemma B.6. Note that in precisely the same way as in the proof of Lemma D.4 one can show now that if **(spr)** does not hold for some Ξ, then it does not hold for a Ξ with $|\Xi| \leq |\Omega| + 1$, where Ω is the set of all role names and their inverses that occur in $\mathcal{T}_1 \cup \mathcal{T}_2$.

In the formulations of the algorithms below we will be taking the *local closures* of types under the TBox rules. More precisely, given a type $\boldsymbol{t}$ and a TBox $\mathcal{T}$ in *DL-Lite$_{horn}$*, we denote by $\mathsf{cl}^\bullet_{\mathcal{T}}(\boldsymbol{t})$ the type where $\mathsf{cl}^\bullet_{\mathcal{T}}(\boldsymbol{t})^+$ is the result of applying iteratively the 'rules' from $\mathcal{T}$ to $\boldsymbol{t}^+$ in the Datalog manner: if $\bigsqcap B_i \sqsubseteq B \in \mathcal{T}$ and $B \notin \boldsymbol{t}^+$ then add B to $\boldsymbol{t}^+$. Note that the only difference between $\mathsf{cl}^\bullet_{\mathcal{T}}(\boldsymbol{t})$ and the 'global closure' $\mathsf{cl}_{\mathcal{T}}(\boldsymbol{t})$ is that the latter can contain $\bot$ even when $\bot \notin \mathsf{cl}^\bullet_{\mathcal{T}}(\boldsymbol{t})$. Indeed, consider the TBox

$$\mathcal{T} = \{A \sqsubseteq \exists R,\;\; A \sqcap \exists R^- \sqsubseteq \bot,\;\; \top \sqsubseteq A\}$$

and the type $\boldsymbol{t} = \{\top\}$. Then $\mathsf{cl}^\bullet_{\mathcal{T}}(\boldsymbol{t}) = \{\top, A, \exists R, \neg\exists R^-\}$, while $\mathsf{cl}_{\mathcal{T}}(\boldsymbol{t}) = \{\top, A, \exists R, \exists R^-, \bot\}$ because $\mathcal{T} \models \top \sqsubseteq \bot$; see Example 4.

Theorem E.3 *Σ-query entailment for DL-Lite$_{horn}$ TBoxes is* coNP*-complete.*

Proof. We present a nondeterministic polynomial-time algorithm for deciding whether $\mathcal{T}_1$ does *not* Σ-query entails $\mathcal{T}_2$ in *DL-Lite$_{horn}$*, where without loss of generality we may assume that $\Sigma \subseteq sig(\mathcal{T}_1) \cap sig(\mathcal{T}_2)$. The algorithm is formulated in a straightforward manner without a reduction to propositional logic. Let Ω be the set of all role names and their inverses that occur in $\mathcal{T}_1 \cup \mathcal{T}_2$.

1. Guess a set Ξ of $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types of size $\leq |\Omega| + 1$.
2. For each $\boldsymbol{t} \in \Xi$, we compute (in time polynomial in $|\mathcal{T}_1|$) its local closure $\mathsf{cl}^\bullet_{\mathcal{T}_1}(\boldsymbol{t})$ and denote the set of all such closures by $\mathsf{cl}^\bullet_{\mathcal{T}_1}(\Xi)$ (these are all $sig(\mathcal{T}_1) Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types). Now, Ξ is precisely $\mathcal{T}_1$-realisable iff the following conditions hold:

– $\boldsymbol{t} = \mathsf{cl}^{\bullet}_{\mathcal{T}_1}(\boldsymbol{t}) \restriction \Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$ and $\bot \notin \boldsymbol{t}$, for all $\boldsymbol{t} \in \Xi$;
– for every $\boldsymbol{t} \in \mathsf{cl}^{\bullet}_{\mathcal{T}_1}(\Xi)$ with $(\geq 1\, R) \in \boldsymbol{t}$, there exists $\boldsymbol{t}' \in \mathsf{cl}^{\bullet}_{\mathcal{T}_1}(\Xi)$ with $(\geq 1\, \mathit{inv}(R)) \in \boldsymbol{t}'$.

3. If Ξ is precisely $\mathcal{T}_1$-realisable, then we do the following. First, compute $\Theta_0 = \mathsf{cl}^{\bullet}_{\mathcal{T}_2}(\Xi)$ and check whether

   – $\boldsymbol{t} = \mathsf{cl}^{\bullet}_{\mathcal{T}_2}(\boldsymbol{t}) \restriction \Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$ and $\bot \notin \boldsymbol{t}$, for all $\boldsymbol{t} \in \Xi$.

   If this is not the case, stop with answer 'No.' Now, if Θ_i, $i \geq 0$, has already been computed and there is $\boldsymbol{t} \in \Theta_i$ with $(\geq 1\, R) \in \boldsymbol{t}$, for some role R, but there is no $\boldsymbol{t}' \in \Theta_i$ with $(\geq 1\, \mathit{inv}(R)) \in \boldsymbol{t}'$, then we construct the type $\boldsymbol{t}' = \mathsf{cl}^{\bullet}_{\mathcal{T}_2}(\{\geq 1\, \mathit{inv}(R)\})$, check whether the following holds

   – $\bot \notin \boldsymbol{t}'$ and there is $\boldsymbol{t} \in \Xi$ with $\boldsymbol{t}'^{+} \restriction \Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2} \subseteq \boldsymbol{t}^{+}$,

   and if it does, we add $\boldsymbol{t}'$ to Θ_i and denote the result by Θ_{i+1}; otherwise we terminate with answer 'No.' We stop when $\Theta_n = \Theta_{n+1}$. Clearly, all this can be done in polynomial time.

$\mathcal{T}_1$ does not Σ-query entails $\mathcal{T}_2$ in *DL-Lite$_{horn}$* iff there is a set Ξ guessed at step 1 such that the conditions at step 2 are satisfied, while step 3 terminates with answer 'No.' ❑

Finally, we formulate a coNP algorithm for deciding strong Σ-concept and Σ-query entailment *DL-Lite$_{horn}$* TBoxes, using criterion **(mpr)** of Lemma B.11. Note first that we again have the following:

Lemma E.4 *Suppose that a set Ξ of $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types is precisely $\mathcal{T}_1$-realisable but does not satisfy condition* **(mpr)** *from Lemma B.11. Let Ω be the set of role names and their inverses that occur in $\mathcal{T}_1 \cup \mathcal{T}_2$. Then there is some $\Theta \subseteq \Xi$ with $|\Theta| \leq |\Omega| + 1$ such that Θ is precisely $\mathcal{T}_1$-realisable but not meet-precisely $\mathcal{T}_2$-realisable.*

Proof. As in the proof of Lemma D.4, for every $\boldsymbol{t} \in \Xi$, we take $\Omega_0 \subseteq \Omega$ such that the set $\Theta_{\boldsymbol{t}} = \{\boldsymbol{t}\} \cup \{\boldsymbol{t}_R \mid R \in \Omega_0\}$ is precisely $\mathcal{T}_1$-realisable. We again claim that at least one of these $\Theta_{\boldsymbol{t}}$, for $\boldsymbol{t} \in \Xi$, is not meet-precisely $\mathcal{T}_2$-realisable. Indeed, suppose, on the contrary, that **(mpr)** holds for all the $\Theta_{\boldsymbol{t}}$. Then we have models $\mathcal{I}_{\boldsymbol{t}} \models \mathcal{T}_2$ such that $\mathcal{I}_{\boldsymbol{t}}$ realises all types in $\Theta_{\boldsymbol{t}}$ and each $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-type realised in $\mathcal{I}_{\boldsymbol{t}}$ is $\mathcal{T}_{\Theta_{\boldsymbol{t}}}$-realisable and h-contained in a type from $\Theta_{\boldsymbol{t}}$. Let $\mathcal{J}$ be the disjoint union of all these $\mathcal{I}_{\boldsymbol{t}}$. Clearly, $\mathcal{J} \models \mathcal{T}_2$, $\mathcal{J}$ realises all types in Ξ and each $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-type realised in $\mathcal{J}$ is h-contained in a type from Ξ. And since each $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-type realised in $\mathcal{J}$ is $\mathcal{T}_{\Theta_{\boldsymbol{t}}}$-realisable, it must be also $\mathcal{T}_{\Xi}$-realisable, as $\Theta_{\boldsymbol{t}} \subseteq \Xi$, and so $\mathcal{T}_{\Theta_{\boldsymbol{t}}} \supseteq \mathcal{T}_{\Xi}$. But then Ξ satisfies **(mpr)**, which is a contradiction. ❑

In the proof of the next theorem we will be taking the *local closures* of types under a TBox $\mathcal{T}$ in *DL-Lite$_{horn}$* and the TBox $\mathcal{T}_{\Xi}$ induced by some set Ξ of types. More precisely, given a type $\boldsymbol{t}$, a TBox $\mathcal{T}$ in *DL-Lite$_{horn}$* and a set Ξ of types, we denote by $\mathsf{cl}^{\bullet}_{\mathcal{T} \cup \mathcal{T}_{\Xi}}(\boldsymbol{t})$ the type where $\mathsf{cl}^{\bullet}_{\mathcal{T}}(\boldsymbol{t})^{+}$ is the result of the following iterative procedure:

1. if $\bigsqcap B_i \sqsubseteq B$ is in $\mathcal{T}$ and all conjuncts of $\bigsqcap B_i$ are in $\boldsymbol{t}^{+}$, then add B to $\boldsymbol{t}^{+}$;
2. if $\Lambda_{\boldsymbol{t}} = \{\boldsymbol{t}' \in \Xi \mid \boldsymbol{t}^{+} \subseteq \boldsymbol{t}'^{+}\}$ is empty then add $\bot$ to $\boldsymbol{t}^{+}$ and stop; otherwise add the concepts from $\bigcap_{\boldsymbol{t}' \in \Lambda_{\boldsymbol{t}}} \boldsymbol{t}'^{+}$ to $\boldsymbol{t}^{+}$;
3. stop if $\boldsymbol{t}^{+}$ has not been incremented; otherwise go to step 1.

Theorem E.5 *The strong Σ-concept and Σ-query entailment problems for TBoxes in DL-Lite$_{horn}$ are* coNP*-complete.*

Proof. We show a nondeterministic polynomial-time algorithm for deciding whether $\mathcal{T}_1$ does *not* strongly Σ-concept (or Σ-query) entail $\mathcal{T}_2$ in *DL-Lite$_{horn}$*, where without loss of generality we assume that $\Sigma \subseteq \mathit{sig}(\mathcal{T}_1) \cap \mathit{sig}(\mathcal{T}_2)$. Let Ω be the set of all role names and their inverses that occur in $\mathcal{T}_1 \cup \mathcal{T}_2$.

1. Guess a set Ξ of $\Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types of size $\leq |\Omega| + 1$.
2. For each $\boldsymbol{t} \in \Xi$, we compute (in time polynomial in $|\mathcal{T}_1|$) its $\mathcal{T}_1$-closure $\mathsf{cl}^{\bullet}_{\mathcal{T}_1}(\boldsymbol{t})$ and denote the set of all such closures by $\mathsf{cl}^{\bullet}_{\mathcal{T}_1}(\Xi)$ (these are all $\mathit{sig}(\mathcal{T}_1) Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$-types). Now, Ξ is precisely $\mathcal{T}_1$-realisable iff the following conditions hold:

   – $\boldsymbol{t} = \mathsf{cl}^{\bullet}_{\mathcal{T}_1}(\boldsymbol{t}) \restriction \Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$ and $\bot \notin \boldsymbol{t}$, for all $\boldsymbol{t} \in \Xi$;
   – for every $\boldsymbol{t} \in \mathsf{cl}^{\bullet}_{\mathcal{T}_1}(\Xi)$ with $(\geq 1\, R) \in \boldsymbol{t}$, there exists $\boldsymbol{t}' \in \mathsf{cl}^{\bullet}_{\mathcal{T}_1}(\Xi)$ with $(\geq 1\, \mathit{inv}(R)) \in \boldsymbol{t}'$.

3. If Ξ is precisely $\mathcal{T}_1$-realisable, then we do the following. First, compute $\Theta_0 = \mathsf{cl}^{\bullet}_{\mathcal{T}_2}(\Xi)$ and check whether

   – $\boldsymbol{t} = \mathsf{cl}^{\bullet}_{\mathcal{T}_2}(\boldsymbol{t}) \restriction \Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2}$ and $\bot \notin \boldsymbol{t}$, for all $\boldsymbol{t} \in \Xi$.

   If this is not the case, stop with answer 'No.' Now, if Θ_i, $i \geq 0$, has already been computed and there is $\boldsymbol{t} \in \Theta_i$ with $(\geq 1\, R) \in \boldsymbol{t}$, for some role R, but there is no $\boldsymbol{t}' \in \Theta_i$ with $(\geq 1\, \mathit{inv}(R)) \in \boldsymbol{t}'$, then we construct the type $\boldsymbol{t}' = \mathsf{cl}^{\bullet}_{\mathcal{T}_2 \cup \mathcal{T}_{\Xi}}(\{\geq 1\, \mathit{inv}(R)\})$ according to the procedure above, check whether the following holds

   – $\bot \notin \boldsymbol{t}'$ and there is $\boldsymbol{t} \in \Xi$ with $\boldsymbol{t}'^{+} \restriction \Sigma Q_{\mathcal{T}_1 \cup \mathcal{T}_2} \subseteq \boldsymbol{t}^{+}$,

   and if it does, we add $\boldsymbol{t}'$ to Θ_i and denote the result by Θ_{i+1}; otherwise we terminate with answer 'No.' We stop when $\Theta_n = \Theta_{n+1}$. Clearly, all this can be done in polynomial time.

$\mathcal{T}_1$ does not strongly Σ-concept (or Σ-query) entail $\mathcal{T}_2$ in *DL-Lite$_{horn}$* iff there is a set Ξ guessed at step 1 such that the conditions at step 2 are satisfied, while step 3 terminates with answer 'No.' ❑

F QBF for *DL-Lite$_{bool}$*

In what follows, for a TBox $\mathcal{T}$, we denote by $m_{\mathcal{T}}$ the number of role names in $\mathcal{T}$.

Let Q be a set of numerical parameters containing 1 and all parameters from $\mathcal{T}_1$ and $\mathcal{T}_2$. Denote by Σ_1 and Σ_2 the signatures of $\mathcal{T}_1$ and $\mathcal{T}_2$, respectively. We also assume that $\mathit{sig}(\mathcal{T}_1) \cap \mathit{sig}(\mathcal{T}_2) \subseteq \Sigma$.

Σ-concept entailment

First we refine the criterion of Theorem 11 with the aim of encoding it by means of QBFs. $\mathcal{T}$-realisability of a type $\boldsymbol{t}$ means that there is a precisely $\mathcal{T}$-realisable set Ξ of types at least one of which expands $\boldsymbol{t}$. And it turns out that one can always find such a Ξ of size $\leq m_{\mathcal{T}} + 1$. Moreover, we can order the types in Ξ in such a way that its i's type $\boldsymbol{t}_i$ 'takes care of the role P_i.' To make this claim more precise we need a definition. For a ΣQ-type $\boldsymbol{t}$, a sequence $\Theta^{\mathcal{T}}_{\boldsymbol{t}} = \boldsymbol{t}_0, \boldsymbol{t}_1, \dots, \boldsymbol{t}_{m_\mathcal{T}}$ of (not necessarily distinct) $sig(\mathcal{T})Q$-types is called a *$\mathcal{T}$-witness set for $\boldsymbol{t}$* if

(a₁) $\boldsymbol{t} \subseteq \boldsymbol{t}_0$;

(b₁) each type in $\boldsymbol{t}_0, \boldsymbol{t}_1, \dots, \boldsymbol{t}_{m_\mathcal{T}}$ is $\mathcal{T}$-realisable;

(c₁) $\exists P_i \in \boldsymbol{t}_j$, for some j, iff $\exists P_i \in \boldsymbol{t}_i$ or $\exists P_i^- \in \boldsymbol{t}_i$, for each of the role names P_i in $\mathcal{T}$, $1 \leq i \leq m_\mathcal{T}$.

Theorem F.1 *A ΣQ-type $\boldsymbol{t}$ is $\mathcal{T}$-realisable iff there is a $\mathcal{T}$-witness set for $\boldsymbol{t}$. So $\mathcal{T}_1$ Σ-concept entails $\mathcal{T}_2$ in DL-Lite$_{bool}$ iff, for every ΣQ-type $\boldsymbol{t}$, whenever there is a $\mathcal{T}_1$-witness set for $\boldsymbol{t}$ then there is also a $\mathcal{T}_2$-witness set for $\boldsymbol{t}$.*

To translate the criterion of Theorem F.1 into QBF, with each basic $\Sigma_0 Q$-concept (different from $\bot$) we associate a propositional variable. Fix some linear order on the set of all basic concepts, and let $B_1, \dots, B_n$ be the induced list of $\Sigma_0 Q$-concepts. Then any vector $\boldsymbol{t} = (b_1, \dots, b_n)$ of distinct propositional variables b_i can be used to encode $\Sigma_0 Q$-types: every classical assignment $\mathfrak{a}$ (of the truth values F and T to propositional variables) gives rise to the $\Sigma_0 Q$-type $\boldsymbol{t}^{\mathfrak{a}}(\boldsymbol{t})$ such that $B_i \in \boldsymbol{t}^{\mathfrak{a}}(\boldsymbol{t})$ iff $\mathfrak{a}(b_i) = \mathsf{T}$ (and so if $\mathfrak{a}(b_i) = \mathsf{F}$ then $\neg B \in \boldsymbol{t}^{\mathfrak{a}}(\boldsymbol{t})$). We will call $\boldsymbol{t}$ a *$\Sigma_0 Q$-vector* and $\boldsymbol{t}^{\mathfrak{a}}(\boldsymbol{t})$ the *$\Sigma_0 Q$-type of $\boldsymbol{t}$ under* $\mathfrak{a}$. We also set $\boldsymbol{t}(B_i) = b_i$ and extend this map inductively to complex $\Sigma_0 Q$-concepts:

$$\boldsymbol{t}(\bot) = \bot, \qquad \boldsymbol{t}(\neg C) = \neg \boldsymbol{t}(C),$$
$$\boldsymbol{t}(C_1 \sqcap C_2) = \boldsymbol{t}(C_1) \wedge \boldsymbol{t}(C_2).$$

We use concatenation $\boldsymbol{t}_0 \cdot \boldsymbol{t}_1$ of types $\boldsymbol{t}_0$, $\boldsymbol{t}_1$ (when extending $\Sigma_0 Q$-types to $\Sigma_0' Q$-types, $\Sigma_0 \subset \Sigma_0'$) and projection $\boldsymbol{t}\restriction_{\{B_1,\dots,B_k\}} = (\boldsymbol{t}(B_1), \dots, \boldsymbol{t}(B_k))$ (not a $\Sigma_0 Q$-vector, in general). A sequence $\boldsymbol{t}^n, \dots, \boldsymbol{t}^m$ of ΣQ_0-vectors is denoted by $\boldsymbol{t}^{n..m}$.

Let $\boldsymbol{t}_0^0$ be a ΣQ-vector, $\hat{\boldsymbol{t}}_1^0$ a $(\Sigma_1 \setminus \Sigma)Q$-vector, $\boldsymbol{t}_1^{1..m_{\mathcal{T}_1}}$ a sequence of $\Sigma_1 Q$-vectors, $\hat{\boldsymbol{t}}_2^0$ a $(\Sigma_2 \setminus \Sigma)Q$-vector, and $\boldsymbol{t}_2^{1..m_{\mathcal{T}_2}}$ a sequence of $\Sigma_2 Q$-vectors. By Theorem F.1, the condition '$\mathcal{T}_1$ *Σ-concept entails* $\mathcal{T}_2$' can be represented by means of the following closed quantified Boolean formula

$$\forall \boldsymbol{t}_0^0 \Big[\exists \hat{\boldsymbol{t}}_1^0 \boldsymbol{t}_1^{1..m_{\mathcal{T}_1}} \phi_{\mathcal{T}_1}(\boldsymbol{t}_0^0 \cdot \hat{\boldsymbol{t}}_1^0, \boldsymbol{t}_1^{1..m_{\mathcal{T}_1}}) \ \to \ \exists \hat{\boldsymbol{t}}_2^0 \boldsymbol{t}_2^{1..m_{\mathcal{T}_2}} \phi_{\mathcal{T}_2}(\boldsymbol{t}_0^0 \cdot \hat{\boldsymbol{t}}_2^0, \boldsymbol{t}_2^{1..m_{\mathcal{T}_2}}) \Big], \quad (3)$$

where, for a TBox $\mathcal{T}$ and $N \geq m_\mathcal{T}$,

$$\phi_\mathcal{T}(\boldsymbol{t}^{0..N}) \ = \ \bigwedge_{j=0}^{N} \theta_\mathcal{T}(\boldsymbol{t}^j) \quad \wedge \quad \bigwedge_{i=1}^{m_\mathcal{T}} \varrho_{P_i,i}(\boldsymbol{t}^{0..N}\restriction_{\{\exists P_i, \exists P_i^-\}}),$$

$$\theta_\mathcal{T}(\boldsymbol{t}) \ = \ \bigwedge_{D_1 \sqsubseteq D_2 \in \mathcal{T}} \big(\boldsymbol{t}(D_1) \to \boldsymbol{t}(D_2)\big),$$

$$\begin{aligned}\varrho_{P,i}(\boldsymbol{p}^{0..N}) \ &= \big(\boldsymbol{p}^i(\exists P) \to \bigvee\nolimits_{j=0}^{N} \boldsymbol{p}^j(\exists P^-)\big) \\ &\wedge \big(\boldsymbol{p}^i(\exists P^-) \to \bigvee\nolimits_{j=0}^{N} \boldsymbol{p}^j(\exists P)\big) \\ &\wedge \big(\neg \boldsymbol{p}^i(\exists P) \wedge \neg \boldsymbol{p}^i(\exists P^-) \to \\ &\qquad \bigwedge\nolimits_{\substack{j=0 \\ j \neq i}}^{N} \neg \boldsymbol{p}^j(\exists P) \wedge \bigwedge\nolimits_{\substack{j=0 \\ j \neq i}}^{N} \neg \boldsymbol{p}^j(\exists P^-)\big).\end{aligned}$$

Theorem F.2 *For each assignment $\mathfrak{a}$, we have $\mathfrak{a}(\phi_\mathcal{T}(\boldsymbol{t}^{0..N})) = \mathsf{T}$ iff the set $\{\boldsymbol{t}^{\mathfrak{a}}(\boldsymbol{t}^0), \dots, \boldsymbol{t}^{\mathfrak{a}}(\boldsymbol{t}^N)\}$ of $sig(\mathcal{T})Q$-types is precisely $\mathcal{T}$-realisable in a model $\mathcal{I}$ where $P_i^\mathcal{I} \neq \emptyset$ iff $\mathfrak{a}(\boldsymbol{t}^i(\exists P_i)) = \mathsf{T}$ or $\mathfrak{a}(\boldsymbol{t}^i(\exists P_i^-)) = \mathsf{T}$, for $1 \leq i \leq m_\mathcal{T}$. In particular, $\mathcal{T}_1$ Σ-concept entails $\mathcal{T}_2$ w.r.t. Σ iff QBF* (3) *is satisfiable.*

There are different ways of transforming (3) into a prenex CNF, which is a standard input to QBF solvers (see `http://dcs.bbk.ac.uk/~roman/qbf` for some options). One of the versions we used in our experiments is of the form

$$\begin{aligned}&\forall \boldsymbol{t}_0^0 \forall \hat{\boldsymbol{t}}_1^0 \boldsymbol{t}_1^{1..m_{\mathcal{T}_1}} \ \exists \hat{\boldsymbol{t}}_2^0 \boldsymbol{t}_2^{1..m_{\mathcal{T}_2}} \\ &\quad \exists \boldsymbol{u}_1 \dots \boldsymbol{u}_{m_1} \exists \boldsymbol{w}^{0..m_{\mathcal{T}_1}} \exists p \\ &\quad \Big[\phi'_{\mathcal{T}_1}(\boldsymbol{t}_0^0 \cdot \hat{\boldsymbol{t}}_1^0, \boldsymbol{t}_1^{1..m_{\mathcal{T}_1}}, \boldsymbol{u}_1 \dots \boldsymbol{u}_{m_{\mathcal{T}_1}}, \boldsymbol{w}^{0..m_{\mathcal{T}_1}}, p) \\ &\qquad\qquad \wedge \ \phi''_{\mathcal{T}_2}(\boldsymbol{t}_0^0 \cdot \hat{\boldsymbol{t}}_2^0, \boldsymbol{t}_2^{1..m_{\mathcal{T}_2}}, p)\Big],\end{aligned}$$

where $\boldsymbol{u}_1, \dots, \boldsymbol{u}_{m_1}$, $\boldsymbol{w}^{0..m_{\mathcal{T}_1}}$ and p are K auxiliary variables, $K = (m_{\mathcal{T}_1}+1)C_{\mathcal{T}_1} + 3m_{\mathcal{T}_1} + 1$ and $C_\mathcal{T}$ is the number of axioms in $\mathcal{T}$. In total the prenex QBF has $(m_{\mathcal{T}_1}+1)W_{\mathcal{T}_1}$ universal and $(m_{\mathcal{T}_2}+1)W_{\mathcal{T}_2} - W_0 + K$ existential variables, where $W_\mathcal{T}$ and W_0 are the numbers of basic concepts in $\mathcal{T}$ and Σ, respectively. CNFs $\phi'_\mathcal{T}(\boldsymbol{t}^{0..N}, \boldsymbol{u}_1 \dots \boldsymbol{u}_{m_\mathcal{T}}, \boldsymbol{w}^{0..N}, p)$ and $\phi''_\mathcal{T}(\boldsymbol{t}^{0..N}, p)$ contain $(N+1)B_\mathcal{T} + 1 + (2N+7)m_\mathcal{T}$ and $(N+1)(C_\mathcal{T} + B'_\mathcal{T}) + 2(N+1)m_\mathcal{T}$ clauses, where $B_\mathcal{T}$ and $B'_\mathcal{T}$ are the numbers of basic concepts in the left- and right-hand sides in $\mathcal{T}$, respectively.

The order of the variables in the prefix has a strong impact on the solvers' performance (as is well-known in the QBF community), and usually one can fine-tune it depending on the solver. Another important parameter, which has not been studied comprehensively yet by the QBF community, is the structure of the prefix. For example, some of the existential quantifiers can be moved right after the universal ones they depend on, which gives a prefix of the form $\forall\exists \dots \forall\exists$. The impact of this transformation is not completely clear. However, our experiments show—especially for the more complex Σ-query entailment—that the structure of the prefix may become crucial for a solver to succeed.

Σ-query entailment

To make the criterion of Theorem 11 for Σ-query entailment in *DL-Lite$_{bool}$* more efficient, we observe first that a set Ξ of $sig(\mathcal{T})Q$-types is precisely $\mathcal{T}$-realisable iff every type in Ξ has a $\mathcal{T}$-witness set within Ξ. So the following conditions are equivalent:

- $\mathcal{T}_1$ Σ-query entails $\mathcal{T}_2$;
- for every $\mathcal{T}_1$-witness set $\Theta^{\mathcal{T}_1}_{\boldsymbol{t}}$ for a ΣQ-type $\boldsymbol{t}$, the set $\Theta^{\mathcal{T}_1}_{\boldsymbol{t}} \restriction \Sigma$ is precisely $\mathcal{T}_2$-realisable, where $\Theta^{\mathcal{T}_1}_{\boldsymbol{t}} \restriction \Sigma$ is the set of restrictions of types in $\Theta^{\mathcal{T}_1}_{\boldsymbol{t}}$ to Σ.

Intuitively, this result means that we do not have to consider arbitrary sets of $\Sigma_1 Q$-types, but only those of size $\leq m_{\mathcal{T}_1}+1$ that are 'generated' by a ΣQ-type $\boldsymbol{t}$ and ordered in such a way that a certain type $\boldsymbol{t}_i$ in the ordering 'takes care of P_i.' Now we extend the notion of a $\mathcal{T}$-witness set as follows. For a $\mathcal{T}_1$-witness set $\Theta_{\boldsymbol{t}}^{\mathcal{T}_1} = \boldsymbol{t}_0, \boldsymbol{t}_1, \dots, \boldsymbol{t}_{m_{\mathcal{T}_1}}$ and $M = m_{\mathcal{T}_2} - m_0$, call a sequence $\Theta_{\boldsymbol{t}}^{\mathcal{T}_1\mathcal{T}_2} = \hat{\boldsymbol{t}}_0, \hat{\boldsymbol{t}}_1, \dots, \hat{\boldsymbol{t}}_{m_{\mathcal{T}_1}}, \mathfrak{s}_1, \dots, \mathfrak{s}_M$ of $\Sigma_2 Q$-types a *$\mathcal{T}_2$-witness set for* $\Theta_{\boldsymbol{t}}^{\mathcal{T}_1}$ if

(a$_2$) for each $1 \le i \le m_{\mathcal{T}_1}$, $\boldsymbol{t}_i \restriction \Sigma \subseteq \hat{\boldsymbol{t}}_i$,

(a$'_2$) for each $1 \le j \le M$, there is $1 \le k \le m_{\mathcal{T}_1}$ with $\boldsymbol{t}_k \restriction \Sigma \subseteq \mathfrak{s}_j$,

(b$_2$) each type in $\hat{\boldsymbol{t}}_0, \hat{\boldsymbol{t}}_1, \dots, \hat{\boldsymbol{t}}_{m_{\mathcal{T}_1}}, \mathfrak{s}_1, \dots, \mathfrak{s}_M$ is $\mathcal{T}_2$-realisable,

(c$_2$) $\exists P_i \in \hat{\boldsymbol{t}}_j$, for some $1 \le j \le m_{\mathcal{T}_1}$, or $\exists P_i \in \mathfrak{s}_k$, for some $1 \le k \le M$, iff $\exists P_i \in \mathfrak{s}_i$ or $\exists P_i^- \in \mathfrak{s}_i$, for each role name P_i in $\Sigma_2 \setminus \Sigma$, $1 \le i \le M$.

Theorem F.3 *$\mathcal{T}_1$ Σ-query entails $\mathcal{T}_2$ in DL-Lite$_{bool}$ iff, for every $\mathcal{T}_1$-witness set $\Theta_{\boldsymbol{t}}^{\mathcal{T}_1}$ for some ΣQ-type $\boldsymbol{t}$, there is a $\mathcal{T}_2$-witness set for $\Theta_{\boldsymbol{t}}^{\mathcal{T}_1}$.*

In the criterion of Theorem F.1, we had to take a ΣQ-type $\boldsymbol{t}$, (i) extend $\boldsymbol{t}$ to a $\Sigma_1 Q$-type, (ii) check whether there are 'witnesses' for all the roles in that type and the types providing those witnesses, and if this is the case, we finally had to repeat steps (i) and (ii) again for Σ_2 in place of Σ_1. The criterion of Theorem F.3 is much more complex not only because now we have to start with a set of $(m_{\mathcal{T}_1} + 1)$ $\Sigma_1 Q$-types rather than a single type. More importantly, now the $\mathcal{T}_2$-witnesses we choose for these types are not *arbitrary* but must have the same Σ-restrictions as the original $\Sigma_1 Q$-types. This last condition makes the QBF translation much more complex (see below) and, consequently, computationally more costly.

Let $M = m_{\mathcal{T}_2} - m_0$, $\boldsymbol{t}_0^{0..m_{\mathcal{T}_1}}$ be ΣQ-vectors, $\hat{\boldsymbol{t}}_1^{0..m_{\mathcal{T}_1}}$ $(\Sigma_1 \setminus \Sigma)Q$-vectors, $\hat{\boldsymbol{t}}_2^{0..m_{\mathcal{T}_1}}$, $\boldsymbol{s}_2^{1..M}$ be $(\Sigma_2 \setminus \Sigma)Q$-vectors. By Theorem F.3, the condition '$\mathcal{T}_2$ Σ-*query entails* $\mathcal{T}_2$' can be expressed by the following closed QBF

$$\forall \boldsymbol{t}_0^{0..m_{\mathcal{T}_1}} \Big[\exists \hat{\boldsymbol{t}}_1^{0..m_{\mathcal{T}_1}} \phi_{\mathcal{T}_1}((\boldsymbol{t}_0 \cdot \hat{\boldsymbol{t}}_1)^{0..m_{\mathcal{T}_1}}) \;\; \rightarrow \\ \exists \hat{\boldsymbol{t}}_2^{0..m_{\mathcal{T}_1}} \exists \boldsymbol{s}_2^{1..M} \;\; \beta_{\mathcal{T}_2}(\boldsymbol{t}_0^{0..m_{\mathcal{T}_1}}, \hat{\boldsymbol{t}}_2^{0..m_{\mathcal{T}_1}}, \boldsymbol{s}_2^{1..M}) \Big], \quad (4)$$

where $\beta_{\mathcal{T}_2}(\boldsymbol{t}_0^{0..m_{\mathcal{T}_1}}, \hat{\boldsymbol{t}}_2^{0..m_{\mathcal{T}_1}}, \boldsymbol{s}_2^{1..M})$ is the formula

$$\bigwedge_{j=0}^{m_{\mathcal{T}_1}} \theta_{\mathcal{T}_2}(\boldsymbol{t}_0^j \cdot \hat{\boldsymbol{t}}_2^j) \quad \wedge \quad \bigwedge_{j=1}^{M} \bigvee_{k=0}^{m_{\mathcal{T}_1}} \theta_{\mathcal{T}_2}(\boldsymbol{t}_0^k \cdot \boldsymbol{s}_2^j) \\ \wedge \quad \bigwedge_{i=1}^{M} \varrho_{P_{m_0+i},i}((\boldsymbol{s}_2^{1..M})\restriction_{\{\exists P_i, \exists P_i^-\}}, (\hat{\boldsymbol{t}}_2^{0..m_{\mathcal{T}_1}})\restriction_{\{\exists P_i, \exists P_i^-\}}),$$

and $\phi_{\mathcal{T}}$, $\theta_{\mathcal{T}}$ and $\varrho_{P,i}$ are defined as before (here we assume that all concepts $\exists R$ for Σ-roles precede those for $\Sigma_2 \setminus \Sigma$-roles).

Theorem F.4 *$\mathcal{T}_1$ Σ-query entails $\mathcal{T}_2$ iff* (4) *is satisfiable.*

It can be checked that (4) is equivalent to the prenex QBF

$$\forall \boldsymbol{t}_0^{0..m_{\mathcal{T}_1}} \exists \hat{\boldsymbol{t}}_2^{0..m_{\mathcal{T}_1}} \exists \boldsymbol{s}_2^{1..M} \exists \boldsymbol{q}^{0..m_{\mathcal{T}_1}} \exists p \\ \forall \hat{\boldsymbol{t}}_1^0 \exists \boldsymbol{w}^0 \cdots \forall \hat{\boldsymbol{t}}_1^{m_{\mathcal{T}_1}} \exists \boldsymbol{w}^{m_{\mathcal{T}_1}} \exists \boldsymbol{u}_1 \dots \boldsymbol{u}_{m_{\mathcal{T}_1}} \\ \Big[\phi'_{\mathcal{T}_1}((\boldsymbol{t}_0^0 \cdot \hat{\boldsymbol{t}}_1)^{0..m_{\mathcal{T}_1}}, \boldsymbol{u}_1 \dots \boldsymbol{u}_{m_{\mathcal{T}_1}}, \boldsymbol{w}^{0..m_{\mathcal{T}_1}}, p) \wedge \\ \beta''_{\mathcal{T}_2}(\boldsymbol{t}_0^{0..m_{\mathcal{T}_1}}, \hat{\boldsymbol{t}}_2^{0..m_{\mathcal{T}_1}}, \boldsymbol{s}_2^{1..M}, \boldsymbol{q}^{0..m_{\mathcal{T}_1}}, p) \Big],$$

where $\boldsymbol{q}^j = (q_1^j, \dots, q_M^j)$, for $0 \le j \le m_{\mathcal{T}_1}$, $\phi_{\mathcal{T}}$ is as before and $\beta''_{\mathcal{T}}$ is a CNF equivalent to $(p \rightarrow \beta_{\mathcal{T}})$. The latter CNF contains $(M+1)(N+1)(C_{\mathcal{T}} + B'_{\mathcal{T}}) + 2M(M+N+1) + M$ clauses, where $N = m_{\mathcal{T}_1}$, which is quadratic in $m_{\mathcal{T}_2}$, the number of roles in Σ_2 (unlike $\phi''_{\mathcal{T}_2}$, which is only linear in $m_{\mathcal{T}_2}$).

References

A. Flögel, H. Kleine Büning, and T. Lettmann. On the restricted equivalence of subclasses of propositional logic. *ITA*, 27(4):327–340, 1993.