Query Inseparability for Description Logic Knowledge Bases

E. Botoeva, R. Kontchakov, V. Ryzhikov, F. Wolter and M. Zakharyaschev

1Faculty of Computer Science
Free University of Bozen-Bolzano, Italy
(botoeva,ryzhikov)@inf.unibz.it
2Department of Computer Science
University of Liverpool, U.K. wolter@liverpool.ac.uk
3Department of Computer Science
Birkbeck, University of London, U.K. (roman,michael)@dcs.bbk.ac.uk

Abstract

We investigate conjunctive query inseparability of description logic (DL) knowledge bases (KBs) with respect to a given signature, a fundamental problem for KB versioning, module extraction, forgetting and knowledge exchange. We study the data and combined complexity of deciding KB query inseparability for fragments of Horn-ALCHI, including the DLs underpinning OWL 2 QL and OWL 2 EL. While all of these DLs are P-complete for data complexity, the combined complexity ranges from P to EXPTIME and 2EXP TIME. We also resolve two major open problems for Horn-ALCHI: whether the data complexity of Horn-ALCHI is P-complete and whether the combination complexity of Horn-ALCHI is \( T \)-complete.

Introduction

A description logic (DL) knowledge base (KB) consists of a terminological box (TBox), storing conceptual knowledge, and an assertion box (ABox), storing data. Typical applications of KBs involve answering queries over incomplete data sources (ABoxes) augmented by ontologies (TBoxes) that provide additional information about the domain of interest as well as a convenient vocabulary for user queries. The standard query language in such applications, which balances expressiveness and computational complexity, is the language of conjunctive queries (CQs).

With typically large data, often tangled ontologies, and the hard problem of answering CQs over ontologies, various transformation and comparison tasks are becoming indispensable for KB engineering and maintenance. For example, to make answering certain CQs more efficient, one may want to extract from a given KB a smaller module returning the same answers to those CQs as the original KB; to provide the user with a more convenient query vocabulary, one may want to reformulate the KB in a new language. These tasks are known as module extraction (Arenas et al. 2012); other relevant tasks include versioning, revision and forgetting (Jiménez-Ruiz et al. 2011; Wang, Wang, and Topor 2010; Lin and Reiter 1994).

In this paper, we investigate the following relationship between KBs which is fundamental for all such tasks. Let \( \Sigma \) be a signature consisting of concept and role names. We call KBs \( K_1 \) and \( K_2 \) \( \Sigma \)-query inseparable and write \( K_1 \equiv_{\Sigma} K_2 \) if any CQ formulated in \( \Sigma \) has the same answers over \( K_1 \) and \( K_2 \). Note that even for \( \Sigma \) containing all concept and role names, \( \Sigma \)-query inseparability does not necessarily imply logical equivalence. The relativisation to (smaller) signatures is crucial to support the tasks mentioned above:

- (versioning) When comparing two versions \( K_1 \) and \( K_2 \) of a KB with respect to their answers to CQs in a relevant signature \( \Sigma \), the basic task is to check whether \( K_1 \equiv_{\Sigma} K_2 \).
- (modularisation) A \( \Sigma \)-module of a KB \( K \) is a KB \( K' \subseteq K \) such that \( K' \equiv_{\Sigma} K \). If we are only interested in answering CQs in \( \Sigma \) over \( K \), then we can achieve our aim by querying any \( \Sigma \)-module of \( K \) instead of \( K \) itself.
- (knowledge exchange) In knowledge exchange, we want to transform a KB \( K_1 \) in a signature \( \Sigma_1 \) to a new KB \( K_2 \) in a disjoint signature \( \Sigma_2 \) connected to \( \Sigma_1 \) via a declarative mapping specification given by a TBox \( T \). Thus, the target KB \( K_2 \) should satisfy the condition \( K_1 \sqcup T \equiv_{\Sigma_2} K_2 \), in which case it is called a universal UCQ-solution (CQ and UCQ inseparabilities coincide for Horn DLs).
- (forgetting) A KB \( K \) results from forgetting a signature \( \Sigma \) in a KB \( K \) if \( K' \equiv_{\Sigma} K \) and \( \text{sig}(K') \subseteq \text{sig}(K) \setminus \Sigma \). Thus, the result of forgetting \( \Sigma \) does not use \( \Sigma \) and gives the same answers to CQs without symbols in \( \Sigma \) as \( K' \).

We investigate the data and combined complexity of deciding \( \Sigma \)-query inseparability for KBs given in various fragments of the DL Horn-ALCHI (Krötzsch, Rudolph, and Hitzler 2013), which include DL-Lite\(_{\text{core}}\) (Calvanese et al. 2007) and \( \mathcal{EL} \) (Baader, Brandt, and Lutz 2005) underlying the W3C profiles OWL 2 QL and OWL 2 EL. For all of these DLs, \( \Sigma \)-query inseparability turns out to be P-complete for data complexity, which matches the data complexity of CQ evaluation for all of our DLs lying outside the DL-Lite family. For combined complexity, the obtained tight complexity results are summarised in the diagram below. Most interesting are EXP TIME-completeness of DL-Lite\(_{\text{core}}\) and 2EXP TIME-completeness of Horn-ALCHI, which contrast with NP-completeness and EXP TIME-completeness of CQ evaluation for those logics. For DL-Lite without role inclusions and \( \mathcal{EL} \), \( \Sigma \)-query inseparability is P-complete, while CQ evaluation is NP-complete. In general, it is the combined presence of inverse roles and qualified existential
restrictions (or role inclusions) that makes \(\Sigma\)-query inseparability hard. To establish the upper complexity bounds, we develop a uniform game-theoretic technique for checking finite \(\Sigma\)-homomorphic embeddability between (possibly infinite) materialisations of KBs.

\[
\Sigma \text{-query inseparability for KBs has not been investigated systematically before. The polynomial upper bound for } \mathcal{EL} \text{ was established as a preliminary step to study TBox inseparability (Lutz and Wolter 2010), and this notion was also used to study forgetting for } DL-Lite^{\text{horn}} \text{ (Wang et al. 2010).}
\]

We apply our results to resolve two important open problems. First, we show that the membership problem for universal UCQ-solutions in knowledge exchange for KBs in \(DL-Lite^{\text{horn}}\) is \(\text{EXPTIME}\)-complete for combined complexity, which settles an open question of (Arenas et al. 2013), where only \(\text{PSPACE}\)-hardness was established. We also show that \(\Sigma\)-query inseparability of \(DL-Lite^{\text{horn}}\) TBoxes is \(\text{EXPTIME}\)-complete, which closes the \(\text{PSPACE–EXPTIME}\) gap that was left open by Konev et al. (2011).

Recall that TBoxes \(T_1\) and \(T_2\) are \(\Sigma\)-query inseparable if, for all \(\Sigma\)-ABoxes \(A\) (which only use concept and role names from \(\Sigma\)), the KBs \((T_1, A)\) and \((T_2, A)\) are \(\Sigma\)-query inseparable. TBox and KB inseparabilities have different applications. The former supports ontology engineering when data is not known or changes frequently: one can equivalently replace one TBox with another only if they return the same answers to queries for every \(\Sigma\)-ABox. In contrast, KB inseparability is useful in applications where data is stable such as knowledge exchange, module extraction or forgetting for a stable KB in order to re-use it in a new application or as a compilation step to make CQ answering more efficient. As we show below, TBox and KB \(\Sigma\)-query inseparabilities also have different computational properties.

TBox \(\Sigma\)-query inseparability has been extensively studied (Kontchakov, Wolter, and Zakharyaschev 2008; Lutz and Wolter 2010; Konev et al. 2012). For work on different notions of TBox inseparability and the corresponding notions of modules and forgetting, we refer the reader to (Cuenca Grau et al. 2008; Konev, Wolter, and Wolter 2009; Del Vescovo et al. 2011; Nikitina and Rudolph 2012; Nikitina and Glimm 2012; Lutz, Seylan, and Wolter 2012).

Omitted proofs can be found in the full version available at www.dcs.bbk.ac.uk/~roman

**Horn-ALCHI and its Fragments**

All the DLs for which we investigate KB \(\Sigma\)-query inseparability are Horn fragments of \(ALCHI\). To define these DLs, we fix sequences of individual names \(a_i\), concept names \(A_i\), and role names \(P_i\), where \(i < \omega\). A role is either a role name \(P_i\) or an inverse role \(P_i^{-1}\); we assume that \((P_i^{-1})^{-1} = P_i\). \(ALCI\)-concepts, \(C\), are defined by the grammar

\[
C ::= A_1 | T | \bot | \neg C | C_1 \sqcap C_2 | C_1 \sqcup C_2 | \exists R.C | \forall R.C,
\]

where \(R\) is a role. \(ACCI\)-concepts are \(ALCI\)-concepts without out inverse roles; \(\mathcal{E}\)-concepts are \(ALCI\)-concepts without the constructs \(\sqcap, \sqcup, \neg\) and \(\forall R\). \(DL-Lite^{\text{horn}}\)-concepts are \(ALCI\)-concepts without \(\sqcup\) and \(\forall R\); in which \(C = T\) in every occurrence of \(\exists R.C\). Finally, \(DL-Lite^{\text{horn}}\)-concepts are \(ALCI\)-concepts without \(\forall\); in other words, they are basic concepts of the form \(\bot, T, A_i\) or \(\exists R\).

For a DL \(\mathcal{L}\), an \(\mathcal{L}\)-concept inclusion (CI) takes the form \(C \subseteq D\), where \(C\) and \(D\) are \(\mathcal{L}\)-concepts. An \(\mathcal{L}\)-Box, \(T\), contains a finite set of \(\mathcal{L}\)-CIs. An \(ALCHI\), \(DL-Lite^{\text{horn}}\), and \(DL-Lite^{\text{core}}\) TBox can also contain a finite set of role inclusions (RIs) \(R_1 \sqsubseteq R_2\), where the \(R_i\) are roles. In \(\mathcal{EL}\) TBoxes, RIs do not have inverse roles. \(DL-Lite^{\text{horn}}\) TBoxes may also contain disjointness constraints \(B_1 \cap B_2 \sqsubseteq \bot\) and \(R_1 \sqsubseteq R_2 \sqsubseteq \bot\), for basic concepts \(B_i\) and roles \(R_i\).

To introduce the Horn fragments of these DLs, we require the following (standard) recursive definition (Hustadt, Motik, and Sattler 2005; Kazakov 2009): a concept \(C\) occurs positively in \(\mathcal{C}\); if \(C\) occurs positively (respectively, negatively) in \(C'\) then \(C\) occurs positively (negatively) in \(C' \sqcup D, C' \sqcap D, \exists R.C', \forall R.C', D \sqsubseteq C'\), and \(C\) occurs positively (negatively) in \(\neg C'\) and \(C' \sqsubseteq D\). Now, we call a TBox \(T\) Horn if no concept of the form \(C \sqcup D\) occurs positively in \(T\), and no concept of the form \(\neg C\) or \(\forall R.C\) occurs negatively in \(T\). In the DL Horn-\(\mathcal{L}\), where \(\mathcal{L}\) is one of our DLs, only Horn \(\mathcal{L}\) TBoxes are allowed. Clearly, the \(\mathcal{EL}\) and \(DL-Lite^{\text{horn}}\) TBoxes are Horn by definition.

An \(\mathcal{ABox}\), \(A\), is a finite set of assertions of the form \(A_k(a_i)\) or \(P_k(a_i, a_j)\). An \(\mathcal{L}\)-Box \(T\) and an \(\mathcal{ABox}\) \(A\) together form an \(\mathcal{L}\) knowledge base (KB) \(\mathcal{K} = (T, A)\). The set of individual names in \(\mathcal{K}\) is denoted by \(\text{ind}(\mathcal{K})\).

The semantics for the DLs is defined in the usual way based on interpretations \(I = (\Delta^I, \mathcal{I})\) that comply with the unique name assumption: \(a_i^n \neq a_i^j\) for \(i \neq j\) (Baader et al. 2003). We write \(I \models \alpha\) in case an inclusion or assertion \(\alpha\) is true in \(I\). If \(I \models \alpha\), for all \(\alpha \in T \cup A\), then \(I\) is a model of a KB \(\mathcal{K} = (T, A)\); in symbols: \(I \models \mathcal{K}\). \(\mathcal{K}\) is consistent if it has a model. \(\mathcal{K} \models \alpha\) means that \(I \models \alpha\) for all \(I \models \mathcal{K}\).

A conjunctive query (CQ) \(q(\vec{x})\) is a formula \(\exists \vec{y} \phi(\vec{x}, \vec{y})\), where \(\phi\) is a conjunction of atoms of the form \(A_k(z_1)\) or \(P_k(z_1, z_2)\) with \(z_1 \in \vec{x} \cup \vec{y}\). A tuple \(\alpha \in \text{ind}(\mathcal{K})\) (of the same length as \(\vec{x}\)) is a certain answer to \(q(\vec{x})\) over \(\mathcal{K} = (T, A)\) if \(I \models q(\alpha)\) for all \(I \models \mathcal{K}\); in this case we write \(\mathcal{K} \models q(\vec{a})\). If \(\vec{x} = \emptyset\), the answer to \(q\) is ‘yes’ if \(\mathcal{K} \models q\) and ‘no’ otherwise.

For combined complexity, the problem \(\mathcal{K} \models q(\vec{a})?\) is NP-complete for the \(DL-Lite\) logics (Calvanese et al. 2007), \(\mathcal{EL}\) and \(\mathcal{ELH}\) (Rosati 2007), and \(\text{EXPTIME}\)-complete for the remaining Horn DLs above (Eiter et al. 2008). For data complexity (with fixed \(T\) and \(q\)), this problem is in \(\text{AC}^0\) for the \(DL-Lite\) logics (Calvanese et al. 2007) and \(\text{P}\)-complete for the remaining DLs (Rosati 2007; Eiter et al. 2008).

A signature, \(\Sigma\), is a set of concept and role names. By a \(\Sigma\)-concept, \(\Sigma\)-role, \(\Sigma\)-CQ, etc. we understand any concept, role, CQ, etc. constructed using the names from \(\Sigma\).
\begin{center}{\bf \Sigma\nobreakdash-Query Entailment and Inseparability}\\
We define the central notions of this paper.
\end{center}

\begin{definition} \label{def:univ}
Let $K_1$ and $K_2$ be KBs and $\Sigma$ a signature.
\begin{itemize}
\item $K_1$ $\Sigma$-query entails $K_2$ if $K_2 \models q(\bar{a})$ implies $K_1 \models q(\bar{a})$
\quad for all $\Sigma$-CQs $q(\bar{x})$ and all $\bar{a} \in \text{ind}(K_2)$.
\item $K_1$ and $K_2$ are $\Sigma$-query inseparable if they $\Sigma$-query entail each other. In this case we write $K_1 \equiv_{\Sigma} K_2$.
\end{itemize}
Observe that $\Sigma$-query inseparability is weaker than logical equivalence even if $\Sigma = \text{sig}(K_1) \cup \text{sig}(K_2)$, where $\text{sig}(K_1)$ is the signature of $K_1$. For example, $(\emptyset, \{A(a)\})$ is a $(B \sqcap A)$-query inseparable from $(\{B \sqcap A\}, \{A(a)\})$ but the two KBs are clearly not logically equivalent. Since checking $\Sigma$-query inseparability can be reduced to two $\Sigma$-query entailment checks, we can prove complexity upper bounds for entailment. Conversely, for most languages we have a semantically transparent reduction of $\Sigma$-query entailment to $\Sigma$-query inseparability:
\begin{center}{\bf Theorem 2} \label{thm:entailment}
Let $\mathcal{L}$ be any of our DLs containing $\mathcal{E}$ or having role inclusions. Then $\Sigma$-query entailment for $\mathcal{L}$-KBs is \textsc{LogSpace}-reducible to $\Sigma$-query inseparability for $\mathcal{L}$-KBs.
\end{center}
\begin{proof}[Proof sketch.] Let $K_i = (T_i, A_i), \ i = 1, 2,$ and $\Sigma$ be given. We may assume that $\Sigma = \text{sig}(K_1) \cap \text{sig}(K_2)$. We also assume that $\mathcal{L}$ has role inclusions, $K_1$ and $K_2$ are consistent and the trivial interpretation $T_0$ (with $|\Delta^{T_0}| = 1$ and $S^{T_0} = \emptyset$, for any $S$) is a model of the $T_i$ (a proof without those assumptions is given in the full version). Let $K'_i$ be a copy of $K_i$, in which all symbols $S_i$ are replaced by fresh $S_i$, and let $K_i$ extend $K'_i$ with $S_i \subseteq S$, for $S \in \Sigma$. One can show that $K_i$ $\Sigma$-query entails $K_2$ iff $K_1 \equiv_{\Sigma} K_1^{S_i} \cap K_2^{S_i}$.

That $T_0 \models K_i$ is essential in the reduction. Take $T_1 = \{A \sqsubseteq B, A \sqsubseteq \exists R.C, T_2 = \{T \subseteq B, C \sqcap B \subseteq \bot\}$ and $\Sigma = \{A, B, R, C\}$. Then $K_1 = (T_1, \{A(a)\})$ $\Sigma$-query entails $K_2 = (T_2, \{A(a)\})$ but $K_1 \not\equiv_{\Sigma} K_1^{S} \cap K_2^{S}$.

We now consider the relationship between inseparability and universal UCQ-solutions in knowledge exchange. Suppose $K_1$ and $K_2$ are KBs in disjoint signatures $\Sigma_1$ and $\Sigma_2$. Let $T_{12}$ be a mapping consisting of inclusions of the form $S_1 \subseteq S_2$, where the $S_i$ are concept (or role) names in $\Sigma_i$. Then $K_{12}$ is a universal UCQ-solution for $(K_1, T_{12}, \Sigma_2)$ if $K_1 \cup T_{12} \equiv_{\Sigma_2} K_{12}$ Deciding the latter is called the membership problem for universal UCQ-solutions. For DLs $\mathcal{L}$ with role inclusions, the problem whether $K_1 \cup T_{12} \equiv_{\Sigma_2} K_{12}$ is a $\Sigma_2$-query inseparability problem in $\mathcal{L}$. Conversely, we have:
\begin{center}{\bf Theorem 3} \label{thm:universalIssatisifiable}
$\Sigma$-query entailment for any of our DLs $\mathcal{L}$ is \textsc{LogSpace}-reducible to the membership problem for universal UCQ-solutions in $\mathcal{L}$.
\end{center}
\begin{proof}[Proof sketch.] We want do decide whether $K_1$ $\Sigma$-query entails $K_2$. We again assume that $T_0 \models T_i$ and use the proof of Theorem 2 (for the general case, see the full version). We may assume that $\Sigma = \text{sig}(K_1) \cap \text{sig}(K_2)$. Let $S_1 = \text{sig}(K_1)$. Then $K_1$ $\Sigma$-query entails $K_2$ iff $K_1$ $\Sigma_1$-query entails $K_2$. By the proof of Theorem 2, the latter is the case iff $K_1$ $\Sigma_1$-query entails $K_{12}^{S_1} \cup K_{12}^{S_2}$. Clearly, $K_2^{S_1} \cap K_2^{S_2}$ $\Sigma$-query entails $K_1$, and so the two KBs are $\Sigma_1$-query inseparable. Then $K_1$ $\Sigma$-query entails $K_2$ if $K_1$ is a universal UCQ-solution for $(K_1, T_{12}, \Sigma_2)$, where $T_{12} = \{S_1 \subseteq S, S_2 \subseteq S \mid S \in \Sigma_1\}$.

\begin{center}{\bf Semantic Characterisation}\\
In this section, we give a semantic characterisation of KB $\Sigma$-query entailment based on an abstract notion of materialisation and finite homomorphisms between such models.
\end{center}
Let $K$ be a KB. An interpretation $I$ is called a materialisation of $K$ if, for all CQs $q(x)$ and tuples $\bar{a} \in \text{ind}(K)$, $K \models q(\bar{a})$ iff $I \models q(\bar{a})$.

We say that $K$ is materialisable if it has a materialisation.

Materialisations can be used to characterise KB $\Sigma$-query entailment by means of $\Sigma$-homomorphisms. For an interpretation $I$ and a signature $\Sigma$, the $\Sigma$-types $t_2^I(x)$ and $r_2^I(x, y)$ of $x, y \in \Delta^I$ are defined by taking:
\begin{align*}
t_2^I(x) &= \{ \Sigma\text{-concept name } A \mid x \in A^I \}, \\
r_2^I(x, y) &= \{ \Sigma\text{-role } R \mid (x, y) \in R^I \}.
\end{align*}

Suppose $I_i$ is a materialisation of $K_i$, $i = 1, 2$. A function $h : \Delta^{I_2} \rightarrow \Delta^{I_1}$ is a $\Sigma$-homomorphism from $I_2$ to $I_1$, if, for any $a \in \text{ind}(K_2)$ and any $x, y \in \Delta^{I_2}$,
\begin{enumerate}
\item $h(a^{I_2}) = a^{I_1}$ whenever $t_2^I(a) \not= 0$ or $r_2^I(a, y) \not= 0$ for some $y \in \Delta^{I_2}$, and
\item $t_2^I(x) \subseteq t_1^I(h(x))$, $r_2^I(x, y) \subseteq r_1^I(h(x), h(y))$.
\end{enumerate}

As answers to $\Sigma$-CQs are preserved under $\Sigma$-homomorphisms, $K_1$ $\Sigma$-query entails $K_2$ if there is a $\Sigma$-homomorphism from $I_2$ to $I_1$. However, the converse does not hold:
\begin{example} \label{ex:2}
Suppose $I_2$ and $I_1$ below are materialisations of KBs $K_2$ and $K_1$, where $a$ is the only ABox individual:
\begin{center}
\begin{tabular}{c|c|c|c}
  & $T$ & $Q$ & $P$ \\
\hline
$a$ & $\bullet$ & $\bullet$ & $\bullet$ \\
\end{tabular}
\end{center}
Let $\Sigma = \{Q, R, S, T\}$. Then there is no $\Sigma$-homomorphism from $I_2$ to $I_1$ (as $r_2^I(a, u) = 0$, we can map $u$ to, say, $x$ but then only the shaded part of $I_2$ can be mapped $\Sigma$-homomorphically to $I_1$). However, for any $\Sigma$-query $q(\bar{x})$, $I_2 \models q(\bar{a})$ implies $I_1 \models q(\bar{a})$ as any finite subinterpretation of $I_2$ can be $\Sigma$-homomorphically mapped to $I_1$.

We say that $I_2$ is finitely $\Sigma$-homomorphically embeddable into $I_1$ if, for every finite subinterpretation $I_2'$ of $I_2$, there exists a $\Sigma$-homomorphism from $I_2'$ to $I_1$.

To prove the following theorem, one can regard any finite subinterpretation of $I_2$ as a CQ whose variables are elements of $\Delta^{I_2}$, with the answer variables being in $\text{ind}(K_2)$.
\begin{theorem} \label{thm:finiteEmbeddable}
Suppose $K_i$ is a consistent KB with a materialisation $I_i$, $i = 1, 2$. Then $K_1$ $\Sigma$-query entails $K_2$ iff $I_2$ is finitely $\Sigma$-homomorphically embeddable into $I_1$.
\end{theorem}
One problem with applying Theorem 5 is that materialisations are in general infinite for any of the DLs considered in this paper. We address this problem by introducing finite representations of materialisations. Let $K = (T, A)$ be a KB and let $G = (\Delta^G, \varphi, \rightarrow)$ be a finite structure such that $\Delta^G = \text{ind}(K) \cup \Omega$, for $\Omega$, $\rightarrow$ is an interpretation
Suppose a DL $\mathcal{L}$ has finitely generated materialisations, $\mathcal{K}_i$ is a consistent $\mathcal{L}$-KB, for $i = 1, 2$, and $\Sigma$ a signature. Let $\mathcal{G}_i = (\Delta^{\mathcal{G}_i}, \bar{u}_i, \bar{w}_i)$ be a generating structure for $\mathcal{K}_i$ and let $\mathcal{M}_i$ be its materialisation; $\mathcal{G}_i^\Sigma$ and $\mathcal{M}_i^\Sigma$ denote the restrictions of $\mathcal{G}_i$ and $\mathcal{M}_i$ to $\Sigma$.

We begin with a very simple game on the finite generating structure $\mathcal{G}_2^\Sigma$ and the possibly infinite materialisation $\mathcal{M}_1^\Sigma$.

**Infinite game $G_{\Sigma}(\mathcal{G}_2,\mathcal{M}_1)$**. This game is played by two players: player 2 and player 1. The states of the game are of the form $s_i = (u_i \rightarrow \sigma_i)$, for $i \geq 0$, where $u_i \in \Delta^{\mathcal{G}_2}$ and $\sigma_i \in \Delta^{\mathcal{M}_1}$ satisfy the following condition:

$$(s_1) \quad t^\Sigma_{\mathcal{G}_2}(u_i) \subseteq t^\Sigma_{\mathcal{M}_1}(\sigma_i).$$

The game starts in a state $s_0 = (u_0 \rightarrow \sigma_0)$ with $\sigma_0 = u_0$ in case $u_0 \in \text{ind}(\mathcal{K}_2)$. In each round $i > 0$, player 2 challenges player 1 with some $u_i \in \Delta^{\mathcal{G}_2}$ such that $u_{i-1} \rightarrow \sigma^{\mathcal{G}_2}_{\mathcal{M}_1}(u_i)$. Player 1 has to respond with a $\sigma_i \in \Delta^{\mathcal{M}_1}$ satisfying $(s_1)$ and

$$(s_2) \quad t^\Sigma_{\mathcal{G}_2}(u_i, u_i) \subseteq t^\Sigma_{\mathcal{M}_1}(\sigma_{i-1}, \sigma_i).$$

This gives the next state $s_i = (u_i \rightarrow \sigma_i)$. Note that all of the $u_i$'s may be ABox individuals; however, there is no such a restriction on the $\sigma_i$. A play of length $n > 0$ starting from $s_0$ has any sequence $s_0, \ldots, s_n$ of states obtained as described above. For an ordinal $\lambda \leq \omega$, we say that player 1 has a $\lambda$-winning strategy in the game $G_{\Sigma}(\mathcal{G}_2,\mathcal{M}_1)$ starting from a state $s_0$ if, for any play of length $i < \lambda$, which starts from $s_0$ and conforms with this strategy, and any challenge of player 2 in round $i + 1$, player 1 has a response.

The following theorem gives a game-theoretic flavour to the criterion of Theorem 5 (see the full paper for a proof).

**Theorem 8** $\mathcal{M}_2$ is finitely $\Sigma$-homomorphically embeddable into $\mathcal{M}_1$ iff the following conditions hold:

- **(abox)** $r^\Sigma_{\mathcal{M}_2}(a, b) \subseteq r^\Sigma_{\mathcal{M}_1}(a, b)$, for any $a, b \in \text{ind}(\mathcal{K}_2)$;
- **(win)** for any $u_0 \in \Delta^{\mathcal{G}_2}$ and $n < \omega$, there exists $\sigma_0 \in \Delta^{\mathcal{M}_1}$ such that player 1 has an $n$-winning strategy in the game $G_{\Sigma}(\mathcal{G}_2,\mathcal{M}_1)$ starting from $(u_0 \rightarrow \sigma_0)$.

**Example 9** Let $\Sigma = \{Q, R, S, T\}$. Consider $G_{\Sigma}^\mathcal{G}_2$ and $\mathcal{M}_2^\Sigma$ shown in the picture below:

For any $n < \omega$ and $u \in \Delta^{\mathcal{G}_2}$, player 1 has an $n$-winning strategy in $G_{\Sigma}(\mathcal{G}_2,\mathcal{M}_1)$. A 4-winning strategy starting from $(u \rightarrow \sigma)$ is shown by dotted lines (in round 2, player 2 has two possible challenges). For a larger $n$, a suitable $\sigma$ can be chosen further away from the root $a$ of $\mathcal{M}_1$. The criterion of Theorem 8 does not seem to be a big improvement on Theorem 5 as we still have to deal with an infinite materialisation. Our aim now is to show that condition **(win)** in the infinite game $G_{\Sigma}(\mathcal{G}_2,\mathcal{M}_1)$ can be checked.
by analysing a more complex game on the finite generating structures \(G_2\) and \(G_1\). We consider four types of strategies in \(G_{\Sigma}(G_2, M_1)\). For each type, \(\tau\), we define a game \(G_{\Sigma}(G_2, G_1)\) such that, for any \(u_0 \in \Delta^0\), the following conditions are equivalent:

\[
\begin{align*}
&(< \omega) \text{ for every } n < \omega, \text{ player } 1 \text{ has an } n\text{-winning strategy of type } \tau \text{ in } G_{\Sigma}(G_2, M_1) \text{ starting from some } (u_0 \rightarrow \sigma^0_0); \\
&(\omega) \text{ player } 1 \text{ has an } \omega\text{-winning strategy in } G_{\Sigma}(G_2, G_1) \text{ starting from some state depending on } u_0 \text{ and } \tau.
\end{align*}
\]

We start by considering ‘forward’ winning strategies that are sufficient for the DLs without inverse roles.

**Forward strategy and game** \(G_{\Sigma}(G_2, G_1)\). We say that a \(\lambda\)-strategy (\(\lambda \leq \omega\)) for player 1 in the game \(G_{\Sigma}=(G_2, M_1)\) is forward if, for any play of length \(i-1 < \lambda\), which conforms with this strategy, and any challenge \(u_{i-1} \sim^\Sigma u_i\) by player 2, the response \(\sigma_i\) of player 1 is such that either \(\sigma_i \sim \sigma_i\) or \(\sigma_i = \sigma_i\), for some \(v \in \Delta^\Sigma\).

For example, if the \(G_i\), \(i = 1, 2\), satisfy the condition

\[(\text{f}) \text{ the } \Sigma\text{-labels on } \sim^\Sigma \text{-edges contain no inverse roles, then every strategy in } G_{\Sigma}(G_2, M_1) \text{ is forward. This is clearly the case for Horn-ALCH, Horn-ALC, ELCH and ECL, which by definition do not have inverse roles.}
\]

The existence of a forward \(\lambda\)-winning strategy for player 1 in \(G_{\Sigma}(G_2, M_1)\) is equivalent to the existence of such a strategy in the game \(G_{\Sigma}(G_2, G_1)\), which is defined similarly to \(G_{\Sigma}(G_2, M_1)\) but with two modifications: (1) it is played on \(G_2\) and \(G_1\); and (2) the response \(x_i \in \Delta^\Sigma\) of player 1 to a challenge \(x_{i-1} \sim^\Sigma u_i\) must be such that either \(x_{i-1}, x_i \in \text{ind}(K_1)\) or \(x_{i-1} \sim^\Sigma x_i\), and (s1)-(s2) hold (with \(G_1\) and \(x_i\) in place of \(M_1\) and \(\sigma_i\)).

**Example 10** Let \(G_2\) and \(G_1\) be as shown below. Then, for any \(u \in \Delta^2\), there is \(x_i \in \Delta^\Sigma\) such that player 1 has an \(\omega\)-winning strategy in \(G_{\Sigma}(G_2, G_1)\) starting from \((u \rightarrow \sigma)\).

\[
\begin{align*}
&G_2^\Sigma \\
&\text{---------} \\
&\text{---------} \\
&G_1^\Sigma
\end{align*}
\]

The next theorem follows from König’s Lemma:

**Lemma 11** For \(u_0 \in \Delta^2\), condition \((< \omega)\) holds for forward strategies in \(G_{\Sigma}(G_2, M_1)\) iff \((\omega)\) holds in \(G_{\Sigma}(G_2, G_1)\) for some state \((u_0 \rightarrow \sigma_0)\).

\(G_{\Sigma}(G_2, G_1)\) is a standard simulation or reachability game on finite graphs, where the existence of \(\omega\)-winning strategies for player 1 follows from the existence of \(n\)-winning strategies for \(n = O(|G_2| \times |G_1|)\), which can be checked in polynomial time (Mazala 2001; Baier and Katoen 2007). By Theorem 6 and (f), we obtain:

**Theorem 12** For combined complexity, checking \(\Sigma\)-query entailment is in \(P\) for \(\mathcal{E}\) and \(\mathcal{E}\mathcal{L}\) KBs, and in \(\text{EXPTIME}\) for Horn-\(\mathcal{A}\)-\(\mathcal{L}\) and Horn-\(\mathcal{A}\)-\(\mathcal{L}\)\(\mathcal{C}\) KBs. For data complexity, it is in \(P\) for all these DLs.

In comparison to forward strategies, the winning strategies used in Example 9 can be described as ‘backward.’

**Backward strategy and game** \(G_{\Sigma}(G_2, G_1)\). A \(\lambda\)-strategy for player 1 in \(G_{\Sigma}(G_2, M_1)\) is backward if, for any play of length \(i = 1 < \lambda\), which conforms with this strategy, and any challenge \(u_{i-1} \sim^\Sigma u_i\) by player 2, the response \(\sigma_i\) of player 1 is the immediate predecessor of \(\sigma_{i-1}\) in \(M_1\) in the sense that \(\sigma_{i-1} = \sigma_i w\) for some \(w \in \Delta^2\) (player 1 loses in case \(\sigma_{i-1} \in \text{ind}(K_1)\)). Note that, since \(M_1\) is tree-shaped, the response of player 1 to any different challenge \(u_{i-1} \sim^\Sigma u_i\) must be the same \(\sigma_i\); cf. Example 9.

That is why the states of the game \(G_{\Sigma}(G_2, G_1)\) are of the form \(s_i = (\Xi_i \mapsto x_i)\), where \(\Xi_i \subseteq \Delta^2\), \(\Xi_i \neq \emptyset\), and \(x_i \in \Delta^\Xi\) satisfy the following condition:

\[(s_i^1) \quad \tau_{\Xi_i}^1(u) \subseteq \tau_{\Xi_i}^1(x_i), \text{ for all } u \in \Xi_i.
\]

The game starts in a state \(s_0 = (\Xi_0 \mapsto x_0)\) such that \((s_0)\) if \(u \in \Xi_0 \cap \text{ind}(K_2)\), then \(x_0 = u \in \text{ind}(K_2)\).

For each \(i \geq 0\), player 2 always challenges player 1 with the set \(\Xi_i = \Xi_{i-1}^\Sigma\), where

\[
\Xi_{i}^\Sigma = \{v \in \Delta^2 | u \sim^\Sigma v, \text{ for some } u \in \Xi_i\},
\]

provided that it is not empty (otherwise, player 2 loses).

Player 1 responds with \(x_i \in \Delta^\Xi\) such that \(x_i \sim^\Xi x_{i-1}\) and \((s_i)-(s_2)\) and the following condition hold:

\[(s_i^2) \quad r_{\Xi_i}^2(u, v) \subseteq r_{\Xi_i}^2(x_{i-1}, x_i), \text{ for all } u \in \Xi_{i-1}, v \in \Xi_i.
\]

**Lemma 13** For \(u_0 \in \Delta^2\), condition \((< \omega)\) holds for backward strategies in \(G_{\Sigma}(G_2, M_1)\) iff \((\omega)\) holds in \(G_{\Sigma}(G_2, G_1)\) for some state \((u_0 \mapsto x_0)\).

Although Lemmas 11 and 13 look similar, the game \(G_{\Sigma}(G_2, G_1)\) turns out to be more complex than \(G_{\Sigma}(G_2, G_1)\).

**Example 14** To illustrate, consider \(G_{\Sigma}^2(G_2, G_1)\) shown below (with concepts and roles omitted) and an arbitrary \(G_1^1\):

\[
\begin{align*}
&G_2^2 \\
&\text{---------} \\
&\text{---------} \\
&G_1^1
\end{align*}
\]

A play in \(G_{\Sigma}^2(G_2, G_1)\) may proceed as: \((\{u \mapsto x_0\}, \{v_1, v_3\} \mapsto x_1), \{v_1, v_3\} \mapsto x_2, \{v_1, v_3\} \mapsto x_3\), etc. This gives at least 6 different sets \(\Xi_i\). But if \(G_2^2\) contained \(k\) cycles of lengths \(p_1, \ldots, p_k\), where \(p_i\) is the \(i\)th prime number, then the number of states in \(G_{\Sigma}^2(G_2, G_1)\) could be exponential (\(p_1 \times \cdots \times p_k\)). In fact, we have the following:

**Lemma 15** Checking \((\omega)\) in Lemma 13 is \(\text{CONP-hard}\).

Observe that in the case of \(\text{DL-Lite}\) and \(\text{DL-Lite}\) (which have inverse roles but no RIs), generating structures \(G = (\Delta^2, G, \sim)\) can be defined so that, for any \(u \in \Delta^2\) and \(R\), there is at most one \(v \in \Delta^2\) with \(u \rightarrow v\) and \(R \in \tau^{\Delta^2}(u, v)\) (Kontchakov et al. 2010). As a result, any \(n\)-winning strategy starting from \((u_0 \mapsto \sigma_0)\) in \(G_{\Sigma}(G_2, M_1)\) consists of a (possibly empty) backward part followed by a (possibly empty) forward part. Moreover, in the backward games for these DLs, the sets \(\Xi_i\) are always singletons. Thus, the number of states in the combined backward/forward games on the \(G_i\) is polynomial, and the existence of winning strategies can be checked in polynomial time.
Theorem 16 Checking $\Sigma$-query entailment for DL-Lite$_{core}$ and DL-Lite$_{horn}$ KBs is in P for both combined and data complexity.

An arbitrary strategy for player 1 in $G_S^2(G_2, M_1)$ is a combination of a backward strategy and a number of start-bounded strategies to be defined next.

Start-bounded strategy and game $G_S^2(G_2, G_1)$. A strategy for player 1 in the game $G_S^2(G_2, M_1)$ starting from a state $(u_0 \mapsto \sigma_0)$ is start-bounded if it never leads to $(u_i \mapsto \sigma_i)$ such that $\sigma_0 = \sigma_i v$, for some $v$ and $i > 0$. In other words, player 1 cannot use those elements of $M_1$ that are located closer to the ABox than $\sigma_0$; the ABox individuals in $M_1$ can only be used if $\sigma_0 \in \text{ind}(K_1)$.

Example 17 The strategy starting from $(u_2 \Rightarrow \sigma_1)$ and shown below is start-bounded:

$G_S^2$

In the game $G_S^2(G_2, G_1)$, player 1 will have to guess all the points of $G_2$ that are mapped to the same point of $M_1$.

The states of $G_S^2(G_2, G_1)$ are of the form $(\Gamma_1, \Xi_i \mapsto x_i)$, $i > 0$, where $\Gamma_1, \Xi_i \subseteq \Delta^{G_2}$, $\Xi_i \not\subseteq \emptyset$, $x_i \in \Delta^{G_1}$, and $(\Psi_i)$ holds. The initial state is of the form $(\emptyset, \Xi_0 \mapsto x_0)$ such that $(\Psi_0)$ holds. In each round $i > 0$, player 2 challenges player 1 with some $u \sim^{G_2} v$ such that $u \in \Xi_1$ and $(\text{nbk})$ if $v \in \Gamma_1$, then $r^{G_2}(u, v) \subseteq r^{G_1}(x_1, x_2)$. Player 1 responds with either a state $(\Xi_{i-1}, \Xi_i \mapsto x_i)$ such that $x_{i-1} \sim^{G_1} x_i$ and $(\Psi_i)$, or a state $(\emptyset, \Xi_i \mapsto x_i)$ such that $x_{i-1}, x_i \in \text{ind}(K_1)$ and $(s_i' \Psi_i)$ holds. We make challenges $u \sim^{G_2} v$, for which $u \in \Xi_1$, and $(\text{nbk})$ does not hold, ‘illegitimate’ because $x_1 < 2$ can always be used as a response. Because of this, player 1 always moves ‘forward’ in $G_1$, but has to guess appropriate sets $\Xi$ in advance. Note that $\Gamma_1$ is always uniquely determined by $x_{i-1}, x_i$ and $\Xi_2$ (and it is either $\Xi_1$ or empty).

Example 18 Let $G_S^2$ and $G_2$ be as follows (cf. Example 17):

We show that player 1 has an $\omega$-winning strategy in $G_S^2(G_2, G_1)$ starting from $(\emptyset, \{u_2, u_9\} \mapsto x_1)$. Player 2 challenges with $u_2 \sim^{G_2} u_6$, and player 1 responds with $\{u_2, u_3\}$, $\{u_6, u_8\}$ and player 1 responds with $\{u_6, u_8\}$, $\{u_7, u_9\}$ and $\{u_7, u_9\}$, where the game ends. Note the crucial guesses $\{u_2, u_3\} \mapsto x_1$ and $\{u_6, u_8\} \mapsto x_3$ made by player 1. If player 1 responded with $\{u_2, u_3\}$, $\{u_6\} \mapsto x_3$ (and failed to guess that $u_8$ must also be mapped to $x_3$), then after the challenge $u_6 \sim^{G_2} u_7$ and response $(\{u_6, u_8\}, \{u_7\} \mapsto x_3)$, player 2 would pick $u_7 \sim^{G_2} u_8$, to which player 1 could not respond.

Lemma 19 For any $u_0 \in \Delta^{G_2}$, condition $(< \omega)$ holds for start-bounded strategies in $G_S^2(G_2, M_1)$ iff $(\omega)$ holds in $G_S^2(G_2, G_1)$ for some state $(\emptyset, \Xi_0 \mapsto x_0)$ with $u_0 \in \Xi_0$.

As we shall see in the next section, the problem of checking the conditions of this lemma is EXPTIME-hard.

Arbitrary strategies and game $G_S^2(G_2, G_1)$. An arbitrary winning strategy in the game $G_S^2(G_2, M_1)$ can be composed of one backward and a number of start-bounded strategies.

Example 20 Consider $G_S^2$ and $M_1^2$ shown below:

Starting from $(u_1 \mapsto \sigma_1)$, player 1 can respond to the challenges $u_1 \sim^{G_2} u_5$, $u_5 \sim^{G_2} u_6$ according to the backward strategy; the challenges $u_2 \sim^{G_2} u_6$, $u_6 \sim^{G_2} u_7$ according to the start-bounded strategy as in Example 17; the challenges $u_3 \sim^{G_2} u_4$ according to the obvious start-bounded strategy; finally, the challenge $u_9 \sim^{G_2} u_{10}$ needs a response according to the backward strategy. We will combine the two backward strategies into a single one, but keep the start-bounded ones separate.

The game $G_S^2(G_2, G_1)$ begins as $G_S^2(G_2, G_1)$, but with states of the form $(\Xi, \Psi_1)$, $i > 0$, where $\Xi_i \subseteq \Delta^{G_2}$ and $x_i \in \Delta^{G_1}$ satisfy $(s_i')$ and $\Psi_i$ is a (possibly empty) subset of $\Xi_i^U$, which indicates initial challenges in start-bounded games. The initial state satisfies $(s_0')$. In each round $i > 0$, if $x_{i-1} \in \text{ind}(K_1)$, then player 2 launches the start-bounded game $G_S^2(G_2, G_1)$ with the initial state $(\emptyset, \Xi_i \mapsto x_i)$. Otherwise, if $x_{i-1} \not\in \text{ind}(K_1)$, player 2 has two options. First, he can challenge player 1 with the set $\Psi_{i-1}$ (that is, similar to the backward game but with a possibly smaller $\Psi_{i-1}$ in place of $\Xi_{i-1}$); player 1 responds to this challenge with a state $(\Xi_i \mapsto x_i, \Psi_i)$ such that $\Psi_i \not\subseteq \Xi_i$, $x_i \sim^{G_1} x_{i-1}$ and $(s_i')$ holds. Second, player 2 can launch the start-bounded game $G_S^2(G_2, G_1)$ with the initial state $(\emptyset, \Xi_1 \mapsto x_1)$, where the first challenge of player 2 must be picked from $\Phi_{i-1} = \Xi_{i-1} \setminus \Psi_{i-1}$.

Example 21 We illustrate the $\omega$-winning strategy for player 1 in $G_S^2(G_2, G_1)$ starting from $(\{u_1\} \mapsto x_2, \{u_2\})$, where $G_S^2$ is from Example 20 and $G_2^2$ looks like $M_1^2$ from Example 20 (but with $x_i$ in place of $\sigma_i$):

- $(u_1 \mapsto \sigma_1) \Rightarrow (u_2 \mapsto \sigma_2) \Rightarrow (u_3, u_10) \\
  u_2 \sim^{G_2} u_6, u_7 \sim^{G_2} u_4 \Rightarrow (\emptyset, \{u_2, u_9\} \mapsto x_1) \Rightarrow (\emptyset, \{u_3, u_10\} \mapsto a)$
Lemma 22 For any \( u_0 \in \Delta^{\varphi_2} \), condition \((\varphi)\) holds for arbitrary strategies in \( G_2^{\varphi_2}(G_2, M_1) \) if \((\omega)\) holds in \( G_2^{\varphi_2}(2, G_1) \) for some state \((\Xi_0 \rightarrow x_0, \Psi_0)\) with \( u_0 \in \Xi_0 \).

Condition \((\omega)\) in the lemma above is checked in time \( O(|\text{ind}(K_2)| \times 2^{2^{|A^2|}} |\text{ind}(G_2)|) \), which can be readily seen by analysing the full game graph for \( G_2^{\varphi_2}(2, G_1) \) (similar to that in Example 21). By Theorem 6, we then obtain:

Theorem 23 For combined complexity, \( \Sigma \)-query entailment is in \( 2\text{EXPTIME} \) for Horn-\( \text{ALCHI} \) and Horn-\( \text{ALCNI} \) KBs, and in \( \text{EXPTIME} \) for DL-Lite\(_{\text{core}}^N\) and DL-Lite\(_{\text{core}}^H\) KBs. For data complexity, these problems are all in \( P \).

Lower Bounds

We have shown that, for all of our DLs, \( \Sigma \)-query entailment and inseparability are in \( P \) for data complexity. The next theorem establishes a matching lower bound:

Theorem 24 For data complexity, \( \Sigma \)-query entailment and inseparability are \( P \)-hard for DL-Lite\(_{\text{core}}^H\) and \( \mathcal{EL} \) KBs.

Proof. The proof is by reduction of the \( P \)-complete entailment problem for \( \text{acyclic} \) Horn ternary clauses: given a conjunction \( \varphi \) of clauses of the form \( a_i \land a_i \lor a_i \rightarrow a_j \), \( i, i', j < j \), decide whether \( a_n \) is true in every model of \( \varphi \). Consider the \( \mathcal{EL} \) TBox \( T = \{ V \in \exists P. (\exists R_1. V \land \exists R_2. V) \} \) and an ABox \( A \) comprised of \( F(a_n) \) and \( P(a_i, a_i, R_1(a_i, a_i), R_2(a_i, a_i)) \), for each clause \( a_i \varphi \) of \( \varphi \).

Set \( \Sigma = \{ T, P, R_1, R_2 \} \), \( K_2 = \{ \{ T, A \cup \{ V(a_n) \} \} \} \) and \( K_1 = \{ \emptyset, A \} \). Obviously, \( K_2 \)-\( \Sigma \)-query entails \( K_1 \). On the other hand, the materialisation of \( K_2 \) is (finitely) \( \Sigma \)-homomorphically embeddable in the materialisation of \( K_1 \) iff \( \varphi \) derives \( a_n \) (see the full version for details). For DL-Lite\(_{\text{core}}^H\), we take \( T \) to contain \( V \subseteq \exists P, \exists P' \subseteq \exists R_i \) and \( \exists R_i \subseteq \forall V \), for \( i = 1, 2 \).

For combined complexity, \( \text{EXPTIME} \)-hardness of \( \Sigma \)-query inseparability for Horn-\( \text{ALCNI} \) can be proved by reduction of the subsumption problem: we have \( T \models \models A \subseteq B \iff \langle T \cup \{ A(a) \} \rangle \) and \( \langle T \cup \{ A(a) \} \rangle \) are \( \{ B \} \)-query inseparable. We now establish matching lower bounds in the technically challenging cases.

Theorem 25 For combined complexity, \( \Sigma \)-query entailment and inseparability are (i) \( 2\text{EXPTIME} \)-hard for Horn-\( \text{ALCHI} \) KBs and (ii) \( \text{EXPTIME} \)-hard for DL-Lite\(_{\text{core}}^H\) KBs.

Proof. The proof of (i) is by encoding alternating Turing machines (ATMs) with exponential tape and using the fact that \( \text{AEXPTIME} = \text{2EXPTIME} \); see, e.g., (Kozen 2006).

Let \( M = (\Gamma, Q, q_0, \varphi, \delta) \) be an ATM with a tape alphabet \( \Gamma \), a set of states \( Q \) partitioned into existential \( Q_\exists \) and universal \( Q_\forall \) states, an initial state \( q_0 \in Q_\exists \), an accepting state \( q_1 \in Q \), and a transition function \( \delta: (Q \setminus \{ q_1 \}) \times \Gamma \times \{ 1, 2 \} \rightarrow Q \times \Gamma \times \{ -1, 0, +1 \} \), which, for a state \( q \) and symbol \( a \), gives two instructions, \( \delta(q, a, 1) \) and \( \delta(q, a, 2) \). We assume that existential and universal states strictly alternate: any transition from an existential state results in a universal state, and vice versa. We extend \( \delta \) with the instructions \( \delta(q_1, a, k) = (q_1, a, 0), \) for \( a \in \Gamma \) and \( k = 1, 2 \), which go into an infinite loop if \( M \) reaches the accepting state \( q_1 \). Thus, assuming that \( M \) terminates on every input, it accepts \( \overline{w} \) iff the modified ATM \( M' \) has a run on \( \overline{w} \), all branches of which are infinite.

Our aim is to construct, given \( M \) and \( \overline{w} \), TBoxes \( T_1 \) and \( T_2 \) and a signature \( \Sigma \) such that \( M' \) has a run with only infinite branches iff the materialisation \( M_2 \) of \( (T_2, \{ A(c) \}) \) is finitely \( \Sigma \)-homomorphically embeddable into the materialisation \( M_1 \) of \( (T_1, \{ A(c) \}) \). Let \( f \) be a polynomial such that, on any input of length \( n \), \( M \) uses at most \( 2^n - 2 \) tape cells, with \( n = f(m) \), which are numbered from 1 to \( 2^n - 2 \), and the head stays to the right of cell 0, which contains the marker \( > \in \Gamma \). The construction proceeds in five steps.

Step 0. We use tuples of \( 2n \) concepts to represent distances of up to \( 2^n \) between the cells on the tape in consecutive configurations. We refer to a tuple \( Y_{n-1}, Y_{n-1}, \ldots, Y_0, Y_0 \) of concept names as \( Y \) and assume that the TBox contains the following CIs to encode an \( n \)-bit \( R \)-counter on \( Y \):

\[
Y_k \sqcap Y_{k-1} \sqcap \cdots \sqcap Y_0 \sqsubseteq \forall R, (Y_k \sqcap Y_{k-1} \sqcap \cdots \sqcap Y_0),
\]

\[
n > k \geq 0,
\]

\[
Y_i \sqcap Y_k \sqsubseteq \forall R, Y_i \text{ and } Y_i \sqcap Y_k \sqsubseteq \forall R, Y_i, n > i > k.
\]

We use the expression \( i < 2^n - 1 \) on the left-hand side of CIs to say that the \( Y \)-value is \( 2^n - 1 \) (which is a shortcut for \( Y_{n-1} \sqcap \cdots \sqcap Y_0 \)); we also use \( i < 2^n - 1 \) on the left-hand side of CIs for the complementary statement (which is a shortcut for \( n \)-CIs with \( i > 2^n - 1 \) replaced by each of \( Y_{n-1}, \ldots, Y_0 \)). Finally, we use set\(_0\) on the right-hand side of CIs for the reset command (which is equivalent to \( Y_{n-1} \sqcap \cdots \sqcap Y_0 \)). Note that the counter stops at \( 2^n - 1 \): the \( R \)-successors of a domain element in \( Y_{2^n-1} \) do not have to encode any value.

Step 1. First we encode configurations and transitions of \( M' \) using \( T_1 \). We represent a configuration by a \( \text{block} \), which is a sequence of \( 2^n + 1 \) domain elements connected by a role \( P \). The first element distinguishes the blocks for the two alternative transitions; using a \( P \)-counter on a tuple \( T \), we assign indices from 0 to \( 2^n - 1 \) to all other elements in each block. The element with index 0 is needed for padding. Each of the remaining \( 2^n - 1 \) elements belongs to a concept \( C_0 \), for some \( a \in \Gamma \); if the element with index \( i + 1 \) is in \( C_0 \), then the cell \( i \) is assumed to contain \( a \) in the configuration represented by the block (in particular, the element with index 1 contains \( b \) for cell 0) as shown below:

\[
M_1 = \begin{array}{cccccccc}
A & C_0 & C_1 & C_2 & C_3 & C_4 & C_5 & C_6
\end{array}
\]

The first block represents the initial configuration: the input \( \overline{w} = a_1 \cdots a_m \) is followed by \( 2^n - m - 2 \) blank symbols \( \omega \) and the head is positioned over cell 1, which is indicated by the 0 value of the \( P \)-counter on a tuple \( H \). This is achieved by the following CIs in the TBox \( T_1 \):

\[
A \sqsubseteq \exists P_1. (\text{set}_0^H \sqcap \exists P_1. (C_0 \sqcap \exists P_1. (C_1 \sqcap \text{set}_0^H \sqcap C_2))) \sqcap \exists P_1. (C_2 \sqcap \exists P_1. (\ldots \exists P_1. (C_m \sqcap I \ldots) \ldots)), \quad (T_1-1)
\]

\[
\text{if}_T < 2^n - 1 \sqcap I \sqsubseteq \exists P_1. (I \sqcap C_0), \quad (T_1-2)
\]

\[
\text{if}_T = 2^n - 1 \sqcap I \sqsubseteq Z_{0+1}, \quad (T_1-3)
\]
Step 2. The contents of the tape and the head position in each configuration is encoded in a block of length $2^n + 1$: the current state $q \in Q$ is recorded in the concept $Z_q^0$ that contains the last element of the block (if $q \in \Gamma$ specifies the contents of the active cell scanned by the head). At the end of the step, when the $T$-value reaches $2^n - 1$, we branch out one block for each of the two transitions, reset the $P$-counter on $T$, and propagate via $Z_q^1$ and $Z_q^2$ the current state and symbol in the active cell: for $q \in Q$ and $a \in \Gamma$, we add to $T_1$ the CI

$$\text{if } T = 2^n - 1 \text{ then } Z_q^0 \subseteq \bigcup_{k=1,2} \exists P. (X_k \cap \exists P. (T_q \cap Z_q^k)). \quad (T_1-4)$$

where $X_1$ and $X_2$ are two fresh concept names.

The acceptance condition for $M'$ is enforced by means of $T_2$, which uses a $P$-counter on a tuple $T_0$ for a block representing the initial configuration (a $T_0$-block):

$$A \subseteq \exists P. T_0, \quad \text{if } T_0 = 0 \text{ then } \exists P\ . \quad (T_2-1)$$

Two $P$-counters, on $T_1$ and $T_2$, are used for blocks representing configurations with universal states ($T_1$- and $T_2$-blocks respectively) and one $P$-counter, on a tuple $T_3$, suffices for blocks representing configurations with existential states ($T_3$-blocks). These blocks are arranged into an infinite tree-like structure: the $T_0$-block is the root, from which a $T_1$- and a $T_2$-blocks branch out (successors of the initial state $q_0$ are universal). Each of them is followed by a $T_3$-block, which branches out a $T_1$- and a $T_2$-blocks, and so on. This is achieved by adding to $T_2$ the following CIs:

$$\text{if } T_1 = 2^n - 1 \text{ then } \bigcup_{j=1,2} \exists P. (X_j \cap \exists P. T_0), \text{ for } k = 0, 3. \quad (T_2-3)$$

$$\text{if } T_2 = 2^n - 1 \text{ then } \exists P. G, \text{ for } k = 1, 2, 3. \quad (T_2-4)$$

$$\text{if } T_3 = 2^n - 1 \text{ then } \exists P. \exists P. T_0, \text{ for } k = 1, 2. \quad (T_2-5)$$

where $G$ is a concept name. If $\Sigma = \{A, X_1, X_2, P\}$ then there is a unique $\Sigma$-homomorphism from the $T_0$-block in $M_2$ to the block of the initial configuration in $M_1$. Next, concepts $X_1$ and $X_2$ ensure that the $T_1$- and $T_2$-blocks are $\Sigma$-homomorphically mapped (in a unique way) into the respective blocks in $M_1$, which reflects the acceptance condition of universal states. The following $T_3$-block, however, contains neither $X_1$ nor $X_2$ and can be mapped to either of the blocks in $M_1$, which reflects the choice in existential states; see the picture below, where possible $\Sigma$-homomorphisms are shown by thick dashed arrows:

Step 3. Recall that the $P$-counter on $H$ measures the distance from the head: if the active cell in the current configuration is $k$, then its $H$-value is 0 and the $H$-value of the cell $k-1$ in a successor configuration is $2^n-1$. So, until the $H$-counter reaches $2^n-1$, the following CIs in $T_1$ propagate the state and symbol in the active cell along the blocks: for $q \in Q$, $a \in \Gamma$ and $k = 0, 1, 2$,

$$\text{if } T < 2^n - 1 \text{ then } \exists P. (C_b \cap Z_q^b) \quad (T_1-5)$$

(for each $b \in \Gamma$, these CIs generate a branch in $M_1$ to represent the same cell but with a different symbol, $b$, tentatively assigned to the cell—Step 4 will ensure that the correct branch and symbol are selected to match the cell contents in the preceding configuration). When the distance from the last head position is $2^n$, the contents of the cell and the current state are changed according to $\delta$:

$$\text{if } T < 2^n - 1 \text{ then } \exists P. (C_b \cap Z_q^b) \subseteq \bigcup_{b \in \Gamma} \exists P. (C_b \cap \Delta_{q_{a,b}}^k). \quad (T_1-6)$$

(there is a unique $\Sigma$-homomorphism from the $T_0$-block in $M_2$ to the block of the initial configuration in $M_1$. Next, concepts $X_1$ and $X_2$ ensure that the $T_1$- and $T_2$-blocks are $\Sigma$-homomorphically mapped (in a unique way) into the respective blocks in $M_1$, which reflects the acceptance condition of universal states. The following $T_3$-block, however, contains neither $X_1$ nor $X_2$ and can be mapped to either of the blocks in $M_1$, which reflects the choice in existential states; see the picture below, where possible $\Sigma$-homomorphisms are shown by thick dashed arrows:

Step 4. The CIs $(T_1-5)-(T_1-6)$ generate a separate $P$-successor for each $b \in \Gamma$. The correct one is chosen by a

(Note that there is only one branch for the modified cell, which corresponds to the new symbol, $a'$, in that cell; see explanations below.) Then, the current state and the symbol in the active cell are propagated along the tape using $(T_1-5)$. The correct one is chosen by a
finite \( \Sigma \)-homomorphism, \( h \), from \( M_2 \) to \( M_1 \). To exclude wrong choices, we take
\[
\Sigma = \{ A, P, X_1, X_2 \} \cup \{ D_a \mid a \in \Gamma \}.
\]
Recall that if \( d_1 \in C^{M_1}_a \), for some \( a \in \Gamma \), then it represents a cell containing \( a \). The following CIs in \( \mathcal{T}_1 \) ensure that, for each \( b \in \Gamma \) different from \( a \), there is a block of \((2^n + 1)\)-many \( P \)-connected elements that ends in the concept \( D_b \) (called a \( D_b \)-block in the sequel):
\[
C_a \subseteq D_a \cap \bigcap_{b \in \Gamma \setminus \{ a \}} G_b, \quad (T_1.7)
\]
\[
G_b \subseteq \exists P^+ (S_b \cap \text{set}_b), \quad (T-1)
\]
\[
\text{if } B_{2^n - 1} \cap S_b \subseteq \exists P^-. S_b, \quad (T-2)
\]
\[
\text{if } B_{2^n - 1} \cap S_b \subseteq \exists P-. D_b, \quad (T-3)
\]
where we use a \( P^- \)-counter on a tuple \( B \) (unlike \( P^- \)-counters in all other cases) and a concept \( S_b \) to propagate \( b \) along the whole block. Suppose \( h(d_2) = d_1 \) and \( d_2 \) belongs to \( G \) in \( M_2 \) (it represents a cell in a non-initial configuration). Then the following CI and \((T-1)-(T-3)\), added to \( \mathcal{T}_2 \), generate a \( D_b \)-block, for each \( b \in \Gamma \) (including \( a \)):}
\[
G \subseteq \bigcap_{b \in \Gamma} G_b. \quad (T_2.6)
\]
Each of the \( D_b \)-blocks in \( M_2 \), for \( b \in \Gamma \) with \( b \neq a \), can be mapped by \( h \) to the respective \( D_b \)-block in \( M_1 \). By the choice of \( \Sigma \), the only remaining \( D_a \)-block, in case \( a \) is tentatively contained in this cell, could be mapped (in the reverse order) along the branch in \( M_1 \) but only if the cell contains \( a \) in the preceding configuration (that is, the element which is \( 2^n + 1 \) steps closer to the root of \( M_1 \) belongs to \( D_a \):

![Diagram](image)

Note (see \( \Delta_{\varphi, \psi}^{k} \)) that the cell whose content is changed generates the additional \( D_a \)-block in \( M_1 \) to allow the respective \( D_a \)-block from \( M_2 \) to be mapped there.

One can show that \( M' \) has a run with only infinite branches iff \((\mathcal{T}_1, \{ A'(c) \}) \Sigma \)-query entails \((\mathcal{T}_2, \{ A'(c) \}) \). It follows, by Theorem 2, that deciding \( \Sigma \)-query inseparability is \( 2\text{ExpTime} \)-hard.

(ii) A proof of \( \text{ExpTime} \)-hardness of \( \Sigma \)-query inseparability for \( \text{DL-Lite}^H \) KBs is given in the full paper. It uses the same idea of encoding computations of ATMs. One essential difference is that the expressive power of \( \text{DL-Lite}^H \) is not enough to represent \( n \)-bit counters in Step 0, and so we can only encode computations on polynomial tape.

As a consequence of Theorems 3, 23 and 25 we obtain:

**Theorem 26** For combined complexity, the membership problem for universal UCQ-solutions is \( 2\text{ExpTime-complete} \) for Horn-\( \text{ALCHI} \) and Horn-\( \text{ACCT} \); \( \text{ExpTime-complete} \) for Horn-\( \text{ACCH} \), Horn-\( \text{ALC} \), \( \text{DL-Lite}^H \) and \( \text{DL-Lite}^H_{\text{core}} \); and \( P \)-complete for \( \mathcal{EL} \) and \( \mathcal{ELH} \). For data complexity, all these problems are \( P \)-complete.

In the case of \( \text{DL-Lite}^H_{\text{core}} \) we also obtain an \( \text{ExpTime} \) algorithm for checking the existence and computing universal UCQ-solutions. Indeed, given a KB \( K \), a target signature \( \Sigma_2 \) and a mapping \( \mathcal{T}_2 \), we first compute the \( \Sigma_2 \)-ABox over \( \text{ind}(K_1) \) that is implied by \( K_1 \) and \( \mathcal{T}_1 \), and then check whether at least one KB \( K_2 \) in \( \Sigma_2 \) with this ABox is a universal UCQ-solution (there are \( \leq O(2^{2^{2^n}}) \) such KBs). This gives an \( \text{ExpTime} \) upper bound for the non-emptiness problem for universal UCQ-solutions in \( \text{DL-Lite}^H_{\text{core}} \) (Areias et al. 2013). Similarly, we can check in \( \text{ExpTime} \) whether the result of forgetting a signature in a \( \text{DL-Lite}^H_{\text{core}} \) KB exists.

\( \Sigma \)-query inseparability of \( \text{DL-Lite}^H_{\text{core}} \) TBoxes was known to sit between \( \text{PSPACE} \) and \( \text{ExpTime} \) (Konev et al. 2011). Using the fact that witness ABoxes for \( \text{DL-Lite}^H_{\text{core}} \) Tbox separability can always be chosen among the singleton ABoxes (Konev et al. 2011, Theorem 8), we can modify the proof of Theorem 25 to improve the \( \text{PSPACE} \) lower bound:

**Theorem 27** \( \Sigma \)-query inseparability of \( \text{DL-Lite}^H_{\text{core}} \) TBoxes is \( \text{ExpTime-complete} \).

For more expressive DLs, TBox \( \Sigma \)-query inseparability is often harder than KB inseparability: for \( \text{DL-Lite}^H \), the space of relevant witness ABoxes for TBox separability is of exponential size and, in fact, TBox inseparability is \( \text{NP-hard} \), while KB inseparability is in \( \text{P} \). Similarly, \( \Sigma \)-query inseparability of \( \mathcal{EL} \) KBs is tractable, while \( \Sigma \)-query inseparability of TBoxes is \( \text{ExpTime-complete} \) (Lutz and Wolter 2010). The complexity of TBox inseparability for Horn-DLs extending \( \text{Horn-ALC} \) is not known.

**Future Work**

From a theoretical point of view, it would be of interest to investigate the complexity of \( \Sigma \)-query inseparability for KBs in more expressive Horn DLs (e.g. \( \text{Horn-SHIQ} \)) and non-Horn DLs extending \( \text{ALC} \). We conjecture that the game technique developed in this paper can be extended to those DLs as well. Our games can also be used to define \textit{efficient approximations} of \( \Sigma \)-query entailment and inseparability for KBs. The existence of a forward strategy, for example, provides a sufficient condition for \( \Sigma \)-query entailment for all of our DLs. Thus, one can extract a \( \Sigma \)-query module of a given KB \( K \) by exhaustively removing from \( K \) those inclusions and assertions \( a \) for which player 1 has a winning strategy in the game \( G_1^\Sigma (G_1, G_2) \), where \( G_1 \) is a generating structure for \( K \) \( \setminus \{ a \} \) and \( G_2 \). The resulting modules are minimal for our DLs without inverse roles, and we conjecture that in practice they are often minimal for DLs with inverse roles as well; see (Konev et al. 2011) for experiments testing similar ideas for module extraction from TBoxes.

Finally, we plan to use the developed technique to investigate the complexity of the non-emptiness problem for universal UCQ-solutions in data exchange as well as algorithms for computing universal UCQ-solutions in various DLs.
References


Appendix

The proofs of the theorems and lemmas from the paper are presented in the following order. We begin by proving Theorem 6, where we show how to construct finite generating structures for each of the languages and check that these structures give rise to materialisations. Theorem 6 is then used to prove Theorem 5, which characterises $\Sigma$-query entailment through finite $\Sigma$-homomorphisms. After that we prove Theorems 2 and 3 that establish a connection between query entailment, query inseparability and universal UCQ-solutions.

Next, we prove the results for our games: first, Theorem 8 relating finite $\Sigma$-homomorphisms with infinite games, and second, Lemma 22 saying that arbitrary games cover and admit arbitrary strategies. Proofs of Lemmas 11, 13 and 19 are obtained as corollaries of the proof of Lemma 22. Finally, we prove Lemma 19, establishing $\text{coNP}$-hardness of backward strategies.

We conclude with the proofs of lower bounds. First, we prove Theorem 24. Then, in the proof of Theorem 25 we show $\text{ExpTime}$-hardness of $\Sigma$-query entailment in DL-Lite$_{\text{core}}^H$.

Proof of Theorem 6: construction of generating structures

Theorem 6 Horn-ALCHI and all of its fragments defined above have finitely generated materialisations. Moreover,

- for any $L\in\{\text{ALCHI, ALCT, ALCH, ALC}\}$ and any Horn-L KB $(T, A)$, a generating structure can be constructed in time $|A| \cdot 2^{|T|}$, $p$ a polynomial;
- for any $L$ in the $\text{EL}$ and DL-Lite families and any KB $(T, A)$, a generating structure can be constructed in time $|A| \cdot p(|T|)$, $p$ a polynomial.

We construct the generating structures first for Horn-ALCHI, then for $\text{ELCH}$, and finally for DL-Lite$^H_{\text{core}}$. The construction for Horn-ACCT, Horn-ACCH, and Horn-ACC is the same as for Horn-ALCHI, the construction for $\text{EL}$ is the same as for $\text{ELCH}$, and the construction for DL-Lite$^H_{\text{core}}$, DL-Lite$^H_{\text{core}}$, and DL-Lite$_{\text{horn}}$ is the same as for DL-Lite$^H_{\text{horn}}$ and therefore omitted.

Horn-ALCHI

To construct the generating structure for Horn-ALCHI TBoxes we first transform the TBox into normal form (Krötzsch, Rudolph, and Hitzler 2007). A Horn-ALCHI TBox is in normal form if its concept inclusions are of the following form

$$A \sqsubseteq B, \quad A \sqsubseteq A_1 \sqcap A_2 \sqsubseteq B,$$
$$A \sqsubseteq \bot, \quad T \sqsubseteq B,$$
$$A \sqsubseteq \exists R.B, \quad A \sqsubseteq \forall R.B,$$
$$\exists R.A \sqsubseteq B, \quad R_1 \sqsubseteq R_2,$$

where $A, A_1, A_2, B$ are concept names and $R, R_1, R_2$ are roles. The following result is well-known (Krötzsch, Rudolph, and Hitzler 2007; Eiter et al. 2008):

Theorem A.28 For every Horn-ALCHI TBox $T$ one can construct in polynomial time a Horn-ALCHI TBox $T'$ in normal form such that $T$ and $T'$ are $\Sigma$-query inseparable for the signature $\Sigma$ of $T$. Moreover,

- if $T$ does not contain any role inclusions then $T'$ does not contain any role inclusions;
- if $T$ does not contain any inverse roles then $T'$ does not contain any inverse roles.

Now assume a consistent KB $\mathcal{K} = (T, A)$ with a Horn-ALCHI TBox $T$ in normal form is given. By sub($T$) we denote the set of subconcepts of concepts in $T$. The $T$-type of $d$ in $I$ is defined by taking

$$h^T_I(d) = \{ c \in \text{sub}(T) \mid d \in C^T_I \}.$$

We say that $t$ is a $T$-type if there exists a model $I$ of $T$ such that $t = h^T_I(d)$, for some $d \in \Delta^T_I$. Denote by type($T$) the set of all $T$-types. It is well known that type($T$) can be computed in exponential time in $|T|$. We now construct the finitely generating structure $G = (\Delta^G, \mathcal{G}, \rightarrow)$ for $\mathcal{K}$, where $\Delta^G = \text{ind}(\mathcal{K}) \cup \mathcal{G}$.

For any role $R$ in $T$, we set

$$[R] = \{ S \mid T \models R \subseteq S, T \models S \subseteq R \}.$$

We write $[R] \leq_T [S]$ if $T \models R \subseteq S$; thus, $\leq_T$ is a partial order on the set $\{ [R] \mid R$ a role in $T \}$. $\Omega$ will be a subset of the set of pairs $([R], t)$ with $t \in \text{type}(T)$. First define $\rightarrow$ as follows:

- $a \rightsquigarrow ([R], t)$ if $a \in \text{ind}(\mathcal{K})$ and $t$ is a maximal (with respect to set-inclusion) $T$-type such that $\mathcal{K} \models \exists R. (\bigwedge_{C \subseteq \mathcal{G}} C)(a)$ and $\mathcal{K} \not\models R(a, b)$ for any $b \in \text{ind}(\mathcal{K})$ with $t \subseteq \{ C \in \text{sub}(T) \mid C \models C(b) \}$;
- $([R_1], t_1) \rightsquigarrow ([R_2], t_2)$ if $t_2$ is a maximal $T$-type such that $\mathcal{K} \models (\bigwedge_{C \subseteq \mathcal{G}} C) \subseteq \exists R_2. (\bigwedge_{C \subseteq \mathcal{G}} C)$.

$\Omega$ is defined as the set of all pairs $([R], t)$ such that there are $a \in \text{ind}(\mathcal{K}), R_1, \ldots, R_n = R$ and $t_1, \ldots, t_n = t$ such that

$$a \rightsquigarrow ([R_1], t_1) \rightarrow \cdots \rightarrow ([R_n], t_n).$$

We define the interpretation function $\mathcal{G}$ by setting

$$A^\mathcal{G} = \{ a \in \text{ind}(\mathcal{K}) \mid \mathcal{K} \models A(a) \} \cup \{ ([R], t) \in \Omega \mid A \in t \},$$

$$P^\mathcal{G} = \{ (a, b) \mid \text{there is } R(a, b) \in A \text{ with } T \models R \subseteq P \},$$

and for every edge $u \rightsquigarrow v$ with $v = (R, t)$, we set

$$(u, v)^\mathcal{G} = \{ S \mid [R] \leq_T [S] \}.$$
The proof above also shows that \( T \)-types of a node in \( \mathcal{M} \) coincide with the \( T \)-types used in the construction of the node.

**Lemma A.30** Let \( \mathcal{M} \) be the unravelling of \( \mathcal{G} \). Then

1. for all \( a \in \text{ind}(\mathcal{K}) \), \( h^\mathcal{M}(a) = \{C \in \text{sub}(T) \mid \mathcal{K} \models C(a)\} \);
2. for all \( \sigma \cdot ([R], t) \in \Delta^\mathcal{M} \), \( h^\mathcal{M}(\sigma \cdot ([R], t)) = t \).

**Proposition A.31** If \( \mathcal{I} \) is a model of \( \mathcal{K} \), then there exists a homomorphism from \( \mathcal{M} \) to \( \mathcal{I} \).

**Proof.** We define a function \( h : \Delta^\mathcal{M} \to \Delta^\mathcal{I} \) for each \( \sigma \in \Delta^\mathcal{M} \) by induction on the length of \( \sigma \), and simultaneously show it is a homomorphism, i.e.,

\[(1) \quad h(a^\mathcal{M}) = a^\mathcal{I} \quad \text{for} \quad a \in \text{ind}(\mathcal{K}), \]

\[(2) \quad h^\mathcal{M}(\sigma) \subseteq h^\mathcal{I}(h(\sigma)) \quad \text{for} \quad \sigma \in \Delta^\mathcal{M}, \]

\[(3) \quad r^\mathcal{M}(\sigma, \sigma') \subseteq r^\mathcal{I}(h(\sigma), h(\sigma')) \quad \text{for} \quad \sigma, \sigma' \in \Delta^\mathcal{M}. \]

First, for each \( a \in \text{ind}(\mathcal{K}) \), we set \( h(a^\mathcal{M}) = a^\mathcal{I} \). This ensures (1). Conditions 2 and 3 follow for \( \sigma, \sigma' \in \text{ind}(\mathcal{K}) \) from Lemma A.30, the condition that \( \mathcal{I} \) is a model of \( \mathcal{K} \), and the construction of \( \mathcal{M} \).

Let \( \sigma \cdot ([S], t) \in \Delta^\mathcal{M} \) such that \( h(\sigma) \) is defined. By construction of \( \mathcal{M} \), it follows \( \mathcal{K} \models \exists S, (\bigcap_{C \in \mathcal{C}} C)(a) \) if \( a = a' \) or \( \mathcal{I} \models ((\bigcap_{C \in \mathcal{C}} C) \subseteq \exists S, (\bigcap_{C \in \mathcal{C}} C)) \) if \( \text{tail}(\sigma) = ((Q), t') \). By the condition that \( \mathcal{I} \) is a model of \( \mathcal{K} \), by Lemma A.30, and by the induction hypothesis \( h^\mathcal{M}(\sigma) \subseteq h^\mathcal{I}(h(\sigma)) \), it follows that there exists \( z \in \Delta^\mathcal{I} \) such that \( S \in r^\mathcal{I}(h(\sigma), z) \) and \( t \subseteq h^\mathcal{I}(z) \). We set \( h(\sigma \cdot ([S], t)) = z \) and show that (2) and (3) hold. (2) follows immediately from the fact that \( h^\mathcal{M}(\sigma \cdot ([S], t)) = t \) (by Lemma A.30). (3) follows from \( R \in r^\mathcal{M}(\sigma, \sigma \cdot ([S], t)) \) it follows \( T \models S \subseteq R \), and since \( \mathcal{I} \) is a model of \( T \), we get \( R \in r^\mathcal{I}(h(\sigma), z) \).

**\( \mathcal{ELH} \)**

We now construct generating structures for \( \mathcal{ELH} \). Again we first transform the TBox into normal form (Baader, Brandt, and Lutz 2005). An \( \mathcal{ELH} \) TBox is in normal form if its concept inclusions are of the following form:

\[
\begin{align*}
A &\subseteq B, & A_1 \sqcap A_2 &\subseteq B, \\
\top &\subseteq B, & A &\subseteq \exists P B, \\
\exists P A &\subseteq B, & P_1 &\subseteq P_2,
\end{align*}
\]

where \( A, A_1, A_2, B \) are concept names and \( P, P_1, P_2 \) are role names. The following result is well known (Baader, Brandt, and Lutz 2005):

**Theorem A.32** For every \( \mathcal{ELH} \) TBox \( T \) one can construct in polynomial time an \( \mathcal{ELH} \) TBox \( T' \) in normal form such that \( T \) and \( T' \) are \( \Sigma \)-query inseparable for the signature \( \Sigma \) of \( T \).

Assume \( \mathcal{K} = (T, A) \) with \( T \) an \( \mathcal{ELH} \) TBox in normal form is given. We construct the generating structure \( \mathcal{G} = (\Delta^\mathcal{G}, \neg, \sqsupseteq) \) for \( \mathcal{K} \) as follows, where \( \Delta^\mathcal{G} = \text{ind}(\mathcal{K}) \cup \Omega \) and \( \Omega \) is a subset of the set of pairs \( \{P, A\} \), for \( A \) and \( P \), concept and role names in \( T \), respectively. (The class \( \{P\} \) is
defined in the construction for Horn-ALCHI.) Define $\models$ as follows:

- $\alpha \models ([P], A)$ if $\alpha \in \text{ind}(K)$ and $A$ is a concept name in $T$ such that $K \models \exists P.A$ and $K \not\models P(a, b)$ for any $b \in \text{ind}(K)$ with $K \models (A(b))$;
- $([P_1], A_1) \models ([P_2], A_2)$ if $T \models A_1 \subseteq \exists P_2.A_2$.

$\Omega$ is defined as the set of all pairs $([P], A)$ such that there are $\alpha \in \text{ind}(K)$, $P_1, \ldots, P_n = P$ and $A_1, \ldots, A_n = A$ such that

$$\alpha \models ([P_1], A_1) \cdots \models ([P_n], A_n).$$

We define the interpretation function $\mathcal{G}$ by setting

$$A^K = \{ \alpha \in \text{ind}(K) \mid K \models A(\alpha) \} \cup \{ ([P], B) \in \Omega \mid T \models B \subseteq A \},$$

$$P^K = \{ (a, b) \mid \text{there is } P'(a, b) \in A \text{ with } T \models P' \subseteq P \}$$

and for every edge $u \rightsquigarrow v$ with $v = ([P], A)$, we set

$$(u, v)^G = \{ P' \mid [P] \subseteq T [P'] \}.$$

It can be shown that $G$ is a generating structure for $K$. Let $\mathcal{M}$ be the interpretation defined by unravelling $G$.

**Proposition A.33** $\mathcal{M}$ is a model of $K$.

**Proof.** Clearly, $\mathcal{M}$ is a model of $\mathcal{A}$. We show $\mathcal{M} \models T$ by verifying $\mathcal{M} \models \alpha$ for each $\alpha \in T$.

$\alpha = A \sqsubseteq B$. Here we consider $A$ to be either a concept name $A$, or $T$. Let $x \in A^\mathcal{M}$. If $x = a \in \text{ind}(K)$, then $K \models A(a)$ by construction of $A^\mathcal{M}$. Since $A \sqsubseteq B \in T$, it follows $K \models B(a)$, hence $a \in A^\mathcal{M}$. If $x = x' \cdot ([R], C)$, then $T \models C \subseteq A$. Because of $\alpha$, we have that $T \models C \subseteq B$, therefore $x \in B^\mathcal{M}$.

$\alpha = A_1 \sqcap A_2 \sqsubseteq B$. The argument is analogous for $x \in A_1^\mathcal{M} \cap A_2^\mathcal{M}$.

$\alpha = \exists P.B$. Let $x \in A^\mathcal{M}$. If $x = a \in \text{ind}(K)$, then $K \models A(a)$ by construction of $A^\mathcal{M}$. Since $A \sqsubseteq B \in T$, it follows $K \models \exists P.B(a)$. Assume $K \models \{ P(a, b), B(b) \}$ for some $b \in \text{ind}(K)$, then $(a, b) \in P^K$ and $b \in B^\mathcal{M}$, so $\mathcal{M} \models \alpha$. Otherwise, we have that $\alpha \models ([P], B)$. From the construction of $\mathcal{M}$, we have that $(a, b \cdot ([P], B)) \in P^K$ and $([P], B) \in B^\mathcal{M}$. For the case tail$(x) = ([R], t)$, the proof is similar.

$\alpha = \exists P.A \sqsubseteq B$. Assume $(x, y) \in P^K$ and $y \in A^\mathcal{M}$. We consider various cases of $x$ and $y$:

- $x = b, y = a$ for $a, b \in \text{ind}(K)$. Then $K \models P(b, a)$ and $K \models A(a)$, consequently $K \models B(b)$, so $b \in B^\mathcal{M}$ by construction of $\mathcal{M}$.
- $x = b \in \text{ind}(K), y = b \cdot ([S], C)$. Then by construction of $\mathcal{M}$, $K \models \exists S.C(b)$, moreover $T \models \{ S \subseteq P, C \subseteq A \}$. Next, $K \models \exists P.A(b)$, and finally $K \models B(b)$, so $b \in B^\mathcal{M}$.
- tail$(x) = ([Q], D), y = x \cdot ([S], C)$. Then by construction of $\mathcal{M}$, $T \models D \subseteq S.C$, moreover $T \models \{ S \subseteq P, C \subseteq A \}$. It follows $T \models D \subseteq \exists P.A$, and because of $\alpha$, $T \models D \subseteq B$. Hence, by construction of $B^\mathcal{M}$, $x \in B^\mathcal{M}$.

Observe that the case $x = y \cdot ([S], C)$ for some $S$ and $C$ is not possible as it would required that $T \models S \subseteq R$, which is not possible in $\mathcal{LH}$.

$\alpha = P_1 \sqsubseteq P_2$. Assume $(x, y) \in P_1^K$. If $x = a$ and $y = b$ for $a, b \in \text{ind}(K)$, it follows $K \models P_1(a, b)$. From $\alpha$ we obtain that $K \models P_2(a, b)$, therefore $(a, b) \in P_2^K$. If $y = x \cdot ([P], t)$ for some $P$ and $t$, by construction of $P_2^K$, $T \models P \subseteq P_1$. Then because of $\alpha$, $T \models P \subseteq P_2$, so finally, $(x, y) \in R_2^K$.

**Proposition A.34** If $\mathcal{I}$ is a model of $K$, then there exists a homomorphism from $\mathcal{M}$ to $\mathcal{I}$.

**Proof.** Analogous to the proof of Proposition A.31.

**DL-Lite$_{\text{Horn}}$**

Finally, assume $K = (T, A)$ with a DL-Lite$_{\text{Horn}}$ TBox $T$ is given. We construct the finitely generating structure $G = (\Delta^G, \cdot, \models)$ for $K$ as follows, where $\Delta^G = \text{ind}(K) \cup \Omega$. For each $[R]$ with $R$ are role in $T$, we introduce a witness $w_{[R]}$. $\Omega$ will be a subset of the set of all $w_{[R]}$. First define $\models$ as follows:

- $\alpha \models w_{[R]}$ if $\alpha \in \text{ind}(K)$ and $[R]$ is $\leq_T$-minimal such that $K \models \exists R(a)$ and $K \not\models R(a, b)$ for any $b \in \text{ind}(K)$;
- $w_{[S]} \models w_{[R]}$ if $[R]$ is $\leq_T$-minimal with $T \models \exists S \subseteq \exists R$ and $[S] \not\models [R]$.

$\Omega$ is defined as the set of all $w_{[R]}$ such that there are $\alpha \in \text{ind}(K)$ and $R_1, \ldots, R_n = R$ such that

$$\alpha \models w_{[R_1]} \cdots \models w_{[R_n]}.$$

We define the interpretation function $\mathcal{G}$ by setting

$$A^K = \{ \alpha \in \text{ind}(K) \mid K \models A(\alpha) \} \cup \{ w_{[R]} \in \Omega \mid T \models \exists R \subseteq A \},$$

$$P^K = \{ (a, b) \mid \text{there is } R(a, b) \in A \text{ with } T \models R \subseteq P \},$$

and for every edge $u \rightsquigarrow v$ with $v = w_{[R]}$, we set

$$(u, v)^G = \{ R' \mid [R] \leq_T [R'] \}.$$
Proof. We show that \( K \models q[\bar{a}] \) iff \( M \models q[\bar{a}] \), for each CQ \( q(\bar{x}) \) and each tuple of constants \( \bar{a} \subseteq \text{ind}(K) \).

\( \Rightarrow \) Assume \( K \models q[\bar{a}] \). Then for each model \( I \) of \( K \), we have \( I \models q[\bar{a}] \). Since \( M \) is a model of \( K \), we obtain \( M \models q[\bar{a}] \).

\( \Leftarrow \) Let \( M \models q[\bar{a}] \), moreover assume \( \bar{a} = (a_1, \ldots, a_k) \) for \( a_i \in \text{ind}(K) \), and
\[
q(\bar{x}) = \exists y_1 \ldots \exists y_m \phi(x_1, \ldots, x_k, y_1, \ldots, y_m).
\]
Then there exist \( \sigma_1, \ldots, \sigma_m \in \Delta^M \) such that \( M \models \phi[\bar{a}, \sigma_1, \ldots, \sigma_m] \).

Let \( I \) be a model of \( K \), we show that \( I \models q[\bar{a}] \). By Propositions A.31, A.34, A.36, there exists a homomorphism \( h \) from \( M \) to \( I \). Then it is easy to see that
\[
I \models \phi[\bar{a}, \ldots, \sigma_m], h(\sigma_1), \ldots, h(\sigma_m)].
\]
As \( I \) was an arbitrary model of \( K \), it follows that \( K \models q[\bar{a}] \).

\( \square \)

**Proof of Theorem 5**

**Theorem 5** Suppose \( K_2 \) is a consistent KB with a materialisation \( M_i \), \( i = 1, 2 \). Then \( K_1 \) \(-\)-query entails \( K_2 \) iff \( M_2 \) is finitely \( \Sigma \)-homomorphically embeddable into \( M_1 \).

**Proof.** \( \Rightarrow \) Assume \( K_1 \) \(-\)-query entails \( K_2 \). Let \( \Delta \) be a finite subset of \( \Delta^M_2 \) such that \( \Delta = \{\sigma_1, \ldots, \sigma_m\} \) with \( \sigma_i \in \text{Ind}(K_2) \). Consider a query \( q = \exists y_1 \ldots \exists y_m \phi \), where for \( 1 \leq i, i' \leq k \) and \( 1 \leq j, j' \leq m \),
\[
\phi = \bigwedge_{A \in \text{E}^\Delta_2(\sigma_i)} A(a_i) \wedge \bigwedge_{R \in \text{R}^\Delta_2(a_i, a_i')} R(a_i, a_{i'}) \wedge \bigwedge_{R \in \text{R}^\Delta_2(a_i, \sigma_j)} R(a_i, y_j) \wedge \bigwedge_{A \in \text{E}^\Delta_2(\sigma_j')} A(y_j) \wedge \bigwedge_{R \in \text{R}^\Delta_2(\sigma_j', \sigma_j')} R(y_j, y_{j'})
\]
Clearly, \( M_2 \models q \), as \( M_2 \models \phi[\sigma_1, \ldots, \sigma_m] \). By Theorem A.37, \( M_1 \models q \), and thus, \( M_1 \models \phi[\sigma'_1, \ldots, \sigma'_m] \), for some \( \sigma'_1, \ldots, \sigma'_m \in \Delta^M_1 \). We define \( h: \Delta \to \Delta^M_1 \) by taking \( h(a_i) = (a_i) \) and \( h(\sigma_i) = \sigma'_i \). This function is a homomorphism: it maps every constant to itself, and from \( \Delta \) to \( \Delta^M_1 \) by \( h(\sigma_i) = \sigma'_i \). For each \( d, d' \in \Delta \), \( \Delta^M_2(d) \subseteq \Delta^M_1(h(d)) \) and \( \Delta^M_2(d, d') \subseteq \Delta^M_1(h(d), h(d')) \).

\( \Leftarrow \) Assume \( M_2 \) is finitely \( \Sigma \)-homomorphically embeddable into \( M_1 \). Let \( K_2 = (T_2, A_2) \), \( q \) be a \(-\)-query and \((a_1, \ldots, a_k) \subseteq \text{ind}(K_2) \). Let \( \Delta = \{\sigma_1, \ldots, \sigma_m\} \) be a finite subset of \( \Delta^M_2 \) and \( h: \Delta \to \Delta^M_1 \) be a homomorphism from \( \Delta^M_2 \) to \( \Delta^M_1 \) such that the trivial interpretation is a model of all \( \Delta^M_2 \) and \( \Delta^M_1 \). Then \( \Delta \) is a consistent KB with a materialisation \( \Sigma \)-query inseparability for \( K_1 \) and \( K_2 \).

**Proof of Theorem 2**

**Theorem 2** Let \( \mathcal{L} \) be any of our DLs containing \( \mathcal{E} \), \( \mathcal{L} \) having role inclusions. Then \(-\)-query entailment for \( \mathcal{L} \)-KBs is \( \text{LOGSPACE} \)-reducible to \(-\)-query inseparability for \( \mathcal{L} \)-KBs.

**Proof.** We complete the proof given in the paper by showing

Claim 1. \( K_1 \) \(-\)-query entails \( K_2 \) iff \( K_1 \) and \( K_1^{\mathcal{C}} \cup K_2^{\mathcal{C}} \) are \(-\)-query inseparable.

The interesting direction is to show that if \( K_1 \) \(-\)-query entails \( K_2 \), then \( K_1 \) \(-\)-query entails \( K_1^{\mathcal{C}} \cup K_2^{\mathcal{C}} \). Assume that \( K_1 \) \(-\)-query entails \( K_2 \). Consider materialisations \( C_1 \), \( C_2 \), and \( D_1 \) of \( K_1^{\mathcal{C}} \), \( K_2^{\mathcal{C}} \), and \( K_1 \), respectively. The construction of the materialisations above shows that we may assume that

- \( C_1 \) is a model of \( K_1^{\mathcal{C}} \), for \( i = 1, 2 \);
- \( a^{C_1} = a^{C_2} \) for all \( a \in \text{ind}(K_1) \cap \text{ind}(K_2) \);
- \( d \in \Delta^{C_1} \cap \Delta^{C_2} \) if \( d = a^{C_1} \) for some \( a \in \text{ind}(K_1) \cap \text{ind}(K_2) \).

Denote by \( C \) the union of \( C_1 \) and \( C_2 \) defined by setting \( \Delta^C = \Delta^{C_1} \cup \Delta^{C_2} \) and \( X^C = X^{C_1} \cup X^{C_2} \) for all symbols \( X \). We show that

(i) \( C \) is a model of \( K_1^{\mathcal{C}} \cup K_2^{\mathcal{C}} \), and
(ii) \( C \) is finitely \( \Sigma \)-homomorphically embeddable into \( D_1 \).

By (i) and (ii), \( K_1 \) \(-\)-query entails \( K_1^{\mathcal{C}} \cup K_2^{\mathcal{C}} \). Item (i) follows from the assumption that the \( C_i \) are models of \( K_1^{\mathcal{C}} \).

We now consider the case when the trivial interpretation is not a model of \( T_i \). Assume \( K_1 \) and \( K_2 \) are given. We construct \( K'_1 \) and \( K'_2 \) such that the trivial interpretation is a model of \( T_1' \) and \( T_2' \), respectively, and such that \( K_1 \) \(-\)-query entails \( K_2 \) iff \( K'_1 \) \(-\)-query entails \( K'_2 \). The construction is by careful relativisation.

Let \( A_i' \) be fresh concept names, for \( i = 1, 2 \). Set \( A_i' = A \cup \{A_i'(a) \mid a \in \text{Ind}(K_i)\} \).

**Case 1.** \( T_i \) are Horn-\( \mathcal{ALCHI} \)-TBoxes. We assume they are in normal form. Now replace

- any inclusion \( T \subseteq B \) by \( A_i' \subseteq B \);
- any inclusion \( A \subseteq \exists R.B \) by \( A \subseteq \exists R.(A_i' \cap B) \).

The remaining inclusions are not modified. Below we show that the \( K'_i = (T_i', A_i') \) are as required.

Note that the \( K'_i \) are consistent. Consider materialisations (and models) \( C_i \) and \( D_i \) of \( K_i \) and \( K'_i \) respectively. For a subset \( S \subseteq \Delta^{D_i} \), denote by \((S, A_i')\) the set
\[
\{a_1 \cdots a_n \mid a_1 \cdots a_n \in S, n \geq 0\},
\]
where, if $v_j = ([R_j], t_j)$, then $v'_j = ([R_j], t_j \setminus \{A^i\})$, and similarly for a set $S \subseteq \Delta^{D_1} \times \Delta^{D_1}$. We show $\Delta^{D_1} = (A^i)^{D_1}$, $\Delta^{C_i} = (\Delta^{D_1})^{A^i}$, $X^{C_i} = (X^{D_1})^{A^i}$ for each symbol $X$ distinct from $A^i$, from which the required follows. First, by construction of $K_i$ and definition of the generating structure and materialisation, we have

$$(A^i)^{D_1} = \text{ind}(K_i) \cup$$

$$\{av_1 \cdots vn \mid a \in \text{ind}(K_i), v_j \in \Omega, a \rightsquigarrow v_1, v_j \rightsquigarrow v_{j+1}\},$$

thus $\Delta^{D_1} = (A^i)^{D_1}$. On the other hand, clearly $\Delta^{C_i} = \{av'_1 \cdots vn' \mid av_1 \cdots vn \in \Delta^{D_1}\}$ for $v_j = ([R_j], t_j)$ and $v'_j = ([R_j], t_j \setminus \{A^i\})$. Next, let $T_i \models T \subseteq C$ for some concept $C$ such that $C$ is a concept name $B$, or a concept of the form $\exists R.B$ or $\exists R X$, since $T_i$ is in normal form, it follows that $T \subseteq \Omega \setminus A_i$ for some concept name $A$ and $T_i \models A \subseteq C$. Then, $T_i'$ contains axiom $A^i \subseteq A$, and therefore $A^{D_1} = B^{D_1} = \Delta^{D_1}$ if $C$ is a concept name $B$ and $R^{D_1} = \{(x, x \cdot ([R], t)) \mid x \in \Delta^{D_1}, t \text{ is defined accordingly to } x\}$ if $C$ is a concept of the form $\exists R.B$ or $\exists R.X$. On the other hand, $B^{C_i} = \Delta^{C_i}$, or $R^{C_i} = \{(x, x \cdot ([R], t)) \mid x \in \Delta^{C_i}, t \text{ is defined accordingly to } x\}$. Now, if $T_i \not\models T \subseteq C$, it is easy to see that the required holds as well.

Case 2. The $T_i$ are $\mathcal{EL}'$-$\mathcal{T}$-Boxes. We assume they are in normal form. Now replace

- any inclusion $T \subseteq B$ by $A^i \subseteq B$;
- any inclusion $A \subseteq \exists R.B$ by $A \subseteq \exists R.(A^i \cap B)$.

Thus, the construction is the same as above (we do not have to consider this case separately). One can show that the $K_i$ are as required.

Case 3. The $T_i$ are $\mathcal{DL-Lite}^{H}_{\text{core}}$ or $\mathcal{DL-Lite}^{H}_{\text{horn}}$-Boxes. Now replace

- any inclusion $T \subseteq B$ by $A^i \subseteq B$;
- any inclusion $B \subseteq \exists R.B$ by $B \subseteq \exists R \cdot (A^i \cap B)$.

One can show that the $K_i$ are as required.

Next, consider the case when $\mathcal{L}$ is a DL without role inclusions (with conjunction of concepts on the left-hand side). Assume $K_1$ and $K_2$ are given. We construct now $K_1$ and $K_2$ such that $K_1$-query entails $K_2$ iff $K_1 \equiv K_1 \cup K_2$. Define $A^i$ as the union of

- $A_i \cup \{A^i(a) \mid a \in \text{ind}(K_i)\}$
- $\{A(a) \mid K_i \models A(a)\} \cup \{P(a, b) \mid K_i \models P(a, b)\}$.

Define $T_i'$ as follows:

Case 1. If $T_i$ are $\mathcal{Horn-ALCI}$-TBoxes in normal form then replace

- any inclusion $T \subseteq B$ by $A^i \subseteq B$;
- any inclusion $A \subseteq B$ by $A \cap A^i \subseteq B$;
- any inclusion $A \subseteq \exists R.B$ by $A \cap A^i \subseteq \exists R.(A^i \cap B)$;
- any inclusion $A_i \cap A^i_2 \subseteq B$ by $A_i \cap A^i_2 \cap A^i \subseteq B$;
- any inclusion $\exists R.A \subseteq B$ by $A^i \cap \exists R.(A \cap A^i) \subseteq B$;
- any inclusion $A \subseteq \forall R.B$ by $A \cap A^i \subseteq \forall R.(\neg A^i \cup B)$.

Note that we are not in normal form, but still in Horn-ALCI.

Case 2. If the $T_i$ are $\mathcal{EL}$-TBoxes in normal form then the construction is the same except that the final clause does not occur.

Observe that the trivial interpretation is a model of $T_i$. We show that $K_1$-query entails $K_2$ iff $K_1$ and $K_1 \cup K_2$ are $\Sigma$-query inseparable.

Consider materialisations (and models), $C_i$ and $D_i$, of $K_i$ and $K_i'$, respectively. As in the case with careful relativisation, one can show $\Delta^{D_i} = (A^i)^{D_i}$, $\Delta^{C_i} = (\Delta^{D_i})^{A^i}$ and $X^{C_i} = (X^{D_i})^{A^i}$ for each symbol $X$ distinct from $A^i$.

Note that in the case of Horn-ALCII, $(\Delta^{D_1} \setminus \text{ind}(K_1)) \cap (\Delta^{D_2} \setminus \text{ind}(K_2)) = \emptyset$ as for each $d \in \Delta^{D_1} \setminus \text{ind}(K_1)$ such that $\text{tail}(d) = ([S], t)$, we have $A^i \subseteq t$ and $A^i \not\subseteq t$, and the other way around. And in the case of $\mathcal{EL}$, we can assume $(\Delta^{D_1} \setminus \text{ind}(K_1)) \cap (\Delta^{D_2} \setminus \text{ind}(K_2)) = \emptyset$ as we can rename the elements of $\Delta^{D_1}$ to achieve that. Denote by $D$ the union of $D_1$ and $D_2$ defined by setting $\Delta^{D} = \Delta^{D_1} \cup \Delta^{D_2}$ and $X^{D} = X^{D_1} \cup X^{D_2}$ for all symbols $X$. Then $D$ is a model of $K_1$ or $K_2$.

Again, the interesting direction is to show that if $K_1$-query entails $K_2$, then $K_1'$-query entails $K_1' \cup K_2$. Assume that $K_1$-query entails $K_2$, we show $D$ is finitely $\Sigma$-homomorphically embeddable into $D_1$. Let $Y \subseteq \Delta^D$ be finite. Since $K_1$-query entails $K_2$ and $K_1'$ trivially $\Sigma$-query entails $K_1'$ we have $\Sigma$-homomorphisms

- $f_1 : Y \cap \Delta^{D_1} \rightarrow \Delta^{D_1}$ from the $Y$-restriction of $D_1$ to $D_1$
- $f_2 : Y \cap \Delta^{D_2} \rightarrow \Delta^{D_1}$ from the $Y$-restriction of $D_2$ to $D_1$.

We may assume $f_1(a^{D_1}) = f_2(a^{D_2})$ for all $a \in \text{ind}(K_1) \setminus \text{ind}(K_2)$. Then $f_1 \cup f_2$ is a $\Sigma$-homomorphism from the $Y$-restriction of $D$ to $D_1$, as required.

Finally, we consider the case $K_1$ is inconsistent. Let $A^i$ be the ABox extending $A_i$ with

$$\{A(a) \mid a \in \text{ind}(K_2) \land K_2 \not\models q(a)\} \text{ for any } \Sigma\text{-query } q\}$$

for some fresh concept name $A$. We show that $K_1$-query entails $K_2$ iff $K_1' = (T_i, A_i')$ and $K_1 \cup K_2$ are $\Sigma$-query inseparable. Note that $K_1'$ and $K_1 \cup K_2$ are inconsistent. First, from $K_1$-query entails $K_2$ we obtain that $\text{ind}(K_1') = \text{ind}(K_1) \cup \text{ind}(K_2)$, so the “only-if” direction follows immediately. Assume $K_1' \equiv K_1 \cup K_2$, then from their inconsistency, it follows $\text{ind}(K_1') = \text{ind}(K_1) \cup \text{ind}(K_2)$.

By construction of $A_i'$, for each $\Sigma$-query $q$ and tuple of constants $\bar{a} \subseteq \text{ind}(K_2)$ such that $K_2 \models q(\bar{a})$, we obtain $\bar{a} \subseteq \text{ind}(K_1')$. So we conclude $K_1'$-query entails $K_2'$. $\square$

**Proof of Theorem 3**

**Theorem 3** $\Sigma$-query entailment for any of our DLs $\mathcal{L}$ is LOGSPACE-reducible to the membership problem for universal UCQ-solutions in $\mathcal{L}$. 


Proof. We complete the proof given in the paper by considering the case when \( \mathcal{G}_2 \) is not a model of \( \mathcal{G}_1 \). As before, we may assume that \( \Sigma = \operatorname{sig}(\mathcal{K}_1) \cap \operatorname{sig}(\mathcal{K}_2) \). Let \( \Sigma_1 = \operatorname{sig}(\mathcal{K}_1) \). Then \( \mathcal{K}_1 \)-query entails \( \mathcal{K}_2 \) iff \( \mathcal{K}_1 \)-query entails \( \mathcal{K}_2 \).

Define \( \mathcal{K}'_1 = (\mathcal{T}', A') \) to be as in the case \( \mathcal{L} \) is a DL without role inclusions in the proof of Theorem 2, where \( \mathcal{L} \) is DL-Lite\(_{core}\) or DL-Lite\(_{hom}\). \( \mathcal{T}'_1 \) is defined by replacing in \( \mathcal{T}_1 \) any inclusion \( T \subseteq B \) by \( A'_T \subseteq B \), and adding \( \exists R \in A'_T \) for each role \( R \). Moreover, define \( \mathcal{K}''_1 \) to be a copy of \( \mathcal{K}_1' \) in which all symbols \( S \) except for \( A'_T \) are replaced by fresh \( S_i \). Then \( \mathcal{K}_1 \)-query entails \( \mathcal{K}_2 \) iff \( \mathcal{K}_1 \) is a universal UCQ-solution for \( (\mathcal{K}''_1 \cup \mathcal{K}_2', \mathcal{T}_1, \Sigma_1) \), where \( \mathcal{T}_1 = \{ S_i \subseteq S \mid S \in \Sigma_1, i = 1, 2 \} \).

\[ \mathcal{K}_2 \subseteq \mathcal{K}_1 \] is not the case that \( \mathcal{K}_1 \)-query entails \( \mathcal{K}_2 \) iff \( \mathcal{K}_1 \)-query entails \( \mathcal{K}_2 \).

Proof of Theorem 8

Theorem 8 \( \mathcal{M}_2 \) is finitely \( \Sigma \)-homomorphically embeddable into \( \mathcal{M}_1 \) iff the following conditions hold:

- (box) \( \prod_{\mathcal{M}_2}^\Sigma(a, b) \subseteq \prod_{\mathcal{M}_1}^\Sigma(a, b) \), for any \( a, b \in \operatorname{ind}(\mathcal{K}_2) \);
- (win) for any \( u_0 \in \Delta_{\Sigma^2}^2 \) and \( n < \omega \), there exists \( \sigma_0 \in \Delta_{\Sigma^1}^1 \) such that player 1 has an \( n \)-winning strategy in the game \( G_{\Sigma}^2(\mathcal{G}_2, \mathcal{M}_1) \) starting from \( (u_0 \mapsto \sigma_0) \).

Proof. Suppose \( \mathcal{M}_2 \) is finitely \( \Sigma \)-homomorphically embeddable into \( \mathcal{M}_1 \). Then (box) holds by the definition of \( \Sigma \)-homomorphism. To show that (win) holds, suppose \( u_0 \in \Delta_{\Sigma^2}^2 \) and \( n < \omega \) are given. Take the sub-interpretation \( \mathcal{M}_{2,n} \) of \( \mathcal{M}_2 \) that contains \( \sigma_0 u_0, \) for some (say, the shortest) word \( \sigma \), and all those elements of \( \mathcal{M}_2 \) whose distance from \( \sigma \) does not exceed \( n \). Let \( h : \mathcal{M}_{2,n} \to \mathcal{M}_1 \) be a \( \Sigma \)-homomorphism. Take \( \sigma_0 = h(\sigma_0 u_0) \). Clearly, \( u_0 \) and \( \sigma_0 \) satisfy \( (s_1) \) and \( (s_2) \). We show that player 1 has an \( n \)-winning strategy in the game \( G_{\Sigma}^2(\mathcal{G}_2, \mathcal{M}_1) \) starting from the state \( (u_0 \mapsto \sigma_0) \). Suppose player 2 takes \( u_0 \mapsto \Sigma^2 u_1 \). Then \( \sigma_0 u_1 \) is an element of \( \mathcal{M}_{2,n+1} \), and player 1 responds with \( \sigma_1 = h(\sigma_0 u_1) \). Conditions \( (s_1) \) and \( (s_2) \) hold because \( h \) is a \( \Sigma \)-homomorphism. In the same way player 1 uses \( h \) to find responses to all challenges of player 2 in any round \( k < n \) of the game \( G_{\Sigma}^2(\mathcal{G}_2, \mathcal{M}_1) \).

Proof of Lemma 22

Lemma 22 For any \( u_0 \in \Delta_{\Sigma^2}^2 \), condition \( (\prec \omega) \) holds for arbitrary strategies in \( G_{\Sigma}^2(\mathcal{G}_2, \mathcal{M}_1) \) iff \( \omega \) holds in \( G_{\Sigma}^2(\mathcal{G}_2, \mathcal{G}_1) \) for some state \( (\Xi_0 \to x_0, \Psi_0) \) with \( u_0 \in \Xi_0 \).

Proof. Let \( \Xi = \{ S_n \mid n < \omega \} \) be the set of the given \( n \)-winning strategies in \( G_{\Sigma}^2(\mathcal{G}_2, \mathcal{M}_1) \) and suppose that \( S_n \) begins with \( (u_0 \mapsto \sigma_0^0) \), \( n < \omega \).

We define a (possibly infinite) tree \( \Sigma \) whose nodes are of the form \((u \to z, k)\), where \( u \in \Delta_{\Sigma^2}^2 \), \( k \in \Delta_{\Sigma^1}^1 \), \( k < \omega \), whose edges are labelled with \( u \to \Sigma^2 u' \) and the following conditions hold:

1. the root of \( \Sigma \) is of the form \( (u_0 \to w, 0) \), \( w \in \Delta_{\Sigma^1}^1 \);
2. \( t_{\Sigma}^G(u) \subseteq t_{\Sigma}^G(\tau(\Xi)); \)
3. for any node \((u \to z, k)\) in \( \Sigma \) and any \( u \to \Sigma^2 u' \), there exists one \((u \to z, k')\) such that \((u \to z, k) \prec_{\Sigma^2} (u \to z, k') \) in \( \Sigma \), which can be of the following forms:
   - \((u' \to w', k + 1)\), if \( z = w \in \Delta_{\Sigma^1}^1 \), \( w' \to \Sigma^1 w \) and \( \prod_{\Sigma}^G(u, u') \subseteq \prod_{\Sigma}^G(w, w') \);
   - \((u' \to z', k)\), if \( z = z'w \), \( w \in \Delta_{\Sigma^1}^1 \), \( w' \to \Sigma^1 w \) and \( \prod_{\Sigma}^G(u, u') \subseteq \prod_{\Sigma}^G(w, w') \);
   - \((u' \to zw', k)\), if \( z = z'w \), \( w \to \Sigma^1 w' \) and \( \prod_{\Sigma}^G(u, u') \subseteq \prod_{\Sigma}^G(w, w') \);
   - \((u' \to b, -1)\), if \( z = a \in \operatorname{ind}(\mathcal{K}_1) \), \( b \in \operatorname{ind}(\mathcal{K}_1) \) and \( \prod_{\Sigma}^G(u, u') \subseteq \prod_{\Sigma}^G(a, b) \);
4. for any \( k \geq 0 \) and any nodes \((u \to w, k)\), \((u' \to w', k)\) in \( \Sigma \) with \( w, w' \in \Delta_{\Sigma^1}^1 \), it follows \( w = w' \).

We call the tree \( \Sigma \) complete if whenever a node \((u \to z, k)\) is in \( \Sigma \) and \( u \to \Sigma^2 u' \) then some node \((u' \to z', k')\) is its \((u \to \Sigma^2 u')\)-successor in \( \Sigma \). It will be shown later that given a complete tree \( \Sigma \) we can construct an \( \omega \)-winning strategy starting from some \((\Xi_0 \to x_0, \Psi_0)\) in the game \( G_{\Sigma}^2(\mathcal{G}_2, \mathcal{G}_1) \). But first we show how to construct such a tree using \( \Sigma \).

For \( \Xi \in \mathcal{S} \), we say that \( \mathcal{S} \) respects \( \Sigma \) if there exists a map \( f_{\Sigma} : \{ (z, k) \mid (u \to z, k) \in \mathcal{S} \} \to \Delta_{\Sigma^1}^1 \) such that:

1. \( f_{\Sigma}(z, k) = \delta z \), for some \( \delta \);
2. \((u' \to f_{\Sigma}(z, k))\) is in \( \mathcal{S} \), for any \((u \to z, k)\) in \( \mathcal{S} \);
3. \((u' \to z', k')\) is a \((u \to \Sigma^2 u')\)-successor of \((u \to z, k)\) in \( \mathcal{S} \), then, according to \( \mathcal{S} \), player 1 responds to the challenge \( u \to \Sigma^2 u' \) of player 2 in the state \((u \to f_{\Sigma}(z, k))\) with \((u' \to f_{\Sigma}(z', k'))\).
The set $S$ contains an $n$-winning strategy starting from $(u_0 \mapsto \sigma_0^n)$, for any $n < \omega$. As $G_1$ is finite, we can find some $x_0$ such that $x_0 = \text{tail}(\sigma_0^n)$ for infinitely many $n$. Denote by $S_0$ the set of the corresponding strategies from $S$. As an $m$-winning strategy is also an $l$-winning strategy for any $l \leq m$, $S_0$ contains an $m$-winning strategy starting from $(u_0 \mapsto \delta^n x_0)$, for any $n < \omega$. Define $\Sigma_0$ to be a tree with a single node $(u_0 \mapsto x_0, 0)$. For every $S \in S_0$, we set $fs(x_0, 0) = \delta_S x_0$, where $\delta_S$ is the corresponding $\delta^n$. Thus, all the strategies in $S_0$ respect $\Sigma_0$.

Suppose we have already constructed $\Sigma_i$ and $S_i$ such that $S_i$ contains an $n$-winning strategy for any $n < \omega$, and all of them respect $\Sigma_i$. If $\Sigma_i$ is incomplete then it contains a state $(u \mapsto z, k)$ without a $(u \mapsto z', k')$-successor, for some $u \mapsto z'. (We always take such a state that is nearest to the root.) Suppose $fs(z, k) = \delta_S w$. Consider the responses $u' \mapsto \sigma^n$ to the challenge $u \mapsto z'$ according to the $n$-winning strategies in $S_i$, for $n < \omega$. Take some $w' \mapsto (\delta_S^n)$ such that $w' = \text{tail}(\sigma^n)$ for infinitely many $n$. Denote by $S_{i+1}$ the set of the corresponding strategies from $S_i$.

Suppose $w' \mapsto z'$. If $z = w$ then we add the node $(u' \mapsto w', k + 1)$ as a $(u' \mapsto z')$-successor of $(u \mapsto z, k)$ to $\Sigma_i$, thus obtaining $\Sigma_{i+1}$. By the definition of the materialisations, we also have $\delta_S = \delta_S w'$, for all $S \in S_{i+1}$. We then set $fs(w', k + 1) = \delta_S$. If $|z| > 1$ then $z = z' w'$ and $z' w'$ is a suffix of $\delta_S$. In this case, we add the node $(u' \mapsto z' w', k)$ as a $(u' \mapsto z')$-successor of $(u \mapsto z, k)$ to $\Sigma_i$, thus obtaining $\Sigma_{i+1}$, and set $fs(z' w', k) = \delta_S$.

Suppose $w \mapsto z' w'$. In this case, we add $(u' \mapsto z' w')$ as a $(u' \mapsto z')$-successor of $(u \mapsto z, k)$ to $\Sigma_i$, thus obtaining $\Sigma_{i+1}$, and set $fs(z' w', k) = \delta_S w' w'$.

Suppose $w, w' \in \text{ind}(K_i)$ (hence, $\delta'$ is empty). In this case, we add $(u' \mapsto w', -1)$ as a $(u' \mapsto z')$-successor of $(u \mapsto z, k)$ to $\Sigma_i$, thus obtaining $\Sigma_{i+1}$, and set $fs(w, k) = \delta_S w' w'$.

All $S \in S_{i+1}$ clearly respect $\Sigma_{i+1}$. It is easy to see that it satisfies $(4)$.

We proceed in the same way and construct a sequence of growing trees $\Sigma_0 \subseteq \Sigma_1 \subseteq \ldots$ until we reach a complete finite tree $\Sigma_k$, otherwise we take $\Sigma = \bigcup_{n \leq \omega} \Sigma_n$, which is obviously complete.

Now we show that player 1 has an $\omega$-winning strategy starting from some $(\Xi_0 \mapsto x_0, \Psi_0)$ in the game $G^*_{G_2}(G_2, G_1)$. Suppose that we have a complete tree $\Sigma$ with the root $(u_0 \mapsto x_0, 0)$. We then set:

$$\Xi_0 = \{ u | (u \mapsto x_0, 0) \in \Sigma \},$$
$$\Phi_0 = \{ u' | u \mapsto 2 u', u \in \Xi_0, (u' \mapsto x_0w, 0) \in \Sigma \},$$
$$\cup \{ u' | (u' \mapsto b, -1) \in \Sigma \text{ is a } (u \mapsto 2 u') \text{-successor of } (u \mapsto x_0, 0) \in \Sigma \},$$
$$\Psi_0 = \{ u' | u \mapsto 2 u', u \in \Xi_0, (u' \mapsto w, 1) \in \Sigma \}.$$

Note that, by $(4)$, if $(u \mapsto x, 0) \in \Sigma$ (and $|x| = 1$, that is, $x \in \Delta^0$) then $x = x_0$. Moreover, if $x_0 \in \text{ind}(K_1)$, then $\Psi_0 = \emptyset$.

More generally, for any $i > 0$ such that $\Sigma$ contains some $(u \mapsto x, i), |x| = 1$, and $x_{i-1} \notin \text{ind}(K_1)$, we set

$$\Xi_i = \{ u | (u \mapsto x, i) \in \Sigma \},$$
$$\Phi_i = \{ u' | u \mapsto 2 u', u \in \Xi_i, (u' \mapsto xw, i) \in \Sigma \},$$
$$\cup \{ u' | (u' \mapsto b, -1) \in \Sigma \text{ is a } (u \mapsto 2 u') \text{-successor of } (u \mapsto x, i) \in \Sigma \},$$
$$\Psi_i = \{ u' | u \mapsto 2 u', u \in \Xi_i, (u' \mapsto w, i + 1) \in \Sigma \}.$$

Note that, by $(4)$, all $(u \mapsto x, i) \in \Sigma$ with $x \in \Delta^0$ share the same $x$, which we denote by $x_i$. And again, if $x_i \in \text{ind}(K_1)$, then $\Psi_i = \emptyset$.

By $(3)$, the states $s_i = (\Xi_i \mapsto x_i, \Psi_i)$ clearly define the backward part of an $\omega$-winning strategy for player 1 in the game $G^*_{G_2}(G_2, G_1)$ starting from $\emptyset_0$.

Thus, it remains to define $\omega$-winning strategies for the start-bounded game $G^*_{G_2}(G_2, G_1)$ starting from states of the form $(0, \Xi_k \mapsto x_k)$ and first-round challenges $u \mapsto 2 v$ such that $u \in \Xi_k$ and $v \in \Phi_k$.

Let $k \geq 0$ be such that $\Phi_k \neq \emptyset$. We now transform $\Sigma$ into a tree $M_k$ representing an $\omega$-winning strategy for player 1 in the game $G^*_{G_2}(G_2, G_1)$ starting from $(0, \Xi_k \mapsto x_k)$ and first-round challenges $u \mapsto 2 v$ such that $u \in \Xi_k$ and $v \in \Phi_k$.

Thus, $(0, \Xi_k \mapsto x_k)$ is the root of $M_k$ associated with $x_k$. Suppose that we have already defined a node $(\Gamma, \Xi \mapsto w)$ associated with a word $\delta w$. Let $u \in \Xi$ and $u \mapsto 2 v$ be such that the node $(u \mapsto \delta w, k')$ in $\Sigma$, where $k'$ equals to $k$ or $-1$ has a $(u \mapsto 2 v)$-successor of the form $(v \mapsto \delta w', k')$ (if $(\Gamma, \Xi \mapsto w)$ is the root, we also require that $v \in \Phi_k$). Then we add to $M_k$ the node $(\Gamma', \Xi' \mapsto w')$, associated with $\delta w'$, as a $(u \mapsto 2 v)$-successor of $(\Gamma, \Xi \mapsto w)$, where

$$\Xi' = \{ u' | (u' \mapsto \delta w', k') \in \Sigma \},$$
$$\Gamma' = \Xi' \sqcup \Xi.$$

If $(\Gamma, \Xi \mapsto a)$ is associated with $a \in \text{ind}(K_1)$ and the node $(u \mapsto a, k')$, for $u \in \Xi$, where $k'$ equals to $k$ or $-1$, has a $(u \mapsto 2 v)$-successor of the form $(v \mapsto b, -1)$ with $b \in \text{ind}(K_1)$ (note that if $(\Gamma, \Xi \mapsto a)$ is the root, then $\Phi_k = \Xi_k$), then we add to $M_k$ the node $(0, \Xi \mapsto b)$, associated with $b$, as a $(u \mapsto 2 v)$-successor of $(\Gamma, \Xi \mapsto w)$, where

$$\Xi' = \{ u' | (u' \mapsto b, -1) \in \Sigma \}.$$

We claim that $M_k$ thus constructed represents an $\omega$-winning strategy for player 1 in the game $G^*_{G_2}(G_2, G_1)$ starting from $(0, \Xi_k \mapsto x_k)$ and first-round challenges $u \mapsto 2 v$ such that $u \in \Xi_k$ and $v \in \Phi_k$.

$(\ast)$ Given $G^*_{G_2}(G_2, G_1)$ and $u_0 \in \Delta^0$ suppose $(\omega)$ holds for some $x_0, \Xi_0$ and $\Psi_0$ such that $u_0 \in \Xi_0$. Let $n < \omega$, we are going to show there is $s_0 \in \Delta^M$ such that player 1 has an $n$-winning strategy starting from $(u_0 \mapsto s_0)$ in the game $G^*_{G_2}(G_2, M_1)$. To define $s_0$ consider a $N$-winning strategy $S$ of player 1 from $(\Xi_0 \mapsto x_0, \Psi_0)$ for $N = 2 \times 2 |\Omega_1| + 1$, where $\Omega_1$ is such that $\Delta_1 = \text{ind}(K_1) \cup \Omega_1$, and a play

$$(\Xi_m \mapsto x_m, \Psi_m), \ldots, (\Xi_1 \mapsto x_1, \Psi_1)$$

conforming with $S$ such that $\Xi_m = \Xi_0$, $x_m = x_0$, and $\Psi_m = \Psi_0$. Denote by $s_i$ the state $(\Xi_i \mapsto x_i, \Psi_i)$ for $1 \leq i \leq m$. 

Then, either $m < N$ and $\Psi_1 = \emptyset$, or $m = N$ and since the number of all possible states in $G^{\delta}_{\Psi}(G_2, G_1)$ is less than $N$, there are integers $c, r$ such that $m \geq c > c - r \geq 1$ and $\delta = \delta_{c-r}$.

Now, we set $\sigma_0 = \delta'\delta$, where $\delta$ and $\delta'$ are obtained as follows. In the fist case above, $\delta$ is equal to $x_1 \ldots x_m$ and $\delta'$ is any (possibly empty) sequence such that $\delta'\delta \in \Delta^{M_1}$ (such $\delta'$ obviously exists). In the second case $\delta$ is equal to the sequence of length $n + 1$:

$$\delta = x_{c-o} x_{c-o+1} \ldots x_c \cdot \delta_{c-r} \cdot x_{c+1} \ldots x_m$$

where $o = (n - (m - c)) \mod r$, $\delta_{c-r} = x_{c-r+1} x_{c-r+2} \ldots x_c$, and $\delta'$ is obtained as before.

Let $k$ be the length of $\delta$, and $y_i$ denote the $i$-th element of the sequence $\delta$. $1 \leq i \leq k$. We define $\mu(i)$ to be the number such that $y_i = x_{\mu(i)}$. In the first case above $\mu(i) = 1$, whereas in the second case $\mu(i)$ equals

$$\begin{cases} c - (o + i - 1) \mod r, & \text{for } 1 \leq i \leq n - (m - c) + 1, \\ c + i - 1 - (n - (m - c)), & \text{for } n - (m - c) + 2 \leq i \leq n + 1,
\end{cases}$$

which can be presented graphically:

$$\delta = x_{c-o} x_{c-o+1} \ldots x_c \cdot \delta_{c-r} \cdot x_{c+1} \ldots x_m$$

Finally, it remains to produce an $n$-winning strategy $S'$ of player 1 from $(u_0 \rightarrow \sigma_0)$. For each challenge $u_{i-1} \rightarrow u_i$ by player 2 in $G^{\delta}_{\Psi}(G_2, G_1)$, we are going to define $\sigma_i \in \Delta^{M'}$ so that to set the response of player 1 according to $S'$ to be $(u_i \rightarrow \sigma_i)$. We will also define auxiliary $f$-values for these states $(u_i \rightarrow \sigma_i)$ that relate them with the “original” states in $G^{\eta}_{\Psi}(G_2, G_1)$.

We first set $f(u_0 \rightarrow \sigma_0) = (\Xi_{\mu(k)} \rightarrow x_{\mu(k)}, \Psi_{\mu(k)})$ and consider the challenge $u_0 \rightarrow \sigma_0$ by player 2 in $G^{\delta}_{\Psi}(G_2, G_1)$. If $u_1 \in \Psi_{\mu(k)}$ then $k > 1$ by the construction of $\delta$. We set $\sigma_1 = y_1 \ldots y_{k-1}$ and $f(u_1 \rightarrow \sigma_1) = (\Xi_{\mu(k-1)} \rightarrow x_{\mu(k-1)}, \Psi_{\mu(k-1)})$. If $u_1 \notin \Psi_{\mu(k)}$, then consider the start-bounded game $G^{\delta}_{\Psi}(G_2, G_1)$ with the initial state $(\emptyset, \Xi_{\mu(k)} \rightarrow x_{\mu(k)})$ and the first-round challenge $u_0 \rightarrow u_1$ (by the structure of the states in $G^{\delta}_{\Psi}(G_2, G_1)$ this challenge is valid). Let $(\Gamma, \Xi \rightarrow z)$ be the response of player 1 according to $S$, for some $z \in \Delta^{M'}$. If $z \in \text{ind}(K_1)$, we set $\sigma_1 = z$ (note, in this case $\sigma_0 \in \text{ind}(K_1)$), otherwise we set $\sigma_1 = \sigma_0 z$. The $f$-value is defined as $f(u_{i+1} \rightarrow \sigma_{i+1}) = (\Gamma, \Xi \rightarrow z)$.

Suppose now we defined $S'$ for a number of steps $h < n$ and the response of player 1 to the challenge $u_{h-1} \rightarrow u_{h+1}$ from a state $(u_{h \rightarrow} \rightarrow \sigma_{h-1})$ was defined as the state $(u_{h} \rightarrow \sigma_{h})$, moreover assume $\sigma_h = \delta'y_1 \ldots y_k z_1 \ldots z_i$ for $0 \leq k' < k$ and $l \geq 0$ (we have also the value of $f$ for this state). If now there is no valid challenge $u_h \rightarrow u_{h+1}$ then further moves of player 1 need not be defined. Otherwise consider the challenge $u_h \rightarrow u_{h+1}$ of player 2 in $G^{\delta}_{\Psi}(G_2, G_1)$.

Suppose, first, $f(u_h \rightarrow \sigma_h) = (\Gamma', \Xi' \rightarrow x')$ where $x' = \text{tail}(\sigma_h) = z_i$, and by induction hypothesis $u_h \in \Xi'$. Note that in this case, $l > 1$. If $u_h \rightarrow u_{h+1}$ is a challenge also from $(\Gamma', \Xi' \rightarrow x')$ in $G^{\delta}_{\Psi}(G_2, G_1)$, consider the response $(\Gamma', \Xi' \rightarrow z)$, then (nbb) does not hold for this challenge, which means $x' \notin \text{ind}(K_1)$, $u_{h+1} \in \Gamma'$, and $r^{\Psi}_{\Xi'}(u_{h+1}) \subseteq r^{\Psi}_{\Xi'}(z, x')$, where $z$ is the element preceding $z_i$ in $\sigma_h$. Two cases are possible:

- $l = 1$, therefore the predecessor of $(\Gamma', \Xi' \rightarrow x')$ according to $S$ is the starting state $(\emptyset, \Xi_{\mu(k')} \rightarrow x_{\mu(k')})$ of the game $G^{\delta}_{\Psi}(G_2, G_1)$, which has been launched from $(\Xi_{\mu(k')} \rightarrow x_{\mu(k')}, \Psi_{\mu(k')})$ in $G^{\delta}_{\Psi}(G_2, G_1)$. It follows, $r^{\Psi}_{\Xi'}(u_{h+1}, u_{h+1}) \subseteq r^{\Psi}_{\Xi'}(y_k, z_1)$, and as $\Gamma' = \Xi_{\mu(k')}$, we have $u_{h+1} \in \Xi_{\mu(k')}$. So we set $\sigma_{h+1} = \delta'y_1 \ldots y_k z_1$, and $f(u_{h+1} \rightarrow \sigma_{h+1}) = (\Xi_{\mu(k')} \rightarrow x_{\mu(k')}, \Psi_{\mu(k')})$.

- $l > 1$, we consider the predecessor $(\Gamma, \Xi \rightarrow x)$ of $(\Gamma', \Xi' \rightarrow x')$ in $G^{\delta}_{\Psi}(G_2, G_1)$ according to $S$, with $x = z_1$. We have $\Gamma' = \Xi$, hence $u_{h+1} \in \Xi$. So we set $\sigma_{h+1} = \delta'y_1 \ldots y_k z_1 \ldots z_{l-1}$ and $f(u_{h+1} \rightarrow \sigma_{h+1}) = (\Gamma, \Xi \rightarrow x)$.

Alternatively, suppose $f(u_h \rightarrow \sigma_h) = (\Xi' \rightarrow x', \Psi')$, where $x' = \text{tail}(\sigma_h)$, and by induction hypothesis $u_h \in \Xi'$. Then $l = 0$ and $(\Xi' \rightarrow x', \Psi') = (\Xi_{\mu(k')} \rightarrow x_{\mu(k')}, \Psi_{\mu(k')})$. We proceed here as in the base case. If $u_{h+1} \in \Psi_{\mu(k')}$, then $k' > 1$: indeed, by construction of $\delta$, if $k = m$, then $\Psi_1 = \emptyset$, otherwise $k = n + 1$, so provided that $h \leq n$, it cannot be the case $k' = 1$. We set $\sigma_{h+1} = y_1 \ldots y_k z_1 \ldots z_{l-1}$, and $f(u_{h+1} \rightarrow \sigma_{h+1}) = (\Xi_{\mu(k')} \rightarrow x_{\mu(k')}, \Psi_{\mu(k')})$.

If $u_{h+1} \notin \Psi_{\mu(k')}$, then consider the start-bounded game $G^{\delta}_{\Psi}(G_2, G_1)$ with the initial state $(\emptyset, \Xi_{\mu(k')} \rightarrow x_{\mu(k')})$ and the first-round challenge $u_h \rightarrow u_{h+1}$. Let $(\Gamma, \Xi \rightarrow z)$ be the response of player 1 according to $S$, for some $z \in \Delta^{M'}$, $u_{h+1} \in \Xi$. If $z \in \text{ind}(K_1)$, we set $\sigma_{h+1} = z$, otherwise we set $\sigma_{h+1} = \sigma_h z$. The $f$-value is defined as $f(u_{h+1} \rightarrow \sigma_{h+1}) = (\Gamma, \Xi \rightarrow z)$.

We have constructed the strategy $S'$ from $(u_0 \rightarrow \sigma_0)$ in the game $G^{\delta}_{\Psi}(G_2, G_1)$. It can be straightforwardly verified that $S'$ is $n$-winning.

---

**Proofs of Lemmas 11, 13 and 19**

**Lemma 11** For $u_0 \in \Delta^{M_1}$, condition $(\omega)$ holds for forward strategies in $G^{\delta}_{\Psi}(G_2, G_1)$ if $(\omega)$ holds in $G^{\delta}_{\Psi}(G_2, G_1)$ for some state $(u_0 \rightarrow x_0)$.

**Proof.** Can be obtained as a corollary of the proof of Lemma 22. Given $u_0 \in \Delta^{M_1}$, it suffices to observe that condition $(\omega)$ holds for forward strategies in $G^{\delta}_{\Psi}(G_2, G_1)$.
iff $(\omega)$ holds in $G^i_\Sigma(G_2, G_1)$, where all the states of the kinds $(\Xi_i \mapsto x_i, \Psi_i)$ and $(\Gamma_i, \Xi_i \mapsto x_i)$ are such that $\Xi_i = \{u\}$ for $u \in \Delta^2$ and $\Psi_i = \emptyset$, for some state $(\{u_0\} \mapsto x_0, \emptyset)$. Such restricted $G^i_\Sigma(G_2, G_1)$ can be straightforwardly converted to $G^i_\Sigma(G_2, G_1)$. 

\[ \square \]

**Lemma 13** For $u_0 \in \Delta^2$, condition $(< \omega)$ holds for backward strategies in $G_\Sigma(G_2, M_1)$ iff $(\omega)$ holds in $G^i_\Sigma(G_2, G_1)$ for some state $(\{u_0\} \mapsto x_0)$. 

**Proof.** Can be obtained as a corollary of the proof of Lemma 22. Given $u_0 \in \Delta^2$, it suffices to observe that condition $(< \omega)$ holds for backward strategies in $G_\Sigma(G_2, M_1)$ iff $(\omega)$ holds in $G^i_\Sigma(G_2, G_1)$, where all the states of the kind $(\Xi_i \mapsto x_i, \Psi_i)$ occur and $\Psi_i = \Xi_i^\sim$, for some state $(\{u_0\} \mapsto x_0, \{u_0\}^\sim)$. Such restricted $G^i_\Sigma(G_2, G_1)$ can be straightforwardly converted to $G^i_\Sigma(G_2, G_1)$. 

\[ \square \]

**Lemma 19** For any $u_0 \in \Delta^2$, condition $(< \omega)$ holds for start-bounded strategies in $G_\Sigma(G_2, M_1)$ iff $(\omega)$ holds in $G^i_\Sigma(G_2, G_1)$ for some state $(\emptyset, \Xi_0 \mapsto x_0)$ with $u_0 \in \Xi_0$. 

**Proof.** Can be obtained as a corollary of the proof of Lemma 22. Given $u_0 \in \Delta^2$, it suffices to observe that condition $(< \omega)$ holds for start-bounded strategies in $G_\Sigma(G_2, M_1)$ iff $(\omega)$ holds in $G^i_\Sigma(G_2, G_1)$, where all the states of the kind $(\Xi_i \mapsto x_i, \Psi_i)$ are such that $\Psi_i = \emptyset$, for some state $(\Xi_0 \mapsto x_0, \emptyset)$ with $u_0 \in \Xi_0$. Such restricted $G^i_\Sigma(G_2, G_1)$ can be straightforwardly converted to $G^i_\Sigma(G_2, G_1)$. 

\[ \square \]

**Proof of Lemma 15**

**Lemma 15** Checking $(\omega)$ in Lemma 13 is coNP-hard.

**Proof.** The proof is by reduction of the unsatisfiability problem for 3CNFs $\varphi = \bigwedge_{i=1}^m c_i$, where $c_i = l_{i1} \lor l_{i2} \lor l_{i3}$ and each $l_{ij}$ is either one of the propositional variables $v_1, \ldots, v_k$ or a negation of such a variable.

Let $p_1, \ldots, p_k$ be the first $k$ prime numbers (observe that $1 < p_j \leq k^2$, for all $j$). We take a role name $R_i$, a role name $\bar{C}_i$, for each clause $c_i$ in $\varphi$, and role names $S_{ij}$ for $1 \leq j \leq k$, and $1 \leq \ell \leq p_j$. Now we define a KB $K_1 = (T_1; \{\ldots R(a)\})$, where $T_1$ contains the following inclusions, for $1 \leq j \leq k$, and $1 \leq \ell < p_j$,

\[
\exists R^\dagger \sqsubseteq \exists S_{j1}, \quad \exists S_{j\ell}^\dagger \sqsubseteq \exists S_{j\ell+1}, \quad \exists S_{p_j}^\dagger \sqsubseteq \exists S_{j1},
\]

and the following inclusions, for $1 \leq j \leq k$, and $1 \leq i \leq m$:

\[
S_{j1} \sqsubseteq C_i, \quad \text{if } v_j \text{ is a literal of } c_i,
\]

\[
S_{j2} \sqsubseteq \neg C_i, \quad \text{if } \neg v_j \text{ is a literal of } c_i.
\]

Intuitively, $M_2$ is a tree with $k$ branches with a common root edge $R$. The $j$th branch is obtained by unravelling the loop of $p_j$ arrows $S_{j1}, \ldots, S_{jp_j}$; the first arrow, $S_{j1}$, corresponds to $v_j$ being true (in an assignment), while the second arrow, $S_{j2}$, to $v_j$ being false. Therefore, $p_1 \times p_2 \times \cdots \times p_k$ layers (a layer $i$ consists of all edges from points at the distance $i$ from the root) contain representations of all possible assignments to $v_1, \ldots, v_k$ (see figure below). The last two types of role inclusions make sure that roles $C_1, \ldots, C_m$, which constitute the signature $\Sigma$, mark those assignments on which $\varphi$ is true.

\[ \square \]

**Proof of Theorem 24**

**Theorem 24** For data complexity, $\Sigma$-query entailment and inseparability are P-hard for DL-Lite$_{core}$ and $\mathcal{EL}$-KBs.

**Proof.** The proof is by reduction of the P-complete entailment problem for acyclic Horn ternary clauses: given a conjunction $\varphi$ of clauses of the form $a_i \land a_i \land \rightarrow a_j$, decide whether $a_n$ is true in every model of $\varphi$. Consider a DL-Lite$_{core}$ TBox $T$ containing the CIs

\[
V \sqsubseteq \exists P, \quad \exists P^\dagger \sqsubseteq \exists R^\dagger, \quad \exists R^\dagger \sqsubseteq V, \quad \text{for } i = 1, 2, \ldots
\]

and let an ABox $A$ be comprised of $F(a_n)$ and

\[
P(a_1, a_2), R_1(a_1, a_2), R_2(a_1, a_2), \quad \text{for each clause } c_i \in \varphi,
\]

\[
P(a_j, c), R_1(c, a_1), R_2(c, a_2), \quad \text{for } c = a_i \land a_i \land \rightarrow a_j \in \varphi.
\]

Set $\Sigma = \{F, P, R_1, R_2\}$, $K_2 = (\mathcal{T}, A \cup \{V(a_n)\})$ and $K_1 = (\emptyset, A)$. Obviously, $K_2 \Sigma$-query entails $K_1$. On the other hand, the materialisation of $K_2$ is (finitely) $\Sigma$-homomorphically embeddable in the materialisation of $K_1$. 

We define $K_1 = (T_1, \{A(a)\})$, where $T_1$ consists of the following inclusions, for $1 \leq i, i' \leq m$,

\[
A \sqsubseteq \exists T_i, \quad \exists T_i^\dagger \sqsubseteq \exists T_i, \quad \exists T_i \sqsubseteq C_i', \quad \text{if } i' \neq i.
\]

In $M_1$, the path from each point to the root contains edges that are labelled by all of $C_1, \ldots, C_m$ but one (note that the $C_i$ edges point towards to root, in the opposite direction to the $C_i$ edges of $M_2$). Therefore, there is a finite $\Sigma$-homomorphism iff in each of the assignments one of the clauses is false (that is, iff $\varphi$ is unsatisfiable).

The generating structure $G_1$ is essentially a set of loops each of which is missing precisely one of the $C_i$. Therefore, the responses of player 1 correspond to choices of the missing $C_i$. Challenges by player 2, on the other hand, correspond to the subsets of $C_1, \ldots, C_m$ in the layers of $M_2$, the number of which may be exponential in $k$. Thus, player 2 can go through a sequence of exponentially many distinct challenges (assignments), to each of which player 1 will have to find a clause that is false under the assignment. The sequence, however, repeats itself after $p_1 \times p_2 \times \cdots \times p_k$ steps. 

\[ \square \]
 iff \( \varphi \) derives \( a_n \). Indeed, the materialisation \( \mathcal{M}_2 \) of \( K_2 \) is infinite, while finite materialisation \( \mathcal{M}_1 \) of \( K_1 \) is finite. So, the only way to embed finite prefixes of \( M_2 \) of arbitrary depth into \( \mathcal{M}_1 \) is by mapping subtrees of unbounded depth into the loops in \( \mathcal{M}_1 \) for unary clauses \( a_i \) in \( \varphi \), which is only possible if there is a tree of rules of the form \( a_j \land a_{ij} \rightarrow a_i \) with root \( a_n \) and leaves among the clauses \( a_i \) of \( \varphi \) (that is, if there is a derivation of \( a_n \) from \( \varphi \)).

For \( \mathcal{E} \mathcal{L} \), we can take \( T = \{ V \subseteq P.\{ R_{1}, V \cap P.R_{2}, V \} \} \). \( \square \)

**Proof of Theorem 25**

**Theorem 25** For combined complexity, \( \Sigma \)-query entailment and inseparability are (i) \( 2\text{ExpTime}\)-hard for Horn-\( \mathcal{ALCI} \) KBs and (ii) \( \text{ExpTime}\)-hard for \( \text{DL-Lite}_r^\mathcal{K} \) KBs.

**Proof.** The proof of (ii) is by encoding alternating Turing machines (ATMs) with polynomial tape and using the fact that \( \text{APSPACE} = \text{ExpTime} \).

As in the proof of (i), let \( M = (\Gamma, Q, (q_0, q_1, \delta) \) be an ATM and let \( M' \) be the ATM obtained from \( M \) by extending it with two instructions that go into an infinite loop if \( M \) reaches the accepting state. Our aim is to construct, given \( M \) and an input \( \vec{w} \), two TBoxes, \( T_1 \) and \( T_2 \), and a signature \( \Sigma \) such that \( M' \) has a run with only infinite branches iff the materialisation \( \mathcal{M}_2 \) of \( (T_2, (\{A(c)\}) \) is finitely \( \Sigma \)-homomorphically embeddable into the materialisation \( \mathcal{C}_1 \) of \( (T_1, (\{A(c)\}) \). Let \( f \) be a polynomial such that, on any input of length \( m \), \( M' \) uses at most \( n = f(m) \) cells.

The construction proceeds in four steps. In the definition of the TBoxes \( T_1 \) and \( T_2 \), we use concept inclusions of the form \( B \subseteq \exists R.(C_1 \cap \ldots \cap C_k) \) as an abbreviation for

\[
B \subseteq \exists R_0, \quad R_0 \subseteq R \quad \text{and} \quad \exists R_0 \subseteq C_i, \quad \text{for} \quad 1 \leq i \leq k,
\]

where \( R_0 \) is a fresh role name. If \( C_i \) is a complex concept then \( \exists R_0 \subseteq C_i \) is also treated as an abbreviation for the respective concept and role inclusions.

**Step 1.** First we encode configurations and transitions of \( M' \) using \( T_1 \). We represent a configuration (that is, the contents of every cell on the tape, the state and the position of the head) by a sequence of \( (n + 2) \) domain elements connected by some role \( R \), which will be called a block. More precisely, the first element in each block is used to distinguish the type of the block. Each of the remaining \( n \) elements is assigned an index from 0 to \( n \). They encode the contents of the tape: if the element with index \( i \) belongs to \( C_a \), for some \( a \in \Gamma \), then the \( i \)th cell of the tape is assumed to contain \( a \) in the configuration defined by the block (cell 0 contains marker \( \bar{a} \in \Gamma \)) as shown below:

\[
\mathcal{M}_1 = \begin{array}{cccccccc}
A & P & P & P & \cdots & P & P & \downarrow \vec{w}, n \\
C_0 & C_{a_1} & C_{a_2} & C_{a_{n-1}} & C_{a_n} & \downarrow Z_{q_0,a_1,1}^{0,n}
\end{array}
\]

The first block represents the initial configuration, that is, symbols \( a_1, \ldots, a_n \) written in the \( n \) cells of the tape (comprising the input \( \vec{w} \) in the first \( m \) cells padded with blanks) and the initial state \( q_0 \), which is achieved by the following inclusion in \( T_1 \):

\[
A \subseteq \exists P.(C_0 \cap \exists P.(C_{a_1} \cap \exists P.(C_{a_2} \cap \exists P.( \ldots \exists P.(C_{a_n} \cap Z_{q_0,a_1,1}^{0,n}, \ldots))))). \quad (T_1-1)
\]

**Step 2.** The contents of the tape and the head position in each configuration is encoded in a block of length \( n + 2 \); the current state \( q \in Q \) and the position \( k \) of the head are recorded in the concept \( Z_{q,a,k}^{1,n} \), that contains the last element of the block \( (a \in \Gamma \) specifies the contents of the active cell scanned by the head). At the end of the block we branch out one block for each of the two transitions and propagate via the \( Z_{q,a,k}^{1,n} \) and the \( Z_{q,a,k}^{2,n} \) the current state, head position and symbol in the active cell: for \( q \in Q \), \( a \in \Gamma \) and \( 1 \leq k \leq n \), we add to \( T_1 \) the inclusions

\[
Z_{q,a,k}^{0,n} \subseteq \bigcap_{j=1,2} \exists P.(X_j \cap Z_{q,a,k}^{j-1}), \quad (T_1-2)
\]

where \( X_1 \) and \( X_2 \) are two fresh concept names (distinguishing the two branches).

The acceptance condition for \( M' \) is enforced by means of \( T_2 \), which uses four types of blocks. The initial configuration is encoded by the following inclusion in \( T_2 \):

\[
A \subseteq \exists P.\exists P_0 \cdots \exists P_n.(\exists P.X_1 \cap \exists P.X_2). \quad (T_2-1)
\]

Two types of blocks, starting with \( X_1 \) and \( X_2 \), respectively, represent configurations with universal states; and one more type of blocks, starting with \( X_3 \), suffices for representing configurations with existential states. These blocks are arranged into an infinite tree-like structure: the block starting with \( A \) is the root, from which an \( X_1 \)- and an \( X_2 \)-blocks branch out (successors of the initial state \( q_0 \) are universal). Each of them is followed by an \( X_3 \)-block (an existential state), which branches out an \( X_1 \)- and an \( X_2 \)-blocks, and so on. This is achieved by adding to \( T_2 \) the following inclusions: for \( j = 1, 2 \),

\[
X_j \subseteq \exists P.\exists P.(G \cap \exists P.(\cdots \exists P.(G \cap \exists P.X_3)) \cdots \cdots \), \quad (T_2-2)
\]

where \( G \) is a concept name (containing all domain elements representing the tape). If \( \Sigma = \{ A, X_1, X_2, P \} \) then there is a unique \( \Sigma \)-homomorphism from the \( A \)-block in \( M_2 \) to the block of the initial configuration in \( M_1 \). Next, concepts \( X_1 \) and \( X_2 \) ensure that the following \( X_1 \)- and \( X_2 \)-blocks are \( \Sigma \)-homomorphically mapped (in a unique way) into the respective blocks in \( M_1 \), which reflects the acceptance condition of universal states. The following block, however, begins with \( X_3 \), which is not in the signature, and thus can be mapped to either of the blocks in \( M_1 \), which reflects the choice in existential states; see the picture below, where possible \( \Sigma \)-homomorphisms are shown by thick dashed arrows:
Step 3. Recall that the \( Z_{q,a,k}^{i,j} \), for \(-1 \leq i \leq n\), specify the position \( k \) of the head on the tape. Let the active cell in the current configuration be \( k \); then until the cell \( k - 2 \) is reached in a successive configuration, the following inclusions in \( T_1 \) propagate the state \((q \in Q)\), the symbol in the active cell \((a \in \Gamma)\), the head position \((1 \leq k \leq n)\) and the branch marker \((j = 0, 1, 2)\) along the domain elements constituting blocks: for \(-1 \leq i \leq n\) with \( i \neq k - 1\),

\[
Z_{q,a,k}^{i,k-2} \subseteq \bigcap_{b \in \Gamma} \exists P.(C_b \cap Z_{q,a,k}^{q,a,k}) \quad (T_1-3)
\]

(for each \( b \in \Gamma \), these inclusions generate a branch in \( M_1 \) to represent the same cell but with a different symbol, \( b \), tentatively assigned to the cell; Step 4 will ensure that the correct branch and symbol are selected to match the cell contents in the preceding configuration). We point out that, since the size of the tape is polynomial in the length of the input, we can use the subscripts of the \( Z_{q,a,k}^{i,j} \) to specify the head position, \( k \), and the cell number, \( i \); in the proof of item (i), we had to use \( P \)-counters over \( H \) and the \( T_j \) respectively. When the cell \( k - 2 \) is reached, the contents of the active cell, the current state and the head position are changed according to \( \delta \):

\[
Z_{q,a,k}^{j,k-2} \subseteq \bigcap_{b \in \Gamma} \exists P.(C_b \cap \Delta_{q,a,b}^{q,a,k}) \quad (T_1-4)
\]

where \( \delta(q,a,j) = (q',a',\sigma) \) and \( \Delta_{q,a,b}^{q,a,k} \) is the concept

\[
\exists P.(C_{a'} \cap G_{a'} \cap Z_{q',b,k}^{q',b,k-1}), \quad \text{if } \sigma = -1,
\]

\[
\exists P.(C_{a'} \cap G_{a'} \cap Z_{q',b,k}^{0,k}), \quad \text{if } \sigma = 0,
\]

\[
\exists P.(C_{a'} \cap G_{a'} \cap \bigcap_{b' \in \Gamma} \exists P.(C_{b'} \cap Z_{q,b',k+1}^{q,b',k})), \quad \text{if } \sigma = +1
\]

(Note that there is only one branch for the modified cell, which corresponds to the new symbol, \( a' \), in that cell; see explanations below.) Then the current state and the symbol in the active cell are further propagated along the tape using \((T_1-3)\) with \( j = 0 \) and \( i > k - 1 \).

Step 4. The inclusions \((T_1-3)-\) \((T_1-4)\) generate a separate \( P \)-successor for each \( b \in \Gamma \). The correct one is chosen by a finite \( \Sigma \)-homomorphism, \( h \), from \( M_2 \) to \( M_1 \). To exclude wrong choices, we take

\[
\Sigma = \{ A, P, X_1, X_2, P \} \cup \{ D_a \mid a \in \Gamma \}.
\]

Recall that if \( d_1 \in C_a^{M_1} \), for some \( a \in \Gamma \), then it represents a cell containing \( a \). The following inclusions in \( T_1 \) ensure that, for each \( b \in \Gamma \) different from \( a \), there is a block of \( n + 2 \)-many \( P^* \)-connected elements that ends in the concept \( D_b \) (called a \( D_b \)-block in the sequel):

\[
C_a \subseteq D_a \cap \bigcap_{b \in \Gamma \setminus \{a\}} G_b, \quad (T_1-5)
\]

\[
G_b \subseteq \exists P^-, \exists P^-, \cdots \exists P^-, \exists P^- D_b, \quad \text{for } b \in \Gamma. \quad (T-1)
\]

(Note that in this proof we do not need to use binary counters to reach the end of the block.) Suppose \( h(d_2) = d_1 \) and \( d_2 \) belongs to \( G \) in \( M_2 \) (it represents a cell in a non-initial configuration). Then \((T-1)\) and the inclusions

\[
G \subseteq \bigcap_{b \in \Gamma} G_b \quad (T_2-4)
\]

added to \( T_2 \) generate a \( D_b \)-block, for each \( b \in \Gamma \) (including \( a \)). Each of the \( D_b \)-blocks in \( M_2 \), for \( b \in \Gamma \) with \( b \neq a \), can be mapped by \( h \) to the respective \( D_b \)-block in \( M_1 \). By the choice of \( \Sigma \), the only remaining \( D_a \)-block, in case \( a \) is tentatively contained in this cell, could be mapped (in the reverse order) along the branch in \( M_1 \) but only if the cell contains \( a \) in the preceding configuration (that is, the element which is \( n + 2 \) steps closer to the root of \( M_1 \) belongs to \( D_a \)).
Note (see $\Delta_{pa,b}^k$) that the cell whose contents is changed generates the additional $D_a$-block in $M_1$ to allow the respective $D_a$-block from $M_2$ to be mapped there.

One can show now that $T_1$ and $T_2$ are as required: $M'$ has a run with only infinite branches iff the materialisation $M_2$ of $(T_2, \{A(c)\})$ is finitely $\Sigma$-homomorphically embeddable into the materialisation $M_1$ of $(T_1, \{A(c)\})$, where $\Sigma$ contains the concept and role names in $T_2$. It remains to use Theorem 5 and the fact that $\text{APSPACE} = \text{ExpTime}$. By Theorem 2, $\Sigma$-query inseparability is also $\text{ExpTime}$-hard.

References