Is Query Inseparability for $\mathcal{ALC}$ Ontologies Decidable?

E. Botoeva,[1] C. Lutz,[2] V. Ryzhikov,[1] F. Wolter[3] and M. Zakharyaschev[4]

[1]Univ. of Bozen-Bolzano, [2]Univ. of Bremen, [3]Univ. of Liverpool, [4]Birkbeck, Univ. of London

1 Introduction

While query answering with Description Logics (DLs) is now well-developed, this is much less the case for reasoning services that support ontology engineering and target query answering as an application. In ontology versioning, for example, one would like to know whether two versions of an ontology give the same answers to all queries formulated over a given vocabulary of interest, which means that they can be safely replaced in an application [5]. Similarly, if one wants to know whether a given ontology can be safely replaced by a smaller subset (a module), it is the answers to all queries that should be preserved [7]. In this context, the fundamental relationship between ontologies is thus not whether they are logically equivalent (have the same models), but whether they give the same answers to all relevant queries. The resulting entailment problem can be formalized in two ways, with different applications. First, given a class $\mathcal{Q}$ of queries, knowledge bases (KBs) $\mathcal{K}_1 = (\mathcal{T}_1, \mathcal{A}_1)$ and $\mathcal{K}_2 = (\mathcal{T}_2, \mathcal{A}_2)$, and a signature Σ of relevant concept and role names, we say that $\mathcal{K}_1$ *Σ-$\mathcal{Q}$-entails* $\mathcal{K}_2$ if the answers to any Σ-query in $\mathcal{Q}$ over $\mathcal{K}_2$ are contained in the answers to the query over $\mathcal{K}_1$. $\mathcal{K}_1$ and $\mathcal{K}_2$ are *Σ-$\mathcal{Q}$-inseparable* if they Σ-$\mathcal{Q}$-entail each other. This notion of entailment is appropriate if the data is known and does not change frequently. Applications include data-oriented KB versioning and KB module extraction, KB forgetting [9], and knowledge exchange [1].

If the data is not known or changes frequently, it is not KBs that should be compared, but TBoxes. Given a pair $\Theta = (\Sigma_1, \Sigma_2)$ specifying a relevant signature Σ_1 for ABoxes and Σ_2 for queries, we say that a TBox $\mathcal{T}_1$ *Θ-$\mathcal{Q}$-entails* a TBox $\mathcal{T}_2$ if, for every Σ_1-ABox $\mathcal{A}$, the KB $(\mathcal{T}_1, \mathcal{A})$ Σ_2-$\mathcal{Q}$-entails $(\mathcal{T}_2, \mathcal{A})$. $\mathcal{T}_1$ and $\mathcal{T}_2$ are *Θ-$\mathcal{Q}$-inseparable* if they Θ-$\mathcal{Q}$-entail each other. Applications include data-oriented TBox versioning, TBox modularization and TBox forgetting [7].

Most important choices for $\mathcal{Q}$ are conjunctive queries (CQs) and unions thereof (UCQs); we also consider the practically relevant classes of *rooted* CQs (rCQs) and UCQs (rUCQs), in which every variable is connected to an answer variable. So far, CQ-entailment has been studied for Horn DL KBs [3], $\mathcal{EL}$ TBoxes [8, 5], DL-Lite TBoxes [6], and also for OBDA specifications, that is, DL-Lite TBoxes with mappings [2]. No results are available for non-Horn DLs (neither in the KB nor in the TBox case) and for expressive Horn DLs in the TBox case. In particular, query entailment in non-Horn DLs had the reputation of being a technically challenging problem.

This paper makes a first breakthrough into understanding query entailment and inseparability in these cases, with the main results summarized in Figures 1 and 2 (those marked with ($\star$) are from [3]). Most unexpected is the *undecidability* of CQ- and rCQ-entailments between $\mathcal{ALC}$ KBs, even when the first KB is formulated in *Horn*-$\mathcal{ALC}$ (in fact, $\mathcal{EL}$) and without any signature restriction. This should be contrasted with the

Queries	$\mathcal{ALC}$	*Horn*-$\mathcal{ALC}$ to $\mathcal{ALC}$	$\mathcal{ALC}$ to *Horn*-$\mathcal{ALC}$	*Horn*-$\mathcal{ALC}$
CQ	undecidable	undecidable	?	=ExpTime$^{(\star)}$
UCQ	?	?	?	=ExpTime$^{(\star)}$
rCQ	undecidable	undecidable	$\leq$2ExpTime	=ExpTime$^{(\star)}$
rUCQ	$\leq$2ExpTime	$\leq$2ExpTime	$\leq$2ExpTime	=ExpTime$^{(\star)}$

Fig. 1. KB query entailment.

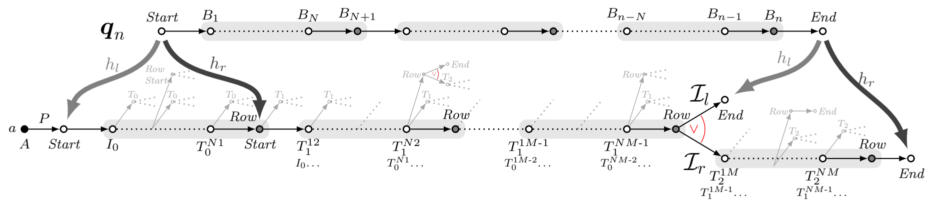
Queries	$\mathcal{ALC}$	*Horn*-$\mathcal{ALC}$ to $\mathcal{ALC}$	$\mathcal{ALC}$ to *Horn*-$\mathcal{ALC}$	*Horn*-$\mathcal{ALC}$
CQ	undecidable	undecidable	?	=2ExpTime
UCQ	?	?	?	=2ExpTime
rCQ	undecidable	undecidable	=ExpTime	=ExpTime
rUCQ	?	?	=ExpTime	=ExpTime

Fig. 2. TBox query entailment.

decidability of subsumption-based entailment between $\mathcal{ALC}$ TBoxes [4] and of CQ-entailment between *Horn*-$\mathcal{ALC}$ KBs [3]. The second surprising result is that entailment between $\mathcal{ALC}$ KBs becomes decidable when CQs are replaced with rUCQs. For $\mathcal{ALC}$ TBoxes, CQ- and rCQ-entailments are undecidable as well. We obtain decidability for *Horn*-$\mathcal{ALC}$ TBoxes (where CQ- und UCQ-entailments coincide) using the fact that non-entailment is always witnessed by tree-shaped ABoxes. As another surprise, CQ-entailment of *Horn*-$\mathcal{ALC}$ TBoxes is 2ExpTime-complete while rCQ-entailment is only ExpTime-complete. This should be contrasted with the $\mathcal{EL}$ case, where both problems are ExpTime-complete [8]. All upper bounds and most lower bounds hold also when entailment is replaced with inseparability.

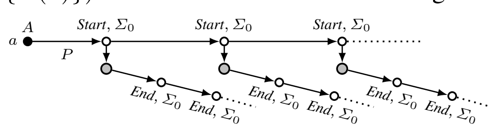
2 Undecidability of CQ-Entailment for $\mathcal{ALC}$ KBs

Here we sketch the construction showing that the problem whether an $\mathcal{EL}$ KB Σ-CQ entails an $\mathcal{ALC}$ KB is undecidable. The proof is by reduction of the undecidable $N \times M$-*tiling problem*: given a finite set $\mathfrak{T}$ of *tile types* T with four colours *up*(T), *down*(T), *left*(T) and *right*(T), a tile type $I \in \mathfrak{T}$, and two colours W (for wall) and C (for ceiling), decide whether there exist $N, M \in \mathbb{N}$ such that the $N \times M$ grid can be tiled using $\mathfrak{T}$ in such a way that $(1,1)$ is covered by a tile of type I; every (N,i), for $i \leq M$, is covered by a tile of type T with *right*$(T) = W$; and every (i,M), for $i \leq N$, is covered by a tile of type T with *up*$(T) = C$. Given an instance of this problem, we first describe a KB $\mathcal{K}_2 = (\mathcal{T}_2, \{A(a)\})$ that uses (among others) three concept names T_k, $k = 0,1,2$, for each tile type $T \in \mathfrak{T}$. If a point x in a model $\mathcal{I}$ of $\mathcal{K}_2$ is in T_k and *right*$(T) =$ *left*(T'), then x has an R-successor in T'_k. Thus, branches of $\mathcal{I}$ define (possibly infinite) horizontal rows of tilings with $\mathfrak{T}$. If a branch contains a point $y \in T_k$ with *right*$(T) = W$, then this y can be the last point in the row, which is indicated by an R-successor $z \in$ *Row* of y. In turn, z has R-successors in all $T_{(k+1) \bmod 3}$ that can be possible beginnings of the next row of tiles. To coordinate the *up* and *down* colours between the rows—which will be done by the CQs separating $\mathcal{K}_1$ and $\mathcal{K}_2$— we make every $x \in T_k$, starting from the second row, an instance of all $T'_{(k-1) \bmod 3}$ with *down*$(T) =$ *up*(T'). The row started by $z \in$ *Row* can be the last one in the tiling, in which case we require that each of its tiles T has *up*$(T) = C$. After the point in *Row* indicating the end of the final row, we add an R-successor in *End* for the end of tiling. The beginning of the first row is indicated by a P-successor in *Start* of the ABox element a, after which we add an R-successor in I_0 for the given initial tile type I; see the lowest branch in the figure below. To generate a tree with all possible branches described above, we only require $\mathcal{EL}$ axioms of the form $E \sqsubseteq D$ and $E \sqsubseteq \exists S.D$.



The existence of a tiling of some $N \times M$ grid for the given instance can be checked by Boolean CQs $\boldsymbol{q}_n$ that require an R-path from *Start* to *End* going through T_k- or *Row*-points: $\exists \boldsymbol{x}(\mathit{Start}(x_0) \wedge \bigwedge_{i=0}^{n} R(x_i, x_{i+1}) \wedge \bigwedge_{i=1}^{n} B_i(x_i) \wedge \mathit{End}(x_{n+1}))$, where $B_i \in \{\mathit{Row}\} \cup \{T_k \mid T \in \mathfrak{T}, k = 0, 1, 2\}$. The key trick is—using an axiom of the form $D \sqsubseteq E \sqcup E'$—to ensure that the *Row*-point before the final row of the tiling has *two alternative* continuations: one as described above, and the other one having just a single R-successor in *End* where $\vee$ indicates an *or-node*. This or-node gives two models of $\mathcal{K}_2$ denoted $\mathcal{I}_l$ and $\mathcal{I}_r$ in the picture. If $\mathcal{K}_2 \models \boldsymbol{q}_n$, then $\boldsymbol{q}_n$ holds in both of them, and so there are homomorphisms $h_l \colon \boldsymbol{q}_n \to \mathcal{I}_l$ and $h_r \colon \boldsymbol{q}_n \to \mathcal{I}_r$. As $h_l(x_{n-1})$ and $h_r(x_{n-1})$ are instances of B_{n-1}, we have $B_{n-1} = T_1^{NM-1}$ in the picture, and so $\mathit{up}(T^{NM-1}) = \mathit{down}(T^{NM})$. By repeating this argument until x_0, we see that the colours between horizontal rows match and the rows are of the same length. (For this trick to work, we have to make the first *Row*-point in every branch an instance of *Start*.) In fact, an instance of the $N \times M$-tiling problem has a positive answer iff there exists $\boldsymbol{q}_n$ such that $\mathcal{K}_2 \models \boldsymbol{q}_n$. It is to be noted that to construct $\mathcal{T}_2$ with the properties described above, one needs quite a few auxiliary concept names.

Let $\mathcal{K}_1 = (\mathcal{T}_1, \{A(a)\})$ be an $\mathcal{EL}$ KB with the following canonical model:



where $\Sigma_0 = \{\mathit{Row}\} \cup \{T_k \mid T \in \mathfrak{T},\ k = 0, 1, 2\}$. Note that the vertical R-successors of the *Start*-points are not instances of any concept name, and so $\mathcal{K}_1$ does not satisfy any query of the form $\boldsymbol{q}_n$. On the other hand, $\mathcal{K}_2 \models \boldsymbol{q}$ implies $\mathcal{K}_1 \models \boldsymbol{q}$, for every Σ-CQ $\boldsymbol{q}$ without a subquery of the form $\boldsymbol{q}_n$ and $\Sigma = \mathsf{sig}(\mathcal{K}_1)$. Therefore, $\mathcal{K}_1$ Σ-CQ entails $\mathcal{K}_2$ iff there exists no tiling as specified above.

3 Future Work

We have made first steps towards understanding query entailment and inseparability for KBs and TBoxes in expressive DLs. From a theoretical viewpoint, it would be of interest to solve the open problems in Figures 1 and 2, and also consider other expressive DLs such as $\mathit{DL\text{-}Lite}_{bool}^{\mathcal{H}}$ or $\mathcal{ALCI}$. For example, our undecidability proof goes through for $\mathit{DL\text{-}Lite}_{bool}^{\mathcal{H}}$, but the other cases remain open. From a practical viewpoint, our model-theoretic criteria for query entailment are a good starting point for developing algorithms for approximations of query entailment based on simulations. Our undecidability and complexity results also indicate that rUCQ-entailment is more amenable to practical algorithms than, say, CQ-entailment and can be used as an approximation of the latter.

References

1. Arenas, M., Botoeva, E., Calvanese, D., Ryzhikov, V.: Exchanging OWL 2 QL knowledge bases. In: Proc. of the 23rd Int. Joint Conf. on Artificial Intelligence (IJCAI 2013). AAAI Press (2013)
2. Bienvenu, M., Rosati, R.: Query-based comparison of OBDA specifications. In: Proc. of the 28th Int. Workshop on Description Logics (DL 2015). vol. 1350. CEUR-WS (2015)
3. Botoeva, E., Kontchakov, R., Ryzhikov, V., Wolter, F., Zakharyaschev, M.: Query inseparability for description logic knowledge bases. In: Proc. of the 14th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2014) (2014)
4. Ghilardi, S., Lutz, C., Wolter, F.: Did I damage my ontology? A case for conservative extensions in description logics. In: Doherty, P., Mylopoulos, J., Welty, C. (eds.) Proc. of the 10th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2006). pp. 187–197 (2006)
5. Konev, B., Ludwig, M., Walther, D., Wolter, F.: The logical difference for the lightweight description logic EL. Journal of Artificial Intelligence Research (JAIR) 44, 633–708 (2012)
6. Kontchakov, R., Pulina, L., Sattler, U., Schneider, T., Seimer, P., Wolter, F., Zakharyaschev, M.: Minimal module extraction from *DL-Lite* ontologies using QBF solvers. In: Proc. of the 21st Int. Joint Conf. on Artificial Intelligence (IJCAI 2009). pp. 836–840 (2009)
7. Kontchakov, R., Wolter, F., Zakharyaschev, M.: Logic-based ontology comparison and module extraction, with an application to DL-Lite. Artificial Intelligence 174, 1093–1141 (2010)
8. Lutz, C., Wolter, F.: Deciding inseparability and conservative extensions in the description logic EL. J. of Symbolic Computation 45(2), 194–228 (2010)
9. Wang, K., Wang, Z., Topor, R.W., Pan, J.Z., Antoniou, G.: Eliminating concepts and roles from ontologies in expressive descriptive logics. Computational Intelligence 30(2), 205–232 (2014)