Query Inseparability for Description Logic Knowledge Bases

E. Botoeva,[1] R. Kontchakov,[2] V. Ryzhikov,[1] F. Wolter[3] and M. Zakharyaschev[2]

[1]Faculty of Computer Science
Free University of Bozen-Bolzano, Italy
{botoeva,ryzhikov}@inf.unibz.it

[3]Department of Computer Science
University of Liverpool, U.K.
wolter@liverpool.ac.uk

[2]Department 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-$\mathcal{ALCHI}$*, 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 2EXPTIME. We also resolve two major open problems for *OWL 2 QL* by showing that TBox query inseparability and the membership problem for universal UCQ-solutions in knowledge exchange are both EXPTIME-complete for combined complexity.

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 (Stuckenschmidt, Parent, and Spaccapietra 2009) and knowledge exchange (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 that is fundamental for all such tasks. Let Σ be a signature consisting of concept and role names. We call KBs $\mathcal{K}_1$ and $\mathcal{K}_2$ Σ-*query inseparable* and write $\mathcal{K}_1 \equiv_\Sigma \mathcal{K}_2$ if any CQ formulated in Σ has the same answers over $\mathcal{K}_1$ and $\mathcal{K}_2$. Note that even for Σ containing all concept and role names, Σ-query inseparability does not necessarily imply logical equivalence. The relativisation to (smaller) signatures is crucial to support the tasks mentioned above:

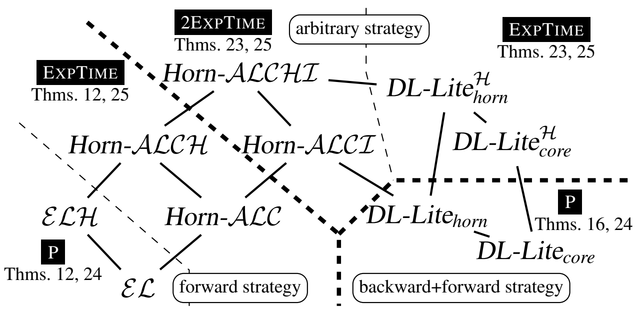
(versioning) When comparing two *versions* $\mathcal{K}_1$ and $\mathcal{K}_2$ of a KB with respect to their answers to CQs in a relevant signature Σ, the basic task is to check whether $\mathcal{K}_1 \equiv_\Sigma \mathcal{K}_2$.

(modularisation) A Σ-*module of* a KB $\mathcal{K}$ is a KB $\mathcal{K}' \subseteq \mathcal{K}$ such that $\mathcal{K}' \equiv_\Sigma \mathcal{K}$. If we are only interested in answering CQs in Σ over $\mathcal{K}$, then we can achieve our aim by querying any Σ-module of $\mathcal{K}$ instead of $\mathcal{K}$ itself.

(knowledge exchange) In knowledge exchange, we want to transform a KB $\mathcal{K}_1$ in a signature Σ_1 to a new KB $\mathcal{K}_2$ in a disjoint signature Σ_2 connected to Σ_1 via a declarative mapping specification given by a TBox $\mathcal{T}_{12}$. Thus, the target KB $\mathcal{K}_2$ should satisfy the condition $\mathcal{K}_1 \cup \mathcal{T}_{12} \equiv_{\Sigma_2} \mathcal{K}_2$, in which case it is called a *universal UCQ-solution* (CQ and UCQ inseparabilities coincide for Horn DLs).

(forgetting) A KB $\mathcal{K}'$ results from *forgetting* a signature Σ in a KB $\mathcal{K}$ if $\mathcal{K}' \equiv_{\mathsf{sig}(\mathcal{K}) \setminus \Sigma} \mathcal{K}$ and $\mathsf{sig}(\mathcal{K}') \subseteq \mathsf{sig}(\mathcal{K}) \setminus \Sigma$. Thus, the result of forgetting Σ does not use Σ and gives the same answers to CQs without symbols in Σ as $\mathcal{K}$.

We investigate the data and combined complexity of deciding Σ-query inseparability for KBs given in various fragments of the DL *Horn-$\mathcal{ALCHI}$* (Krötzsch, Rudolph, and Hitzler 2013), which include *DL-Lite*$^{\mathcal{H}}_{core}$ (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, Σ-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 EXPTIME-completeness of *DL-Lite*$^{\mathcal{H}}_{core}$ and 2EXPTIME-completeness of *Horn-$\mathcal{ALCI}$*, which contrast with NP-completeness and EXPTIME-completeness of CQ evaluation for those logics. For *DL-Lite* without role inclusions and $\mathcal{ELH}$, Σ-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 Σ-query inseparability hard. To establish the upper complexity bounds, we develop a uniform game-theoretic technique for checking finite Σ-homomorphic embeddability between (possibly infinite) materialisations of KBs.



Σ-query inseparability for KBs has not been investigated systematically before. The polynomial upper bound for $\mathcal{EL}$ 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*$^{\mathcal{N}}_{bool}$ (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*$^{\mathcal{H}}_{core}$ is ExpTime-complete for combined complexity, which settles an open question of (Arenas et al. 2013), where only PSpace-hardness was established. We also show that Σ-query inseparability of *DL-Lite*$^{\mathcal{H}}_{core}$ TBoxes is ExpTime-complete, which closes the PSpace–ExpTime gap that was left open by Konev et al. (2011).

Recall that TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ are Σ-query inseparable if, for *all* Σ-ABoxes $\mathcal{A}$ (which only use concept and role names from Σ), the KBs $(\mathcal{T}_1, \mathcal{A})$ and $(\mathcal{T}_2, \mathcal{A})$ are Σ-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* Σ-ABox. In contrast, KB inseparability is useful in applications where data is stable—such as knowledge exchange or variants of module extraction and forgetting with fixed data—in order to use the KB in a new application or as a compilation step to make CQ answering more efficient. As we show below, TBox and KB Σ-query inseparabilities also have different computational properties.

TBox Σ-query inseparability has been extensively studied (Kontchakov, Wolter, and Zakharyaschev 2010; 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, Walther, 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/KR2014.pdf.

Horn-$\mathcal{ALCHI}$ and its Fragments

All the DLs for which we investigate KB Σ-query inseparability are Horn fragments of $\mathcal{ALCHI}$. 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^-; we assume that $(P_i^-)^- = P_i$. $\mathcal{ALCI}$*-concepts*, C, are defined by the grammar

$$C ::= A_i \mid \top \mid \bot \mid \neg C \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \exists R.C \mid \forall R.C,$$

where R is a role. $\mathcal{ALC}$*-concepts* are $\mathcal{ALCI}$-concepts without inverse roles; $\mathcal{EL}$*-concepts* are $\mathcal{ALC}$-concepts without the constructs $\bot$, $\sqcup$, $\neg$ and $\forall R.C$. *DL-Lite*$_{horn}$*-concepts* are $\mathcal{ALCI}$-concepts without $\sqcup$, $\neg$ and $\forall R.C$, in which $C = \top$ in every occurrence of $\exists R.C$. Finally, *DL-Lite*$_{core}$*-concepts* are *DL-Lite*$_{horn}$-concepts without $\sqcap$; in other words, they are *basic concepts* of the form $\bot$, $\top$, A_i or $\exists R.\top$.

For a DL $\mathcal{L}$, an $\mathcal{L}$*-concept inclusion* (CI) takes the form $C \sqsubseteq D$, where C and D are $\mathcal{L}$-concepts. An $\mathcal{L}$*-TBox*, $\mathcal{T}$, contains a finite set of $\mathcal{L}$-CIs. An $\mathcal{ALCHI}$, *DL-Lite*$^{\mathcal{H}}_{horn}$ and *DL-Lite*$^{\mathcal{H}}_{core}$ TBox can also contain a finite set of *role inclusions* (RIs) $R_1 \sqsubseteq R_2$, where the R_i are roles. In $\mathcal{ELH}$ TBoxes, RIs do not have inverse roles. *DL-Lite* TBoxes may also contain *disjointness constraints* $B_1 \sqcap B_2 \sqsubseteq \bot$ and $R_1 \sqcap R_2 \sqsubseteq \bot$, 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 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 it occurs negatively (positively) in $\neg C'$ and $C' \sqsubseteq D$. Now, we call a TBox $\mathcal{T}$ *Horn* if no concept of the form $C \sqcup D$ occurs positively in $\mathcal{T}$, and no concept of the form $\neg C$ or $\forall R.C$ occurs negatively in $\mathcal{T}$. 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*-TBoxes are Horn by definition.

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

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

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

For combined complexity, the problem '$\mathcal{K} \models \boldsymbol{q}(\vec{a})$?' is NP-complete for the *DL-Lite* logics (Calvanese et al. 2007), $\mathcal{EL}$ and $\mathcal{ELH}$ (Rosati 2007), and ExpTime-complete for the remaining Horn DLs above (Eiter et al. 2008). For data complexity (with fixed $\mathcal{T}$ and $\boldsymbol{q}$), this problem is in AC^0 for the *DL-Lite* logics (Calvanese et al. 2007) and P-complete for the remaining DLs (Rosati 2007; Eiter et al. 2008).

A *signature*, Σ, is a set of concept and role names. By a Σ-concept, Σ-role, Σ-CQ, etc. we understand any concept, role, CQ, etc. constructed using the names from Σ.

Σ-Query Entailment and Inseparability

We define the central notions of this paper.

Definition 1 Let $\mathcal{K}_1$ and $\mathcal{K}_2$ be KBs and Σ a signature.

- $\mathcal{K}_1$ *Σ-query entails* $\mathcal{K}_2$ if $\mathcal{K}_2 \models \boldsymbol{q}(\vec{a})$ implies $\mathcal{K}_1 \models \boldsymbol{q}(\vec{a})$ for *all* Σ-CQs $\boldsymbol{q}(\vec{x})$ and all $\vec{a} \subseteq \mathsf{ind}(\mathcal{K}_2)$.
- $\mathcal{K}_1$ and $\mathcal{K}_2$ are *Σ-query inseparable* if they Σ-query entail each other. In this case we write $\mathcal{K}_1 \equiv_\Sigma \mathcal{K}_2$.

Observe that Σ-query inseparability is weaker than logical equivalence even if $\Sigma = \mathsf{sig}(\mathcal{K}_1) \cup \mathsf{sig}(\mathcal{K}_2)$, where $\mathsf{sig}(\mathcal{K}_i)$ is the signature of $\mathcal{K}_i$. For example, $(\emptyset, \{A(a)\})$ is $\{A, B\}$-query inseparable from $(\{B \sqsubseteq A\}, \{A(a)\})$ but the two KBs are clearly not logically equivalent. Since checking Σ-query inseparability can be reduced to two Σ-query entailment checks, we can prove complexity upper bounds for entailment. Conversely, for most languages we have a semantically transparent reduction of Σ-query entailment to Σ-query inseparability:

Theorem 2 *Let $\mathcal{L}$ be any of our DLs containing $\mathcal{EL}$ or having role inclusions. Then Σ-query entailment for $\mathcal{L}$-KBs is* LogSpace*-reducible to Σ-query inseparability for $\mathcal{L}$-KBs.*

Proof sketch. Let $\mathcal{K}_i = (\mathcal{T}_i, \mathcal{A}_i)$, $i = 1, 2$, and Σ be given. We may assume that $\Sigma = \mathsf{sig}(\mathcal{K}_1) \cap \mathsf{sig}(\mathcal{K}_2)$. We also assume that $\mathcal{L}$ has role inclusions, $\mathcal{K}_1$ and $\mathcal{K}_2$ are consistent and the *trivial interpretation* $\mathcal{I}_\emptyset$ (with $|\Delta^{\mathcal{I}_\emptyset}| = 1$ and $S^{\mathcal{I}_\emptyset} = \emptyset$, for any S) is a model of the $\mathcal{T}_i$ (a proof without those assumptions is given in the full version). Let $\mathcal{K}_i'$ be a copy of $\mathcal{K}_i$ in which all symbols S are replaced by fresh S_i, and let $\mathcal{K}_i^\Sigma$ extend $\mathcal{K}_i'$ with $S_i \sqsubseteq S$, for $S \in \Sigma$. One can show that $\mathcal{K}_1$ Σ-query entails $\mathcal{K}_2$ iff $\mathcal{K}_1 \equiv_\Sigma \mathcal{K}_1^\Sigma \cup \mathcal{K}_2^\Sigma$. ❑

That $\mathcal{I}_\emptyset \models \mathcal{K}_i$ is essential in the reduction above. Take $\mathcal{T}_1 = \{A \sqsubseteq B, A \sqsubseteq \exists R.C\}$, $\mathcal{T}_2 = \{\top \sqsubseteq B, C \sqcap B \sqsubseteq \bot\}$ and $\Sigma = \{A, B, R, C\}$. Then $\mathcal{K}_1 = (\mathcal{T}_1, \{A(a)\})$ Σ-query entails $\mathcal{K}_2 = (\mathcal{T}_2, \{A(a)\})$ but $\mathcal{K}_1 \not\equiv_\Sigma \mathcal{K}_1^\Sigma \cup \mathcal{K}_2^\Sigma$.

We now consider the relationship between inseparability and universal UCQ-solutions in knowledge exchange. Suppose $\mathcal{K}_1$ and $\mathcal{K}_2$ are KBs in disjoint signatures Σ_1 and Σ_2. Let $\mathcal{T}_{12}$ be a mapping consisting of inclusions of the form $S_1 \sqsubseteq S_2$, where the S_i are concept (or role) names in Σ_i. Then $\mathcal{K}_2$ is a universal UCQ-solution for $(\mathcal{K}_1, \mathcal{T}_{12}, \Sigma_2)$ if $\mathcal{K}_1 \cup \mathcal{T}_{12} \equiv_{\Sigma_2} \mathcal{K}_2$. Deciding the latter is called the *membership problem for universal UCQ-solutions*. For DLs $\mathcal{L}$ with role inclusions, the problem whether $\mathcal{K}_1 \cup \mathcal{T}_{12} \equiv_{\Sigma_2} \mathcal{K}_2$ is a Σ_2-query inseparability problem in $\mathcal{L}$. Conversely, we have:

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

Proof sketch. We want to decide whether $\mathcal{K}_1$ Σ-query entails $\mathcal{K}_2$. We again assume that $\mathcal{I}_\emptyset \models \mathcal{T}_i$ and use the proof of Theorem 2 (for the general case, see the full version). We may assume that $\Sigma = \mathsf{sig}(\mathcal{K}_1) \cap \mathsf{sig}(\mathcal{K}_2)$. Let $\Sigma_1 = \mathsf{sig}(\mathcal{K}_1)$. Then $\mathcal{K}_1$ Σ-query entails $\mathcal{K}_2$ iff $\mathcal{K}_1$ Σ_1-query entails $\mathcal{K}_2$. By the proof of Theorem 2, the latter is the case iff $\mathcal{K}_1$ Σ_1-query entails $\mathcal{K}_1^{\Sigma_1} \cup \mathcal{K}_2^{\Sigma_1}$. Clearly, $\mathcal{K}_1^{\Sigma_1} \cup \mathcal{K}_2^{\Sigma_1}$ Σ_1-query entails $\mathcal{K}_1$, and so the two KBs are Σ_1-query inseparable. Then $\mathcal{K}_1$ Σ-query entails $\mathcal{K}_2$ iff $\mathcal{K}_1$ is a universal UCQ-solution for $(\mathcal{K}_1' \cup \mathcal{K}_2', \mathcal{T}_{12}, \Sigma_1)$, where $\mathcal{T}_{12} = \{S_1 \sqsubseteq S, S_2 \sqsubseteq S \mid S \in \Sigma_1\}$. ❑

Semantic Characterisation

In this section, we give a semantic characterisation of KB Σ-query entailment based on an abstract notion of materialisation and finite homomorphisms between such structures.

Let $\mathcal{K}$ be a KB. An interpretation $\mathcal{I}$ is called a *materialisation* of $\mathcal{K}$ if, for all CQs $\boldsymbol{q}(\vec{x})$ and tuples $\vec{a} \subseteq \mathsf{ind}(\mathcal{K})$,

$$\mathcal{K} \models \boldsymbol{q}(\vec{a}) \quad \text{iff} \quad \mathcal{I} \models \boldsymbol{q}(\vec{a}).$$

We say that $\mathcal{K}$ is *materialisable* if it has a materialisation.

Materialisations can be used to characterise KB Σ-query entailment by means of Σ-homomorphisms. For an interpretation $\mathcal{I}$ and a signature Σ, the *Σ-types* $\boldsymbol{t}_\Sigma^\mathcal{I}(x)$ and $\boldsymbol{r}_\Sigma^\mathcal{I}(x, y)$ of $x, y \in \Delta^\mathcal{I}$ are defined by taking:

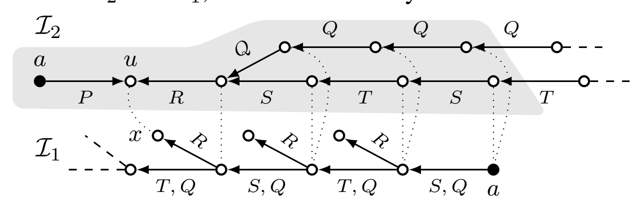
$$\boldsymbol{t}_\Sigma^\mathcal{I}(x) = \{\, \Sigma\text{-concept name } A \mid x \in A^\mathcal{I} \,\},$$
$$\boldsymbol{r}_\Sigma^\mathcal{I}(x, y) = \{\, \Sigma\text{-role } R \mid (x, y) \in R^\mathcal{I} \,\}.$$

Suppose $\mathcal{I}_i$ is a materialisation of $\mathcal{K}_i$, $i = 1, 2$. A function $h\colon \Delta^{\mathcal{I}_2} \to \Delta^{\mathcal{I}_1}$ is a *Σ-homomorphism* from $\mathcal{I}_2$ to $\mathcal{I}_1$ if, for any $a \in \mathsf{ind}(\mathcal{K}_2)$ and any $x, y \in \Delta^{\mathcal{I}_2}$,

- $h(a^{\mathcal{I}_2}) = a^{\mathcal{I}_1}$ whenever $\boldsymbol{t}_\Sigma^{\mathcal{I}_2}(a) \neq \emptyset$ or $\boldsymbol{r}_\Sigma^{\mathcal{I}_2}(a, y) \neq \emptyset$ for some $y \in \Delta^{\mathcal{I}_2}$, and
- $\boldsymbol{t}_\Sigma^{\mathcal{I}_2}(x) \subseteq \boldsymbol{t}_\Sigma^{\mathcal{I}_1}(h(x))$, $\boldsymbol{r}_\Sigma^{\mathcal{I}_2}(x, y) \subseteq \boldsymbol{r}_\Sigma^{\mathcal{I}_1}(h(x), h(y))$.

As answers to Σ-CQs are preserved under Σ-homomorphisms, $\mathcal{K}_1$ Σ-query entails $\mathcal{K}_2$ if there is a Σ-homomorphism from $\mathcal{I}_2$ to $\mathcal{I}_1$. However, the converse does not hold:

Example 4 Suppose $\mathcal{I}_2$ and $\mathcal{I}_1$ below are materialisations of KBs $\mathcal{K}_2$ and $\mathcal{K}_1$, where a is the only ABox individual:



Let $\Sigma = \{Q, R, S, T\}$. Then there is no Σ-homomorphism from $\mathcal{I}_2$ to $\mathcal{I}_1$ (as $\boldsymbol{r}_\Sigma^{\mathcal{I}_2}(a, u) = \emptyset$, we can map u to, say, x but then only the shaded part of $\mathcal{I}_2$ can be mapped Σ-homomorphically to $\mathcal{I}_1$). However, for any Σ-query $\boldsymbol{q}(\vec{x})$, $\mathcal{I}_2 \models \boldsymbol{q}(\vec{a})$ implies $\mathcal{I}_1 \models \boldsymbol{q}(\vec{a})$ as any finite subinterpretation of $\mathcal{I}_2$ can be Σ-homomorphically mapped to $\mathcal{I}_1$.

We say that $\mathcal{I}_2$ is *finitely Σ-homomorphically embeddable into* $\mathcal{I}_1$ if, for every *finite* subinterpretation $\mathcal{I}_2'$ of $\mathcal{I}_2$, there exists a Σ-homomorphism from $\mathcal{I}_2'$ to $\mathcal{I}_1$.

To prove the following theorem, one can regard any finite subinterpretation of $\mathcal{I}_2$ as a CQ whose variables are elements of $\Delta^{\mathcal{I}_2}$, with the answer variables being in $\mathsf{ind}(\mathcal{K}_2)$.

Theorem 5 *Suppose $\mathcal{K}_i$ is a consistent KB with a materialisation $\mathcal{I}_i$, $i = 1, 2$. Then $\mathcal{K}_1$ Σ-query entails $\mathcal{K}_2$ iff $\mathcal{I}_2$ is finitely Σ-homomorphically embeddable into $\mathcal{I}_1$.*

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 $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a KB and let $\mathcal{G} = (\Delta^\mathcal{G}, \cdot^\mathcal{G}, \rightsquigarrow)$ be a finite structure such that $\Delta^\mathcal{G} = \mathsf{ind}(\mathcal{K}) \cup \Omega$, for $\mathsf{ind}(\mathcal{K}) \cap \Omega = \emptyset$, $\cdot^\mathcal{G}$ is an interpretation function on $\Delta^{\mathcal{G}}$ with $A_i^{\mathcal{G}} \subseteq \Delta^{\mathcal{G}}$, $P_i^{\mathcal{G}} \subseteq \mathsf{ind}(\mathcal{K}) \times \mathsf{ind}(\mathcal{K})$, and $(\Delta^{\mathcal{G}}, \rightsquigarrow)$ is a directed graph (containing loops) with nodes $\Delta^{\mathcal{G}}$ and edges $\rightsquigarrow \subseteq \Delta^{\mathcal{G}} \times \Omega$, in which every edge $u \rightsquigarrow v$ is labelled with a set $(u,v)^{\mathcal{G}} \neq \emptyset$ of roles satisfying the condition: if $u_1 \rightsquigarrow v$ and $u_2 \rightsquigarrow v$, then $(u_1,v)^{\mathcal{G}} = (u_2,v)^{\mathcal{G}}$. We call $\mathcal{G}$ a *generating structure for* $\mathcal{K}$ if the interpretation $\mathcal{M}$ defined below is a materialisation of $\mathcal{K}$.

A *path* in $\mathcal{G}$ is a sequence $\sigma = u_0 \dots u_n$ with $u_0 \in \mathsf{ind}(\mathcal{K})$ and $u_i \rightsquigarrow u_{i+1}$ for $i < n$. Let $\mathsf{tail}(\sigma) = u_n$ and let $\mathsf{path}(\mathcal{G})$ be the set of paths in $\mathcal{G}$. The materialisation $\mathcal{M}$ is given by:

$$\Delta^{\mathcal{M}} = \mathsf{path}(\mathcal{G}), \quad a^{\mathcal{M}} = a, \ \text{ for } a \in \mathsf{ind}(\mathcal{K}),$$
$$A^{\mathcal{M}} = \{\sigma \mid \mathsf{tail}(\sigma) \in A^{\mathcal{G}}\},$$
$$P^{\mathcal{M}} = P^{\mathcal{G}} \cup \{(\sigma, \sigma u) \mid \mathsf{tail}(\sigma) \rightsquigarrow u,\ P \in (\mathsf{tail}(\sigma), u)^{\mathcal{G}}\} \cup \{(\sigma u, \sigma) \mid \mathsf{tail}(\sigma) \rightsquigarrow u,\ P^- \in (\mathsf{tail}(\sigma), u)^{\mathcal{G}}\}.$$

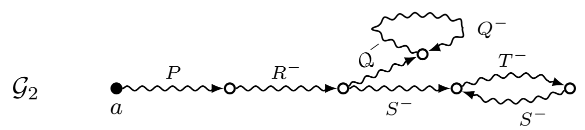
We say that a DL $\mathcal{L}$ has *finitely generated materialisations* if every $\mathcal{L}$-KB has a generating structure.

Theorem 6 *Horn-$\mathcal{ALCHI}$ and all of its fragments defined above have finitely generated materialisations. Moreover,*

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

Finite generating structures have been defined for $\mathcal{EL}$ (Lutz, Toman, and Wolter 2009), *DL-Lite* (Kontchakov et al. 2010) and more expressive Horn DLs (Eiter et al. 2008). With the exception of *DL-Lite*, however, the relation $\rightsquigarrow$ guiding the construction of materialisations was implicit. We show how the existing constructions can be converted to generating structures in the full version.

Example 7 The materialisation $\mathcal{I}_2$ from Example 4 can be generated by the structure $\mathcal{G}_2$ shown below:



For a generating structure $\mathcal{G}$ for $\mathcal{K}$ and a signature Σ, the Σ-*types* $\boldsymbol{t}_\Sigma^{\mathcal{G}}(u)$ and $\boldsymbol{r}_\Sigma^{\mathcal{G}}(u,v)$ of $u, v \in \Delta^{\mathcal{G}}$ are defined by:

$$\boldsymbol{t}_\Sigma^{\mathcal{G}}(u) = \{\, \Sigma\text{-concept name } A \mid u \in A^{\mathcal{G}} \,\},$$
$$\boldsymbol{r}_\Sigma^{\mathcal{G}}(u,v) = \begin{cases} \{\, \Sigma\text{-role } R \mid (u,v) \in R^{\mathcal{G}} \,\}, & \text{if } u, v \in \mathsf{ind}(\mathcal{K}), \\ \{\, \Sigma\text{-role } R \mid R \in (u,v)^{\mathcal{G}} \,\}, & \text{if } u \rightsquigarrow v, \\ \emptyset, & \text{otherwise}, \end{cases}$$

where $(P^-)^{\mathcal{G}}$ is the converse of $P^{\mathcal{G}}$. We also define $\bar{\boldsymbol{r}}_\Sigma^{\mathcal{G}}(u,v)$ to contain the inverses of the roles in $\boldsymbol{r}_\Sigma^{\mathcal{G}}(u,v)$; note that $\bar{\boldsymbol{r}}_\Sigma^{\mathcal{G}}(u,v)$ is not the same as $\boldsymbol{r}_\Sigma^{\mathcal{G}}(v,u)$; cf. the T^-, S^--cycle in Example 7. We write $u \rightsquigarrow^\Sigma v$ if $u \rightsquigarrow v$ and $\boldsymbol{r}_\Sigma^{\mathcal{G}}(u,v) \neq \emptyset$.

In the next section, we show that, for a DL $\mathcal{L}$ having finitely generated materialisations, the problem of checking Σ-query entailment between $\mathcal{L}$-KBs can be reduced to the problem of finding a winning strategy in a game played on the generating structures for these KBs.

Σ-Query Entailment by Games

Suppose a DL $\mathcal{L}$ has finitely generated materialisations, $\mathcal{K}_i$ is a consistent $\mathcal{L}$-KB, for $i = 1, 2$, and Σ a signature. Let $\mathcal{G}_i = (\Delta^{\mathcal{G}_i}, \cdot^{\mathcal{G}_i}, \rightsquigarrow_i)$ 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 Σ.

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 $\mathfrak{s}_i = (u_i \mapsto \sigma_i)$, 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$_1$) $\boldsymbol{t}_\Sigma^{\mathcal{G}_2}(u_i) \subseteq \boldsymbol{t}_\Sigma^{\mathcal{M}_1}(\sigma_i)$.

The game starts in a state $\mathfrak{s}_0 = (u_0 \mapsto \sigma_0)$ with $\sigma_0 = u_0$ in case $u_0 \in \mathsf{ind}(\mathcal{K}_2)$. In each round $i > 0$, player 2 challenges player 1 with some $u_i \in \Delta^{\mathcal{G}_2}$ such that $u_{i-1} \rightsquigarrow_2^\Sigma u_i$. Player 1 has to respond with a $\sigma_i \in \Delta^{\mathcal{M}_1}$ satisfying **(s$_1$)** and

(s$_2$) $\boldsymbol{r}_\Sigma^{\mathcal{G}_2}(u_{i-1}, u_i) \subseteq \boldsymbol{r}_\Sigma^{\mathcal{M}_1}(\sigma_{i-1}, \sigma_i)$.

This gives the next state $\mathfrak{s}_i = (u_i \mapsto \sigma_i)$. Note that of all the u_i only u_0 may be an ABox individual; however, there is no such a restriction on the σ_i. A *play of length* $n \geq 0$ *starting from* $\mathfrak{s}_0$ is any sequence $\mathfrak{s}_0, \dots, \mathfrak{s}_n$ of states obtained as described above. For an ordinal $\lambda \leq \omega$, we say that player 1 has a *λ-winning strategy in the game $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ starting from a state* $\mathfrak{s}_0$ if, for any play of length $i < \lambda$, which starts from $\mathfrak{s}_0$ and conforms with this strategy, and any challenge of player 2 in round $i+1$, player 1 has a response.

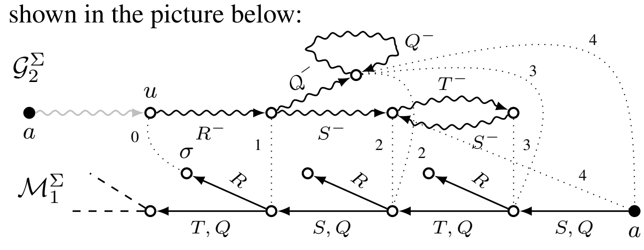
The following theorem gives a game-theoretic flavour to the criterion of Theorem 5 (see the full paper for a proof).

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

(abox) $\boldsymbol{r}_\Sigma^{\mathcal{M}_2}(a,b) \subseteq \boldsymbol{r}_\Sigma^{\mathcal{M}_1}(a,b)$, *for any* $a, b \in \mathsf{ind}(\mathcal{K}_2)$;

(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 $(u_0 \mapsto \sigma_0)$.*

Example 9 Let $\Sigma = \{Q, R, S, T\}$. Consider $\mathcal{G}_2^\Sigma$ and $\mathcal{M}_1^\Sigma$ 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 $(u \mapsto \sigma)$ is shown by dotted lines (in round 2, player 2 has two possible challenges). For a larger n, a suitable σ 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 $\mathcal{G}_2$ and $\mathcal{G}_1$. We consider four types of strategies in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$. For each type, τ, we define a game $G^\tau_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ such that, for any $u_0 \in \Delta^{\mathcal{G}_2}$, the following conditions are equivalent:

($< \omega$) for every $n < \omega$, player 1 has an n-winning strategy of type τ in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ starting from some $(u_0 \mapsto \sigma_0^n)$;

(ω) player 1 has an ω-winning strategy in $G^\tau_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ starting from some state depending on u_0 and τ.

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

Forward strategy and game $G^f_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$. We say that a λ-strategy ($\lambda \leq \omega$) for player 1 in the game $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ is *forward* if, for any play of length $i - 1 < \lambda$, which conforms with this strategy, and any challenge $u_{i-1} \rightsquigarrow^\Sigma_2 u_i$ by player 2, the response σ_i of player 1 is such that either $\sigma_{i-1}, \sigma_i \in \mathsf{ind}(\mathcal{K}_1)$ or $\sigma_i = \sigma_{i-1}v$, for some $v \in \Delta^{\mathcal{G}_1}$.

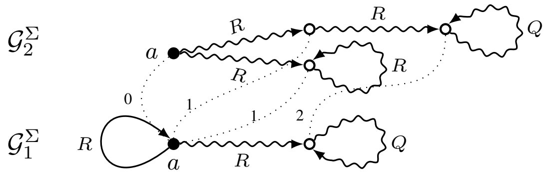
For example, if the $\mathcal{G}_i$, $i = 1, 2$, satisfy the condition

(f) the Σ-labels on $\rightsquigarrow_i$-edges contain no inverse roles,

then *every* strategy in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ is forward. This is clearly the case for *Horn-$\mathcal{ALCH}$*, *Horn-$\mathcal{ALC}$*, $\mathcal{ELH}$ and $\mathcal{EL}$, which by definition do not have inverse roles.

The existence of a forward λ-winning strategy for player 1 in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ is equivalent to the existence of such a strategy in the game $G^f_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$, which is defined similarly to $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ but with two modifications: (1) it is played on $\mathcal{G}_2$ and $\mathcal{G}_1$; and (2) the response $x_i \in \Delta^{\mathcal{G}_1}$ of player 1 to a challenge $u_{i-1} \rightsquigarrow^\Sigma_2 u_i$ must be such that either $x_{i-1}, x_i \in \mathsf{ind}(\mathcal{K}_1)$ or $x_{i-1} \rightsquigarrow_1 x_i$, and **($\mathbf{s}_1$)**–**($\mathbf{s}_2$)** hold (with $\mathcal{G}_1$ and x_i in place of $\mathcal{M}_1$ and σ_i).

Example 10 Let $\mathcal{G}_2$ and $\mathcal{G}_1$ be as shown below. Then, for any $u \in \Delta^{\mathcal{G}_2}$, there is $x \in \Delta^{\mathcal{G}_1}$ such that player 1 has an ω-winning strategy in $G^f_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ starting from $(u \mapsto x)$.



The next theorem follows from König's Lemma:

Lemma 11 *For $u_0 \in \Delta^{\mathcal{G}_2}$, condition $(< \omega)$ holds for forward strategies in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ iff (ω) holds in $G^f_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ for some state $(u_0 \mapsto x_0)$.*

$G^f_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ is a standard simulation or reachability game on finite graphs, where the existence of ω-winning strategies for player 1 follows from the existence of n-winning strategies for $n = O(|\mathcal{G}_2| \times |\mathcal{G}_1|)$, 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 Σ-query entailment is in* P *for $\mathcal{EL}$ and $\mathcal{ELH}$ KBs, and in* ExpTime *for Horn-$\mathcal{ALC}$ and Horn-$\mathcal{ALCH}$ 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 $G^b_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$. A λ-strategy for player 1 in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ is *backward* if, for any play of length $i - 1 < \lambda$, which conforms with this strategy, and any challenge $u_{i-1} \rightsquigarrow^\Sigma_2 u_i$ by player 2, the response σ_i of player 1 is the *immediate predecessor* of σ_{i-1} in $\mathcal{M}_1$ in the sense that $\sigma_{i-1} = \sigma_i w$, for some $w \in \Delta^{\mathcal{G}_1}$ (player 1 loses in case $\sigma_{i-1} \in \mathsf{ind}(\mathcal{K}_1)$). Note that, since $\mathcal{M}_1$ is tree-shaped, the response of player 1 to any different challenge $u_{i-1} \rightsquigarrow^\Sigma_2 u'_i$ must be the same σ_i; cf. Example 9.

That is why the states of the game $G^b_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ are of the form $\mathfrak{s}_i = (\Xi_i \mapsto x_i)$, where $\Xi_i \subseteq \Delta^{\mathcal{G}_2}$, $\Xi_i \neq \emptyset$, and $x_i \in \Delta^{\mathcal{G}_1}$ satisfy the following condition:

($\mathbf{s}'_1$) $\boldsymbol{t}^{\mathcal{G}_2}_\Sigma(u) \subseteq \boldsymbol{t}^{\mathcal{G}_1}_\Sigma(x_i)$, for all $u \in \Xi_i$.

The game starts in a state $\mathfrak{s}_0 = (\Xi_0 \mapsto x_0)$ such that

($\mathbf{s}'_0$) if $u \in \Xi_0 \cap \mathsf{ind}(\mathcal{K}_2)$, then $x_0 = u \in \mathsf{ind}(\mathcal{K}_1)$.

For each $i > 0$, player 2 always challenges player 1 with the set $\Xi_i = \Xi^{\rightsquigarrow}_{i-1}$, where

$$\Xi^{\rightsquigarrow} = \{v \in \Delta^{\mathcal{G}_2} \mid u \rightsquigarrow^\Sigma_2 v, \text{ for some } u \in \Xi\},$$

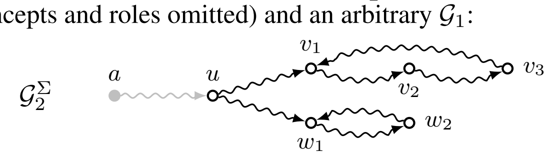
provided that it is not empty (otherwise, player 2 loses). Player 1 responds with $x_i \in \Delta^{\mathcal{G}_1}$ such that $x_i \rightsquigarrow_1 x_{i-1}$ and **($\mathbf{s}'_1$)** and the following condition hold:

($\mathbf{s}'_2$) $\boldsymbol{r}^{\mathcal{G}_2}_\Sigma(u, v) \subseteq \bar{\boldsymbol{r}}^{\mathcal{G}_1}_\Sigma(x_{i-1}, x_i)$, for all $u \in \Xi_{i-1}$, $v \in \Xi_i$.

Lemma 13 *For $u_0 \in \Delta^{\mathcal{G}_2}$, condition $(< \omega)$ holds for backward strategies in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ iff (ω) holds in $G^b_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ for some state $(\{u_0\} \mapsto x_0)$.*

Although Lemmas 11 and 13 look similar, the game $G^b_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ turns out to be more complex than $G^f_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$.

Example 14 To illustrate, consider $\mathcal{G}_2^\Sigma$ shown below (with concepts and roles omitted) and an arbitrary $\mathcal{G}_1$:



A play in $G^b_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ may proceed as: $(\{u\} \mapsto x_0)$, $(\{v_1, w_1\} \mapsto x_1)$, $(\{v_2, w_2\} \mapsto x_2)$, $(\{v_3, w_1\} \mapsto x_3)$, etc. This gives at least 6 different sets Ξ_i. But if $\mathcal{G}_2$ contained k cycles of lengths $p_1, \dots, p_k$, where p_i is the ith prime number, then the number of states in $G^b_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ could be exponential ($p_1 \times \cdots \times p_k$). In fact, we have the following:

Lemma 15 *Checking (ω) in Lemma 13 is* coNP*-hard.*

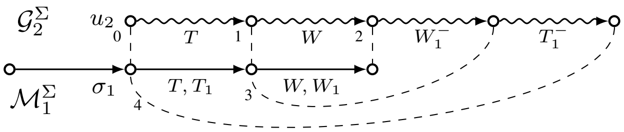
Observe that in the case of *DL-Lite$_{core}$* and *DL-Lite$_{horn}$* (which have inverse roles but no RIs), generating structures $\mathcal{G} = (\Delta^{\mathcal{G}}, \cdot^{\mathcal{G}}, \rightsquigarrow)$ can be defined so that, for any $u \in \Delta^{\mathcal{G}}$ and R, there is *at most one* v with $u \rightsquigarrow v$ and $R \in \boldsymbol{r}^{\mathcal{G}}(u, v)$ (Kontchakov et al. 2010). As a result, any n-winning strategy starting from $(u_0 \mapsto \sigma_0)$ in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ consists of a (possibly empty) backward part followed by a (possibly empty) forward part. Moreover, in the backward games for these DLs, the sets Ξ_i are always *singletons*. Thus, the number of states in the combined backward/forward games on the $\mathcal{G}_i$ is polynomial, and the existence of winning strategies can be checked in polynomial time.

Theorem 16 *Checking Σ-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 $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ is a combination of a backward strategy and a number of start-bounded strategies to be defined next.

Start-bounded strategy and game $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$. A strategy for player 1 in the game $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ 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 $\mathcal{M}_1$ that are located closer to the ABox than σ_0; the ABox individuals in $\mathcal{M}_1$ can only be used if $\sigma_0 \in \mathsf{ind}(\mathcal{K}_1)$.

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



In the game $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$, player 1 will have to guess *all* the points of $\mathcal{G}_2$ that are mapped to the same point of $\mathcal{M}_1$.

The *states* of $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ are of the form $(\Gamma_i, \Xi_i \mapsto x_i)$, $i \geq 0$, where $\Gamma_i, \Xi_i \subseteq \Delta^{\mathcal{G}_2}$, $\Xi_i \neq \emptyset$, $x_i \in \Delta^{\mathcal{G}_1}$ and **(s$'_1$)** holds. The initial state is of the form $(\emptyset, \Xi_0 \mapsto x_0)$ such that **(s$'_0$)** holds. In each round $i > 0$, player 2 challenges player 1 with some $u \rightsquigarrow^\Sigma_2 v$ such that $u \in \Xi_{i-1}$ and

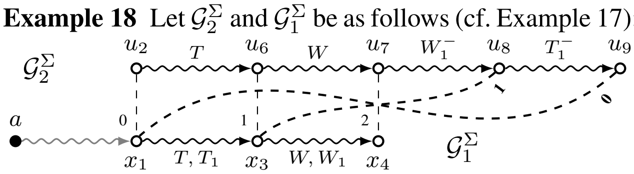
(nbk) if $v \in \Gamma_{i-1}$ then $\boldsymbol{r}^{\mathcal{G}_2}_\Sigma(u, v) \not\subseteq \bar{\boldsymbol{r}}^{\mathcal{G}_1}_\Sigma(x_{i-2}, x_{i-1})$.

Player 1 responds with either a state $(\Xi_{i-1}, \Xi_i \mapsto x_i)$ such that $x_{i-1} \rightsquigarrow_1 x_i$ (and so $x_i \notin \mathsf{ind}(\mathcal{K}_1)$) and **(s$''_2$)** holds, or a state $(\emptyset, \Xi_i \mapsto x_i)$ such that $x_{i-1}, x_i \in \mathsf{ind}(\mathcal{K}_1)$ and

(s$''_2$) $\boldsymbol{r}^{\mathcal{G}_2}_\Sigma(u, v) \subseteq \boldsymbol{r}^{\mathcal{G}_1}_\Sigma(x_{i-1}, x_i)$.

We make challenges $u \rightsquigarrow^\Sigma_2 v$, for which $u \in \Xi_{i-1}$ and **(nbk)** does not hold, 'illegitimate' because x_{i-2} can always be used as a response. Because of this, player 1 always moves 'forward' in $\mathcal{G}_1$, but has to guess appropriate sets Ξ_i in advance. Note that Γ_i is always uniquely determined by x_{i-1}, x_i and Ξ_{i-1} (and it is either Ξ_{i-1} or empty).

Example 18 Let $\mathcal{G}_2^\Sigma$ and $\mathcal{G}_1^\Sigma$ be as follows (cf. Example 17):



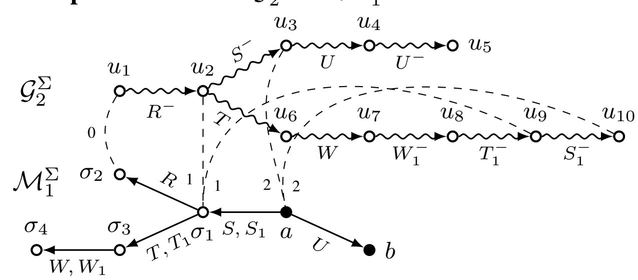
We show that player 1 has an ω-winning strategy in $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ starting from $(\emptyset, \{u_2, u_9\} \mapsto x_1)$. Player 2 challenges with $u_2 \rightsquigarrow^\Sigma_2 u_6$, and player 1 responds with $(\{u_2, u_9\}, \{u_6, u_8\} \mapsto x_3)$. Then player 2 picks $u_6 \rightsquigarrow^\Sigma_2 u_7$ and player 1 responds with $(\{u_6, u_8\}, \{u_7\} \mapsto x_4)$, where the game ends. Note the crucial guesses $\{u_2, u_9\} \mapsto x_1$ and $\{u_6, u_8\} \mapsto x_3$ made by player 1. If player 1 responded with $(\{u_2, u_9\}, \{u_6\} \mapsto x_3)$ (and failed to guess that u_8 must also be mapped to x_3), then after the challenge $u_6 \rightsquigarrow^\Sigma_2 u_7$ and response $(\{u_6\}, \{u_7\} \mapsto x_4)$), player 2 would pick $u_7 \rightsquigarrow^\Sigma_2 u_8$, to which player 1 could not respond.

Lemma 19 *For any $u_0 \in \Delta^{\mathcal{G}_2}$, condition $(< \omega)$ holds for start-bounded strategies in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ iff (ω) holds in $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ 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 $G^a_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$. An arbitrary winning strategy in the game $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ can be composed of one backward and a number of start-bounded strategies.

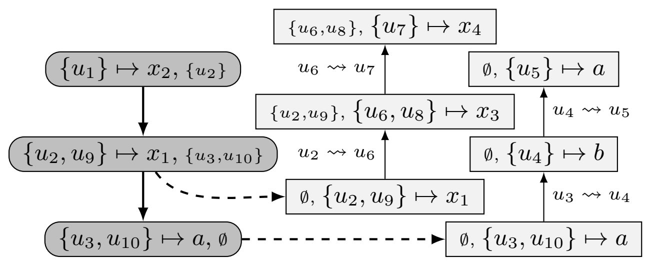
Example 20 Consider $\mathcal{G}_2^\Sigma$ and $\mathcal{M}_1^\Sigma$ shown below:



Starting from $(u_1 \mapsto \sigma_2)$, player 1 can respond to the challenges $u_1 \rightsquigarrow^\Sigma_2 u_2 \rightsquigarrow^\Sigma_2 u_3$ according to the backward strategy; the challenges $u_2 \rightsquigarrow^\Sigma_2 u_6 \rightsquigarrow^\Sigma_2 u_7 \rightsquigarrow^\Sigma_2 u_8 \rightsquigarrow^\Sigma_2 u_9$ according to the start-bounded strategy as in Example 17; the challenges $u_3 \rightsquigarrow^\Sigma_2 u_4 \rightsquigarrow^\Sigma_2 u_5$ also according to the obvious start-bounded strategy; finally, the challenge $u_9 \rightsquigarrow^\Sigma_2 u_{10}$ 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 $G^a_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ begins as $G^b_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$, but with states of the form $(\Xi_i \mapsto x_i, \Psi_i)$, $i \geq 0$, where $\Xi_i \subseteq \Delta^{\mathcal{G}_2}$ and $x_i \in \Delta^{\mathcal{G}_1}$ satisfy **(s$'_1$)** and Ψ_i is a (possibly empty) subset of $\Xi_i^{\rightsquigarrow}$, which indicates initial challenges in start-bounded games. The initial state satisfies **(s$'_0$)**. In each round $i > 0$, if $x_{i-1} \in \mathsf{ind}(\mathcal{K}_1)$ then player 2 launches the start-bounded game $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ with the initial state $(\emptyset, \Xi_{i-1} \mapsto x_{i-1})$. Otherwise, if $x_{i-1} \notin \mathsf{ind}(\mathcal{K}_1)$, player 2 has two options. First, he can challenge player 1 with the set Ψ_{i-1} (that is, similar to the backward game but with a possibly smaller Ψ_{i-1} in place of $\Xi_{i-1}^{\rightsquigarrow}$); player 1 responds to this challenge with a state $(\Xi_i \mapsto x_i, \Psi_i)$ such that $\Psi_{i-1} \subseteq \Xi_i$, $x_i \rightsquigarrow_1 x_{i-1}$ and **(s$'_2$)** holds. Second, player 2 can launch the start-bounded game $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ with the initial state $(\emptyset, \Xi_{i-1} \mapsto x_{i-1})$, where the first challenge of player 2 must be picked from $\Phi_{i-1} = \Xi_{i-1}^{\rightsquigarrow} \setminus \Psi_{i-1}$.

Example 21 We illustrate the ω-winning strategy for player 1 in $G^a_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ starting from $(\{u_1\} \mapsto x_2, \{u_2\})$, where $\mathcal{G}_2^\Sigma$ is from Example 20 and $\mathcal{G}_1^\Sigma$ looks like $\mathcal{M}_1^\Sigma$ from Example 20 (but with x_i in place of σ_i):



Lemma 22 *For any $u_0 \in \Delta^{\mathcal{G}_2}$, condition $(< \omega)$ holds for arbitrary strategies in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ iff (ω) holds in $G^a_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ for some state $(\Xi_0 \mapsto x_0, \Psi_0)$ with $u_0 \in \Xi_0$.*

Condition (ω) in the lemma above is checked in time $O(|\mathsf{ind}(\mathcal{K}_2)| \times 2^{|\Delta^{\mathcal{G}_2} \setminus \mathsf{ind}(\mathcal{K}_2)|} \times |\Delta^{\mathcal{G}_1}|)$, which can be readily seen by analysing the full game graph for $G^a_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ (similar to that in Example 21). By Theorem 6, we then obtain:

Theorem 23 *For combined complexity, Σ-query entailment is in* 2ExpTime *for Horn-$\mathcal{ALCHI}$ and Horn-$\mathcal{ALCI}$ KBs, and in* ExpTime *for DL-Lite$^{\mathcal{H}}_{horn}$ and DL-Lite$^{\mathcal{H}}_{core}$ KBs. For data complexity, these problems are all in* P.

Lower Bounds

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

Theorem 24 *For data complexity, Σ-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 φ of clauses of the form a_i and $a_i \wedge a_{i'} \to a_j$, $i, i' < j$, decide whether a_n is true in every model of φ. Consider the $\mathcal{EL}$ TBox $\mathcal{T} = \{V \sqsubseteq \exists P.(\exists R_1.V \sqcap \exists R_2.V)\}$ and an ABox $\mathcal{A}$ comprised of $F(a_n)$ and

$$P(a_i, a_i),\ R_1(a_i, a_i),\ R_2(a_i, a_i), \text{ for each clause } a_i \text{ in } \varphi,$$
$$P(a_j, c), R_1(c, a_i), R_2(c, a_{i'}), \text{ for } c = a_i \wedge a_{i'} \to a_j \text{ in } \varphi.$$

Set $\Sigma = \{F, P, R_1, R_2\}$, $\mathcal{K}_2 = (\mathcal{T}, \mathcal{A} \cup \{V(a_n)\})$ and $\mathcal{K}_1 = (\emptyset, \mathcal{A})$. Obviously, $\mathcal{K}_2$ Σ-query entails $\mathcal{K}_1$. On the other hand, the materialisation of $\mathcal{K}_2$ is (finitely) Σ-homomorphically embeddable in the materialisation of $\mathcal{K}_1$ iff φ derives a_n (see the full version for details). For *DL-Lite$_{core}$*, we take $\mathcal{T}$ to contain $V \sqsubseteq \exists P$, $\exists P^- \sqsubseteq \exists R_i$ and $\exists R_i^- \sqsubseteq V$, for $i = 1, 2$. ❑

For combined complexity, ExpTime-hardness of Σ-query inseparability for *Horn-$\mathcal{ALC}$* can be proved by reduction of the subsumption problem: we have $\mathcal{T} \models A \sqsubseteq B$ iff $(\mathcal{T}, \{A(a)\})$ and $(\mathcal{T} \cup \{A \sqsubseteq B\}, \{A(a)\})$ are $\{B\}$-query inseparable. We now establish matching lower bounds in the technically challenging cases.

Theorem 25 *For combined complexity, Σ-query entailment and inseparability are* (*i*) 2ExpTime*-hard for Horn-$\mathcal{ALCI}$ KBs and* (*ii*) ExpTime*-hard for DL-Lite$^{\mathcal{H}}_{core}$ KBs.*

Proof. The proof of (*i*) is by encoding alternating Turing machines (ATMs) with exponential tape and using the fact that AExpSpace = 2ExpTime; see, e.g. (Kozen 2006).

Let $M = (\Gamma, Q, q_0, q_1, \delta)$ be an ATM with a tape alphabet Γ, 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

$$\delta \colon (Q \setminus \{q_1\}) \times \Gamma \times \{1, 2\} \to Q \times \Gamma \times \{-1, 0, +1\},$$

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 δ 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 $\vec{w}$ iff the modified ATM M' has a run on $\vec{w}$, all branches of which are infinite.

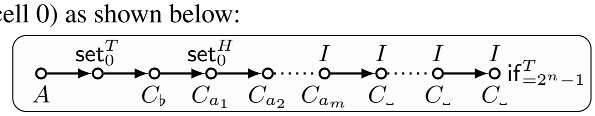
Our aim is to construct, given M and $\vec{w}$, TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ and a signature Σ such that M' has a run with only infinite branches iff the materialisation $\mathcal{M}_2$ of $(\mathcal{T}_2, \{A(c)\})$ is finitely Σ-homomorphically embeddable into the materialisation $\mathcal{M}_1$ of $(\mathcal{T}_1, \{A(c)\})$. Let f be a polynomial such that, on any input of length m, 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 $\flat \in \Gamma$. 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 $Y_{n-1}, \overline{Y}_{n-1}, \dots, Y_0, \overline{Y}_0$ of concept names as Y and assume that the TBox contains the following CIs to encode an n-bit *R-counter on Y*:

$$\overline{Y}_k \sqcap Y_{k-1} \sqcap \dots \sqcap Y_0 \sqsubseteq \forall R.(Y_k \sqcap \overline{Y}_{k-1} \sqcap \dots \sqcap \overline{Y}_0), \qquad n > k \geq 0,$$
$$\overline{Y}_i \sqcap \overline{Y}_k \sqsubseteq \forall R.\overline{Y}_i \text{ and } Y_i \sqcap \overline{Y}_k \sqsubseteq \forall R.Y_i, \quad n > i > k.$$

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

Step 1. First we encode configurations and transitions of M' using $\mathcal{T}_1$. We represent a configuration by a *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_a, for some $a \in \Gamma$: if the element with index $i + 1$ is in C_a, then the cell i is assumed to contain a in the configuration represented by the block (in particular, the element with index 1 contains $\flat$ for cell 0) as shown below:

$\mathcal{M}_1$


The first block represents the initial configuration: the input $\vec{w} = a_1 \dots a_m$ is followed by $2^n - m - 2$ blank symbols $\llcorner$ 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 $\mathcal{T}_1$:

$$A \sqsubseteq \exists P.(\mathsf{set}^T_0 \sqcap \exists P.(C_\flat \sqcap \exists P.(C_{a_1} \sqcap \mathsf{set}^H_0 \sqcap \exists P.(C_{a_2} \sqcap \exists P.(\dots \exists P.(C_{a_m} \sqcap I) \dots))))), \quad (\mathcal{T}_1\text{-}1)$$

$$\mathsf{if}^T_{<2^n-1} \sqcap I \sqsubseteq \exists P.(I \sqcap C_\llcorner), \quad (\mathcal{T}_1\text{-}2)$$

$$\mathsf{if}^T_{=2^n-1} \sqcap I \sqsubseteq Z^0_{q_0 a_1}. \quad (\mathcal{T}_1\text{-}3)$$

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^0_{qa} 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, 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^1_{qa} and Z^2_{qa} the current state and symbol in the active cell: for $q \in Q$ and $a \in \Gamma$, we add to $\mathcal{T}_1$ the CI

$$\mathsf{if}^T_{=2^n-1} \sqcap Z^0_{qa} \sqsubseteq \bigsqcap_{k=1,2} \exists P.(X_k \sqcap \exists P.(\mathsf{set}^T_0 \sqcap Z^k_{qa})), \quad (\mathcal{T}_1\text{-}4)$$

where X_1 and X_2 are two fresh concept names.

The acceptance condition for M' is enforced by means of $\mathcal{T}_2$, which uses a P-counter on a tuple T^0 for a block representing the initial configuration (a T^0-block):

$$A \sqsubseteq \exists P.\mathsf{set}^{T^0}_0, \quad (\mathcal{T}_2\text{-}1)$$

$$\mathsf{if}^{T^0}_{<2^n-1} \sqsubseteq \exists P. \quad (\mathcal{T}_2\text{-}2)$$

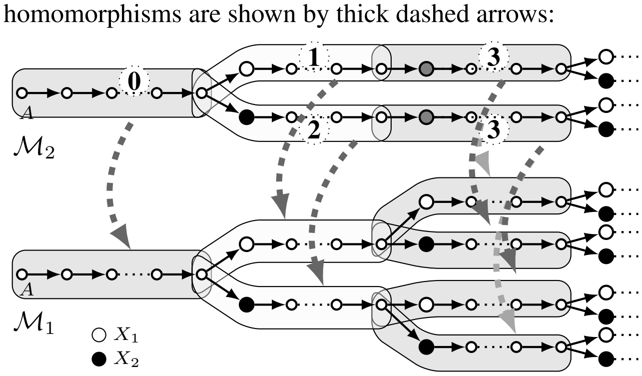
Two P-counters, on T^1 and T^2, are used for blocks representing configurations with universal states (T^1- and T^2-blocks respectively) and one P-counter, on a tuple T^3, suffices for blocks representing configurations with existential states (T^3-blocks). These blocks are arranged into an infinite 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-block, and so on. This is achieved by adding to $\mathcal{T}_2$ the following CIs:

$$\mathsf{if}^{T^k}_{=2^n-1} \sqsubseteq \bigsqcap_{j=1,2} \exists P.(X_j \sqcap \exists P.\mathsf{set}^{T^j}_0), \text{ for } k = 0, 3, \quad (\mathcal{T}_2\text{-}3)$$

$$\mathsf{if}^{T^k}_{<2^n-1} \sqsubseteq \exists P.G, \quad \text{for } k = 1, 2, 3, \quad (\mathcal{T}_2\text{-}4)$$

$$\mathsf{if}^{T^k}_{=2^n-1} \sqsubseteq \exists P.\exists P.\mathsf{set}^{T^3}_0, \quad \text{for } k = 1, 2, \quad (\mathcal{T}_2\text{-}5)$$

where G is a concept name. If $\Sigma = \{A, X_1, X_2, P\}$ then there is a unique Σ-homomorphism from the T^0-block in $\mathcal{M}_2$ to the block of the initial configuration in $\mathcal{M}_1$. Next, concepts X_1 and X_2 ensure that the T^1- and T^2-blocks are Σ-homomorphically mapped (in a unique way) into the respective blocks in $\mathcal{M}_1$, 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 $\mathcal{M}_1$, which reflects the choice in existential states; see the picture below, where possible Σ-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 $k-2$ in a *successor* configuration is $2^n - 1$. So, until the H-counter reaches $2^n - 1$, the following CIs in $\mathcal{T}_1$ propagate the state and symbol in the active cell along the blocks: for $q \in Q$, $a \in \Gamma$ and $k = 0, 1, 2$,

$$\mathsf{if}^T_{<2^n-1} \sqcap \mathsf{if}^H_{<2^n-1} \sqcap Z^k_{qa} \sqsubseteq \bigsqcap_{b \in \Gamma} \exists P.(C_b \sqcap Z^k_{qa}) \quad (\mathcal{T}_1\text{-}5)$$

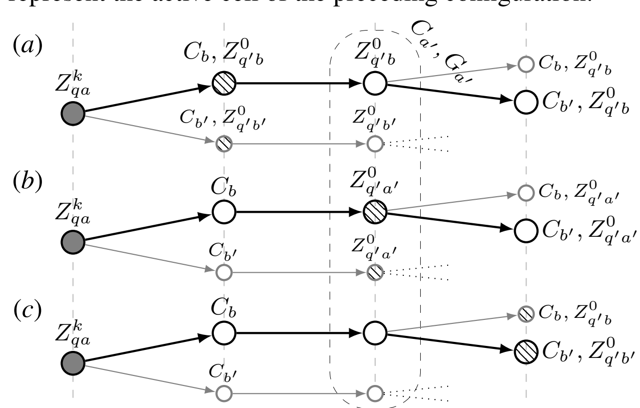
(for each $b \in \Gamma$, these CIs generate a branch in $\mathcal{M}_1$ to represent the same cell but with a different symbol, b, tentatively assigned to the cell—Step 4 will ensure that the correct branch and symbol are selected to match the cell contents in the preceding configuration). When the distance from the last head position is 2^n, the contents of the cell and the current state are changed according to δ:

$$\mathsf{if}^T_{<2^n-1} \sqcap \mathsf{if}^H_{=2^n-1} \sqcap Z^k_{qa} \sqsubseteq \bigsqcap_{b \in \Gamma} \exists P.(C_b \sqcap \Delta^k_{qa,b}), \quad (\mathcal{T}_1\text{-}6)$$

where $\delta(q, a, k) = (q', a', \sigma)$ and $\Delta^k_{qa,b}$ is the concept

$$\begin{array}{ll} \mathsf{set}^H_0 \sqcap Z^0_{q'b} \sqcap \exists P.(C_{a'} \sqcap G_{a'}), & \text{if } \sigma = -1, \\ \exists P.(C_{a'} \sqcap G_{a'} \sqcap \mathsf{set}^H_0 \sqcap Z^0_{q'a'}), & \text{if } \sigma = 0, \\ \exists P.(C_{a'} \sqcap G_{a'} \sqcap \bigsqcap_{b' \in \Gamma} \exists P.(C_{b'} \sqcap \mathsf{set}^H_0 \sqcap Z^0_{q'b'})), & \text{if } \sigma = +1 \end{array}$$

(the symbol in the active cell is changed according to the instruction, and the current state and symbol in the next active cell are then recorded in Z^0_{qa}). Since the head never visits cell 0, this happens over cells 0 to $2^n - 1$, that is, at least one element after the P-counter on T is reset to 0. These three situations are shown below, where grey and hatched nodes denote domain elements with H-values $2^n - 1$ and 0, respectively, and the domain elements in the dashed oval represent the active cell of the preceding configuration:



(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 ($\mathcal{T}_1$-5).

Step 4. The CIs ($\mathcal{T}_1$-5)–($\mathcal{T}_1$-6) generate a separate P-successor for each $b \in \Gamma$. The correct one is chosen by a finite Σ-homomorphism, h, from $\mathcal{M}_2$ to $\mathcal{M}_1$. To exclude wrong choices, we take

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

Recall that if $d_1 \in C_a^{\mathcal{M}_1}$, 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):

$$C_a \sqsubseteq D_a \sqcap \bigsqcap\nolimits_{b\in\Gamma\setminus\{a\}} G_b, \qquad (\mathcal{T}_1\text{-}7)$$

$$G_b \sqsubseteq \exists P^-.(S_b \sqcap \mathsf{set}_0^B), \qquad (\mathcal{T}\text{-}1)$$

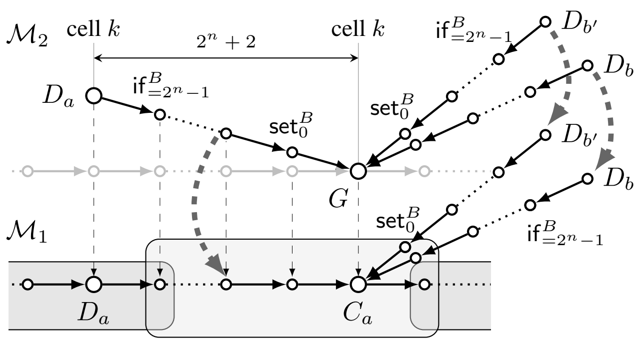
$$\mathsf{if}^B_{<2^n-1} \sqcap S_b \sqsubseteq \exists P^-.S_b, \qquad (\mathcal{T}\text{-}2)$$

$$\mathsf{if}^B_{=2^n-1} \sqcap S_b \sqsubseteq \exists P^-.D_b, \qquad (\mathcal{T}\text{-}3)$$

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 $\mathcal{M}_2$ (it represents a cell in a non-initial configuration). Then the following CI and ($\mathcal{T}$-1)–($\mathcal{T}$-3), added to $\mathcal{T}_2$, generate a D_b-block, for *each* $b \in \Gamma$ (including a):

$$G \sqsubseteq \bigsqcap\nolimits_{b\in\Gamma} G_b. \qquad (\mathcal{T}_2\text{-}6)$$

Each of the D_b-blocks in $\mathcal{M}_2$, for $b \in \Gamma$ with $b \neq a$, can be mapped by h to the respective D_b-block in $\mathcal{M}_1$. By the choice of Σ, 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 $\mathcal{M}_1$ *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 $\mathcal{M}_1$ belongs to D_a):



Note (see $\Delta^k_{qa,b}$) that the cell whose content is changed generates the additional D_a-block in $\mathcal{M}_1$ to allow the respective D_a-block from $\mathcal{M}_2$ to be mapped there.

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

(*ii*) A proof of ExpTime-hardness of Σ-query inseparability for *DL-Lite*$^{\mathcal{H}}_{core}$ 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 *DL-Lite*$^{\mathcal{H}}_{core}$ 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* 2ExpTime-*complete for Horn-$\mathcal{ALCHI}$ and Horn-$\mathcal{ALCI}$;* ExpTime-*complete for Horn-$\mathcal{ALCH}$, Horn-$\mathcal{ALC}$, DL-Lite$^{\mathcal{H}}_{horn}$ and DL-Lite$^{\mathcal{H}}_{core}$; and* P-*complete for $\mathcal{EL}$ and $\mathcal{ELH}$. For data complexity, all these problems are* P-*complete.*

In the case of *DL-Lite*$^{\mathcal{H}}_{core}$, we also obtain an ExpTime algorithm for checking the existence and computing universal UCQ-solutions. Indeed, given a KB $\mathcal{K}_1$, a target signature Σ_2 and a mapping $\mathcal{T}_{12}$, we first compute the Σ_2-ABox over $\mathsf{ind}(\mathcal{K}_1)$ that is implied by $\mathcal{K}_1$ and $\mathcal{T}_{12}$, and then check whether at least one KB $\mathcal{K}_2$ in Σ_2 with this ABox is a universal UCQ-solution (there are $\leq O(2^{|\Sigma_2|})$ such KBs). This gives an ExpTime upper bound for the non-emptiness problem for universal UCQ-solutions in *DL-Lite*$^{\mathcal{H}}_{core}$ (Arenas et al. 2013). Similarly, we can check in ExpTime whether the result of forgetting a signature in a *DL-Lite*$^{\mathcal{H}}_{core}$ KB exists.

Σ-query inseparability of *DL-Lite*$^{\mathcal{H}}_{core}$ TBoxes was known to sit between PSpace and ExpTime (Konev et al. 2011). Using the fact that witness ABoxes for *DL-Lite*$^{\mathcal{H}}_{core}$ 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 PSpace lower bound:

Theorem 27 *Σ-query inseparability of DL-Lite$^{\mathcal{H}}_{core}$ TBoxes is* ExpTime-*complete.*

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

Future Work

From a theoretical point of view, it would be of interest to investigate the complexity of Σ-query inseparability for KBs in more expressive Horn DLs (e.g., *Horn-$\mathcal{SHIQ}$*) and non-Horn DLs extending $\mathcal{ALC}$. 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 *efficient approximations* of Σ-query entailment and inseparability for KBs. The existence of a forward strategy, for example, provides a sufficient condition for Σ-query entailment for all of our DLs. Thus, one can extract a Σ-query module of a given KB $\mathcal{K}$ by exhaustively removing from $\mathcal{K}$ those inclusions and assertions α for which player 1 has a winning strategy in the game $G^f_\Sigma(\mathcal{G}_1, \mathcal{G}_2)$, where $\mathcal{G}_1$ is a generating structure for $\mathcal{K}\setminus\{\alpha\}$ and $\mathcal{G}_2$ for $\mathcal{K}$. 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.

Acknowledgements. The paper was supported by the UK EPSRC grants EP/H043594 and EP/H05099X and the EU IP Optique, n. FP7-318338.

References

Arenas, M.; Botoeva, E.; Calvanese, D.; Ryzhikov, V.; and Sherkhonov, E. 2012. Exchanging description logic knowledge bases. In *Proc. of 13th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR)*. AAAI Press.

Arenas, M.; Botoeva, E.; Calvanese, D.; and Ryzhikov, V. 2013. Exchanging OWL 2 QL knowledge bases. In *Proc. of the 23rd Int. Joint Conf. on Artificial Intelligence (IJCAI)*.

Baader, F.; Calvanese, D.; McGuinness, D.; Nardi, D.; and Patel-Schneider, P. F., eds. 2003. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press. (2nd edition, 2007).

Baader, F.; Brandt, S.; and Lutz, C. 2005. Pushing the EL envelope. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI-05)*, 364–369.

Baier, C., and Katoen, J.-P. 2007. *Principles of Model Checking*. MIT Press.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2007. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *Journal of Automated Reasoning* 39(3):385–429.

Cuenca Grau, B.; Horrocks, I.; Kazakov, Y.; and Sattler, U. 2008. Modular reuse of ontologies: theory and practice. *J. of Artificial Intelligence Research (JAIR)* 31:273–318.

Del Vescovo, C.; Parsia, B.; Sattler, U.; and Schneider, T. 2011. The modular structure of an ontology: Atomic decomposition. In *Proc. of the 22nd Int. Joint Conf. on Artificial Intelligence (IJCAI 2011)*, 2232–2237. AAAI Press.

Eiter, T.; Gottlob, G.; Ortiz, M.; and Simkus, M. 2008. Query answering in the description logic Horn-$\mathcal{SHIQ}$. In *Proc. of the 11th European Conf. on Logics in Artificial Intelligence (JELIA 2008)*, volume 5293 of *Lecture Notes in Computer Science*, 166–179. Springer.

Hustadt, U.; Motik, B.; and Sattler, U. 2005. Data complexity of reasoning in very expressive description logics. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI-05)*, 466–471.

Jiménez-Ruiz, E.; Cuenca Grau, B.; Horrocks, I.; and Berlanga Llavori, R. 2011. Supporting concurrent ontology development: Framework, algorithms and tool. *Data Knowl. Eng.* 70(1):146–164.

Kazakov, Y. 2009. Consequence-driven reasoning for Horn SHIQ ontologies. In *Proc. of the 21st Int. Joint Conf. on Artificial Intelligence (IJCAI 2009)*, 2040–2045.

Konev, B.; Kontchakov, R.; Ludwig, M.; Schneider, T.; Wolter, F.; and Zakharyaschev, M. 2011. Conjunctive query inseparability of OWL 2 QL TBoxes. In *Proc. of the 25th AAAI Conf. on Artificial Intelligence*. AAAI Press.

Konev, B.; Ludwig, M.; Walther, D.; and Wolter, F. 2012. The logical difference for the lightweight description logic EL. *Journal of Artificial Intelligence Research (JAIR)* 44:633–708.

Konev, B.; Walther, D.; and Wolter, F. 2009. Forgetting and uniform interpolation in large-scale description logic terminologies. In *Proc. of the 21st Int. Joint Conf. on Artificial Intelligence (IJCAI 2009)*, 830–835. AAAI Press.

Kontchakov, R.; Lutz, C.; Toman, D.; Wolter, F.; and Zakharyaschev, M. 2010. The combined approach to query answering in DL-Lite. In *Proc. of the 12th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2010)*. AAAI Press.

Kontchakov, R.; Wolter, F.; and Zakharyaschev, M. 2010. Logic-based ontology comparison and module extraction, with an application to DL-Lite. *Artificial Intelligence* 174:1093–1141.

Kozen, D. 2006. *Theory of Computation*. Springer.

Krötzsch, M.; Rudolph, S.; and Hitzler, P. 2013. Complexities of horn description logics. *ACM Trans. Comput. Log.* 14(1):2.

Lin, F., and Reiter, R. 1994. Forget it! In *In Proc. of the AAAI Fall Symposium on Relevance*, 154–159.

Lutz, C., and Wolter, F. 2010. Deciding inseparability and conservative extensions in the description logic EL. *Journal of Symbolic Computation* 45(2):194–228.

Lutz, C.; Seylan, I.; and Wolter, F. 2012. An automata-theoretic approach to uniform interpolation and approximation in the description logic EL. In *Proc. of the 13th Int. Conf. on Principles of Knowledge Representation (KR 2012)*. AAAI Press.

Lutz, C.; Toman, D.; and Wolter, F. 2009. Conjunctive query answering in the description logic EL using a relational database system. In *Proc. of the 21st Int. Joint Conf. on Artificial Intelligence (IJCAI-09)*, 2070–2075.

Mazala, R. 2001. Infinite games. In *Automata, Logics, and Infinite Games*, 23–42.

Nikitina, N., and Glimm, B. 2012. Hitting the sweetspot: Economic rewriting of knowledge bases. In *Proc. of the 11th Int. Semantic Web Conf. (ISWC 2012), Part I*, volume 7649 of *Lecture Notes in Computer Science*, 394–409. Springer.

Nikitina, N., and Rudolph, S. 2012. ExpExpExplosion: Uniform interpolation in general EL terminologies. In *Proc. of the 20th European Conf. on Artificial Intelligence (ECAI 2012)*, 618–623. IOS Press.

Rosati, R. 2007. On conjunctive query answering in EL. In *Proc. of the 2007 Int. Workshop on Description Logics (DL)*.

Stuckenschmidt, H.; Parent, C.; and Spaccapietra, S., eds. 2009. *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, volume 5445 of *Lecture Notes in Computer Science*. Springer.

Wang, Z.; Wang, K.; Topor, R. W.; and Pan, J. Z. 2010. Forgetting for knowledge bases in DL-Lite. *Ann. Math. Artif. Intell.* 58(1-2):117–151.

Wang, Z.; Wang, K.; and Topor, R. W. 2010. Revising general knowledge bases in description logics. In *Proc. of the 12th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2010)*. AAAI Press.