Games for Query Inseparability of Description Logic Knowledge Bases

Elena Botoeva[a], Roman Kontchakov[b], Vladislav Ryzhikov[a], Frank Wolter[c], Michael Zakharyaschev[b]

[a] *KRDB Research Centre, Free University of Bozen-Bolzano, Italy*
[b] *Department of Computer Science and Information Systems, Birkbeck, University of London, UK*
[c] *Department of Computer Science, University of Liverpool, UK*

Abstract

We consider conjunctive query inseparability of description logic knowledge bases with respect to a given signature—a fundamental problem in knowledge base versioning, module extraction, forgetting and knowledge exchange. We give a uniform game-theoretic characterisation of knowledge base conjunctive query inseparability and develop worst-case optimal decision algorithms for fragments of *Horn*-$\mathcal{ALCHI}$, including the description logics underpinning *OWL 2 QL* and *OWL 2 EL*. We also determine the data and combined complexity of deciding query inseparability. While query inseparability for all of these logics is P-complete for data complexity, the combined complexity ranges from P- to ExpTime- to 2ExpTime-completeness. We use these results to resolve two major open problems for *OWL 2 QL* by showing that TBox query inseparability and the membership problem for universal conjunctive query solutions in knowledge exchange are both ExpTime-complete for combined complexity. Finally, we introduce a more flexible notion of inseparability which compares answers to conjunctive queries in a given signature over a given set of individuals. In this case, checking query inseparability becomes NP-complete for data complexity, but the ExpTime- and 2ExpTime-completeness combined complexity results are preserved.

Keywords: Description logic, knowledge base, conjunctive query, query inseparability, games on graphs, computational complexity.

1. Introduction

A description logic (DL) knowledge base (KB) consists of a terminological box (TBox) and an assertion box (ABox). The TBox represents conceptual knowledge by providing a vocabulary for a domain of interest together with axioms that describe semantic relationships between the vocabulary items. To illustrate, consider the following toy TBox $\mathcal{T}_a$, which defines a vocabulary for the automotive industry:

$$\begin{aligned}
\mathit{Minivan} &\sqsubseteq \mathit{Automobile},\\
\mathit{Hybrid} &\sqsubseteq \mathit{Automobile},\\
\mathit{Automobile} &\sqsubseteq \exists \mathit{poweredBy}.\mathit{Engine},\\
\mathit{Hybrid} &\sqsubseteq \exists \mathit{poweredBy}.\mathit{ElectricEngine} \sqcap \exists \mathit{poweredBy}.\mathit{InternalCombustionEngine},\\
\mathit{ElectricEngine} &\sqsubseteq \mathit{Engine},\\
\mathit{InternalCombustionEngine} &\sqsubseteq \mathit{Engine}.
\end{aligned}$$

The first two axioms say that minivans and hybrids are automobiles, the third one claims that every automobile is powered by an engine, and the fourth axiom states that every hybrid is powered by an electric engine and also by an internal combustion engine. Thus, the TBox introduces, among others, the concept names (sets) *Minivan*, *Automobile* and *Engine*, states that the concept *Minivan* is subsumed by the concept *Automobile* and uses the role name (binary

Email addresses: botoeva@inf.unibz.it (Elena Botoeva), roman@dcs.bbk.ac.uk (Roman Kontchakov), ryzhikov@inf.unibz.it (Vladislav Ryzhikov), wolter@liverpool.ac.uk (Frank Wolter), michael@dcs.bbk.ac.uk (Michael Zakharyaschev)

relation) *poweredBy* to say that automobiles are powered by engines. TBoxes, often called ontologies, are represented in many applications using the syntax of the Web Ontology Language OWL 2 (`www.w3.org/TR/owl2-overview`).

The ABox of a knowledge base is a set of facts storing data about the concept and role names introduced in the TBox. As an example ABox in the automotive domain, we will use the following set of assertions:

$$\mathcal{A}_a = \{\, \mathit{Hybrid}(\mathit{toyota\_highlander}),\ \mathit{Minivan}(\mathit{toyota\_highlander}),\ \mathit{Minivan}(\mathit{nissan\_note}),\ \mathit{poweredBy}(\mathit{nissan\_note}, \mathit{hr15de}),\ \mathit{InternalCombustionEngine}(\mathit{hr15de})\,\}.$$

Typical applications of KBs in modern information systems use the semantics of the concepts and roles in the TBox to enable the user to query the data in the ABox. This is particularly useful if the data is incomplete or comes from heterogenous data sources, which is the case, for example, in linked data applications [1] and large-scale data integration projects [2, 3], or if the data comprises the web content gathered by search engines using semantic markup [4].

As the data may be incomplete, the open world assumption is adopted when querying a KB $\mathcal{K}$: a tuple $\boldsymbol{a}$ of individuals from $\mathcal{K}$ is a (certain) answer to a query $\boldsymbol{q}$ over $\mathcal{K}$ if $\boldsymbol{q}(\boldsymbol{a})$ is true in every model of $\mathcal{K}$. Since general first-order queries are undecidable under the open-world semantics, the basic and most important querying instrument is conjunctive queries (CQs), which are ubiquitous in relational database systems and form the core of the Semantic Web query language SPARQL (`www.w3.org/TR/sparql11-query`). In our context, a CQ $\boldsymbol{q}(\boldsymbol{x})$ is a first-order formula $\exists \boldsymbol{y}\, \varphi(\boldsymbol{x}, \boldsymbol{y})$ such that $\varphi(\boldsymbol{x}, \boldsymbol{y})$ is a conjunction of atoms of the form $A(z_1)$ or $P(z_1, z_2)$, for a concept name A, a role name P, and variables z_1, z_2 from $\boldsymbol{x}, \boldsymbol{y}$.[1] For example, to find minivans powered by electric engines, one can use the CQ

$$\boldsymbol{q}(x) = \exists y\, (\mathit{Minivan}(x) \land \mathit{poweredBy}(x, y) \land \mathit{ElectricEngine}(y)),$$

with *toyota_highlander* being the only certain answer to $\boldsymbol{q}(x)$ over $(\mathcal{T}_a, \mathcal{A}_a)$.

The problem of answering CQs over KBs has been the focus of significant research in the DL community: deep complexity results have been obtained for a broad range of DLs (see below), new DLs have been introduced with tractable (in data complexity) query answering [5, 6], a variety of query answering techniques have been invented [6, 7] and implemented in a number of powerful software systems (see, e.g., [8] and references therein).

Apart from developing query answering techniques, a major research problem is KB engineering and maintenance. In fact, with typically large data and often complex and tangled ontologies, tool support for transforming and comparing KBs is becoming indispensable for applications. To begin with, KBs are never static entities. Like most software artefacts, they are updated to incorporate new information, and distinct versions are introduced for different applications. Thus, developing support for KB versioning has become an important research problem [9, 10]. As dealing with a large and semantically tangled KB can be costly, one may want to extract from it a smaller module that is indistinguishable from the whole KB as far as the given application is concerned [11]. Another technique for extracting relevant information is forgetting, where the task is to replace a given KB with a new one, which uses only those concept and role names that are needed by the application but still provides the same information about those names as the original KB [12, 13]. Finally, the vocabulary of a given KB may not be convenient for a new application. In this case, similarly to data exchange in databases [14]—where data structured under a source schema is converted to data under a target schema—one may want to transform a KB in a source signature to a KB given in a more useful target signature and representing the original KB in an accurate way. This task is known as knowledge exchange [15, 16].

In this article, we investigate a relationship between KBs that is fundamental for all such tasks if querying the data via CQs is the main application. Let Σ be a relational signature consisting of a finite set of concept and role names. We say that KBs $\mathcal{K}_1$ and $\mathcal{K}_2$ are 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 in the KBs, Σ-query inseparability does not necessarily imply logical equivalence: 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. Thus, if KBs are used for purposes other than querying data via CQs, then different notions of inseparability are required. We now discuss the applications of Σ-query inseparability for the tasks mentioned above in more detail.

[1] Since we consider Horn DLs, the results of this article actually apply to disjunctions (or unions) of CQs (known as UCQs). For simplicity, however, we consider CQs only.

Versioning. Version control systems for KBs provide a range of operations including, for example, computing the relevant differences between KBs, merging KBs and recovering KBs. All these operations rely on checking whether two versions, $\mathcal{K}_1$ and $\mathcal{K}_2$, of a KB are indistinguishable from the application point of view. If that application is querying the data via CQs in a given relational signature Σ, then $\mathcal{K}_1$ and $\mathcal{K}_2$ should be regarded as indistinguishable just in case they give the same answers to CQs formulated in Σ. Thus, the basic task for a query-centric approach to KB versioning is to check whether $\mathcal{K}_1 \equiv_\Sigma \mathcal{K}_2$.

Modularisation. Modularisation and module extraction are major research topics in ontology engineering and maintenance. In module extraction, the problem is to find a (small) subset of the axioms of a given large KB that is indistinguishable from the KB with respect to the intended application. If that application is querying a KB $\mathcal{K}$ using CQs in a relational signature Σ, then the problem is to find a small Σ-query module of $\mathcal{K}$, that is, a KB $\mathcal{K}' \subseteq \mathcal{K}$ with $\mathcal{K}' \equiv_\Sigma \mathcal{K}$. Note that one can extract a minimal Σ-query module from a KB using a polynomial-time algorithm with the Σ-query inseparability check as an oracle (see, e.g., [17]). To illustrate the notion of Σ-query module, consider the automotive knowledge base $\mathcal{K}_a = (\mathcal{T}_a, \mathcal{A}_a)$ defined above and the relational signature $\Sigma_m = \{\mathit{Automobile}, \mathit{Engine}, \mathit{poweredBy}\}$. Then $\mathcal{K}_m = (\mathcal{T}_m, \mathcal{A}_m)$ is a Σ_m-query module of $\mathcal{K}_a$, where

$$\mathcal{T}_m = \{\mathit{Minivan} \sqsubseteq \mathit{Automobile},\ \mathit{Automobile} \sqsubseteq \exists \mathit{poweredBy}.\mathit{Engine},\ \mathit{InternalCombustionEngine} \sqsubseteq \mathit{Engine}\},$$
$$\mathcal{A}_m = \{\mathit{Minivan}(\mathit{toyota\_highlander}),\ \mathit{Minivan}(\mathit{nissan\_note}),\ \mathit{poweredBy}(\mathit{nissan\_note}, \mathit{hr15de}),\ \mathit{InternalCombustionEngine}(\mathit{hr15de})\}.$$

Knowledge Exchange. In knowledge exchange, we want to transform a KB $\mathcal{K}_1$ in a relational signature Σ_1 to a KB $\mathcal{K}_2$ in a new signature Σ_2 connected to Σ_1 via a declarative mapping specification given by a TBox $\mathcal{T}_{12}$. Such mapping specifications between KBs are also known as ontology alignments or ontology matchings and have been studied extensively [18]. If, as above, we are interested in querying data via CQs, then the target KB $\mathcal{K}_2$ should be a sound and complete representation of $\mathcal{K}_1$ with respect to answers to CQs, and so 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 CQ-solution. To illustrate, consider again the knowledge base $\mathcal{K}_a = (\mathcal{T}_a, \mathcal{A}_a)$ and let $\mathcal{T}_{ae}$ connect the relational signature Σ_a of $\mathcal{K}_a$ to $\Sigma_e = \{\mathit{Car}, \mathit{HybridCar}, \mathit{ElectricMotor}, \mathit{Motor}, \mathit{hasMotor}\}$ by means of the following axioms:

$$\mathit{Automobile} \sqsubseteq \mathit{Car}, \qquad \mathit{Hybrid} \sqsubseteq \mathit{HybridCar}, \qquad \mathit{poweredBy} \sqsubseteq \mathit{hasMotor},$$
$$\mathit{Engine} \sqsubseteq \mathit{Motor}, \qquad \mathit{ElectricEngine} \sqsubseteq \mathit{ElectricMotor}.$$

Then $\mathcal{K}_e = (\mathcal{T}_e, \mathcal{A}_e)$ is a universal CQ-solution, where

$$\mathcal{T}_e = \{\mathit{ElectricMotor} \sqsubseteq \mathit{Motor},\ \mathit{Car} \sqsubseteq \exists \mathit{hasMotor}.\mathit{Motor},\ \mathit{HybridCar} \sqsubseteq \mathit{Car} \sqcap \exists \mathit{hasMotor}.\mathit{ElectricMotor}\},$$
$$\mathcal{A}_e = \{\mathit{HybridCar}(\mathit{toyota\_highlander}),\ \mathit{Car}(\mathit{nissan\_note}),\ \mathit{hasMotor}(\mathit{nissan\_note}, \mathit{hr15de}),\ \mathit{Motor}(\mathit{hr15de})\}.$$

Forgetting. A KB $\mathcal{K}'$ is said to result from forgetting a relational signature Σ in a KB $\mathcal{K}$ if $\mathcal{K}' \equiv_{\mathsf{sig}(\mathcal{K}) \setminus \Sigma} \mathcal{K}$ and $\mathsf{sig}(\mathcal{K}') \subseteq \mathsf{sig}(\mathcal{K}) \setminus \Sigma$, where $\mathsf{sig}(\mathcal{K})$ is the relational signature of $\mathcal{K}$. Thus, the result of forgetting Σ does not use Σ and gives the same answers to CQs without symbols in Σ as $\mathcal{K}$. The result of forgetting is also called a uniform interpolant for $\mathcal{K}$ with respect to $\mathsf{sig}(\mathcal{K}) \setminus \Sigma$. Forgetting is of interest in a number of scenarios. Typically, when reusing an existing KB in a new application, only a small number of its symbols is relevant, and so instead of reusing the whole KB, one can take a potentially smaller KB resulting from forgetting the extraneous symbols. Forgetting can also be used for predicate hiding: if a KB is to be published, but some part of it has to be concealed from the public, then this part can be removed by forgetting its symbols [19]. Finally, forgetting can be used for KB summary: the result of forgetting often provides a smaller and more focused KB that summarises what the original KB says about the retained symbols, potentially facilitating comprehension. To illustrate, the KB $\mathcal{K}_f = (\mathcal{T}_f, \mathcal{A}_f)$ results from forgetting $\Sigma_f = \{\mathit{Minivan}, \mathit{Hybrid}, \mathit{ElectricEngine}, \mathit{InternalCombustionEngine}\}$ in $\mathcal{K}_a$, where

$$\mathcal{T}_f = \{\mathit{Automobile} \sqsubseteq \exists \mathit{poweredBy}.\mathit{Engine}\},$$
$$\mathcal{A}_f = \{\mathit{Automobile}(\mathit{toyota\_highlander}),\ \mathit{Automobile}(\mathit{nissan\_note}),\ \mathit{poweredBy}(\mathit{nissan\_note}, \mathit{hr15de}),\ \mathit{Engine}(\mathit{hr15de})\}.$$

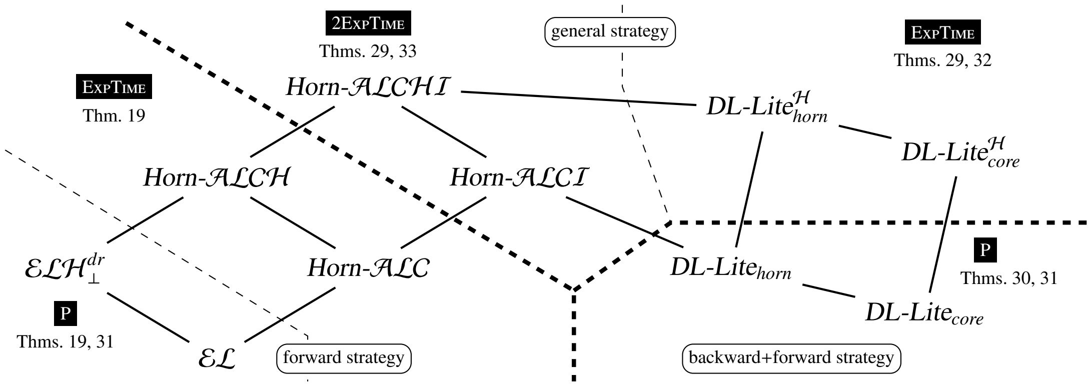


Figure 1: Summary of the combined complexity results.

In this article, we develop worst-case optimal algorithms deciding Σ-query inseparability of KBs given in various fragments of the description logic *Horn-$\mathcal{ALCHI}$* [20], which include $DL\text{-}Lite^{\mathcal{H}}_{core}$ [6, 21] and $\mathcal{ELH}^{dr}_{\bot}$ [22] underlying the OWL 2 profiles *OWL 2 QL* and *OWL 2 EL* (`www.w3.org/TR/owl2-profiles`). The algorithms are based on two characterisations of Σ-query inseparability, one of which is model-theoretic and the other game-theoretic. The former characterises Σ-query inseparability in terms of partial Σ-homomorphisms between materialisations, that is, interpretations $\mathcal{M}$ of KBs $\mathcal{K}$ such that the certain answers to any CQ $\boldsymbol{q}$ over $\mathcal{K}$ coincide with the answers to CQ $\boldsymbol{q}$ over $\mathcal{M}$. Any *Horn-$\mathcal{ALCHI}$* KB has a materialisation. While materialisations can be infinite, we show that one can always compute a finite *generating structure* from which a materialisation is obtained by unravelling. We then develop a game-theoretic machinery for checking the existence of partial Σ-homomorphisms between materialisations by playing two-player games on the corresponding finite generating structures. Thus, our algorithms consist of two components: computing finite generating structures for the given KBs and deciding the existence of winning strategies for the games on these structures.

We use the constructed algorithms to obtain optimal upper bounds for the data and combined complexity of deciding Σ-query inseparability for KBs given in all of the DLs mentioned above. Σ-query inseparability turns out to be P-complete for data complexity, which matches the 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 Fig. 1. Most interesting are ExpTime-completeness of $DL\text{-}Lite^{\mathcal{H}}_{core}$ and 2ExpTime-completeness of *Horn-$\mathcal{ALCI}$*, which contrast with NP- and ExpTime-completeness of CQ evaluation for these logics. For *DL-Lite* without role inclusions, $\mathcal{EL}$ and $\mathcal{ELH}^{dr}_{\bot}$, 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. The matching lower bounds are established by a (rather involved) encoding of suitable alternating Turing machines.

We apply our complexity results for Σ-query inseparability to resolve two important open problems. First, we show that, in knowledge exchange, the membership problem for universal CQ-solutions for $DL\text{-}Lite^{\mathcal{H}}_{core}$ KBs is ExpTime-complete for combined complexity, which settles an open question of [23], where only PSpace-hardness was established. Second, we show that deciding Σ-query inseparability of $DL\text{-}Lite^{\mathcal{H}}_{core}$ TBoxes (for arbitrary ABoxes) is ExpTime-complete, which closes the PSpace–ExpTime gap that was left open by Konev et al. [24].

In the definition of Σ-query inseparability above, we took account of *all* tuples of individuals in the KBs that could be certain answers to CQs. In some applications, however, we may be interested only in a specific set of individuals over which the certain answers should be compared. Let Γ be an individual signature consisting of a finite set of individual names. For KBs $\mathcal{K}_1, \mathcal{K}_2$ and a relational signature Σ, we say that $\mathcal{K}_1$ and $\mathcal{K}_2$ are (Σ, Γ)-query inseparable if any CQ formulated in Σ has the same certain answers among the individuals in Γ over both $\mathcal{K}_1$ and $\mathcal{K}_2$, in which case we write $\mathcal{K}_1 \equiv_{\Sigma,\Gamma} \mathcal{K}_2$. Clearly, if Γ contains all individuals in $\mathcal{K}_1 \cup \mathcal{K}_2$, then (Σ, Γ)-query inseparability implies Σ-query inseparability. (Σ, Γ)-query inseparability can be used to refine Σ-query inseparability as a foundation for versioning, modularisation, forgetting and knowledge exchange.

For instance, a KB $\mathcal{K}'$ is a (Σ, Γ)-query module of a KB $\mathcal{K}$ if $\mathcal{K}' \subseteq \mathcal{K}$ and $\mathcal{K}' \equiv_{\Sigma,\Gamma} \mathcal{K}$. Consider again the automotive ontology $\mathcal{K}_a = (\mathcal{T}_a, \mathcal{A}_a)$ and the relational signature $\Sigma_m = \{\mathit{Automobile}, \mathit{Engine}, \mathit{poweredBy}\}$. Unlike our example illustrating Σ-query modules, we now restrict the individual signature to $\Gamma_m = \{\mathit{toyota\_highlander}, \mathit{nissan\_note}\}$ thereby leaving out *hr15de* from the set of individuals considered. Then the KB $\mathcal{K}'_m = (\mathcal{T}'_m, \mathcal{A}'_m)$ is a (Σ_m, Γ_m)-query module of $\mathcal{K}_a$, where

$$\mathcal{T}'_m = \{\mathit{Minivan} \sqsubseteq \mathit{Automobile}, \mathit{Automobile} \sqsubseteq \exists \mathit{poweredBy}.\mathit{Engine}\},$$
$$\mathcal{A}'_m = \{\mathit{Minivan}(\mathit{toyota\_highlander}), \mathit{Minivan}(\mathit{nissan\_note})\}.$$

Thus, the restriction of the individual signature removes the two assertions with *hr15de* from $\mathcal{A}_m$ as well as an axiom from $\mathcal{T}_m$.

Similarly, a KB $\mathcal{K}'$ results from forgetting (Σ, Γ) in a KB $\mathcal{K}$ if $\mathcal{K}' \equiv_{\mathsf{sig}(\mathcal{K})\setminus\Sigma, \mathsf{ind}(\mathcal{K})\setminus\Gamma} \mathcal{K}$, $\mathsf{sig}(\mathcal{K}') \subseteq \mathsf{sig}(\mathcal{K}) \setminus \Sigma$ and $\mathsf{ind}(\mathcal{K}') \subseteq \mathsf{ind}(\mathcal{K}) \setminus \Gamma$, where $\mathsf{ind}(\mathcal{K})$ is the set of individuals in the ABox of $\mathcal{K}$. In this case, for $\Gamma_f = \{\mathit{hr15de}\}$, the KB $\mathcal{K}'_f = (\mathcal{T}'_f, \mathcal{A}'_f)$ results from forgetting (Σ_f, Γ_f) in $\mathcal{K}_a$, where

$$\mathcal{T}'_f = \{\mathit{Automobile} \sqsubseteq \exists \mathit{poweredBy}.\mathit{Engine}\},$$
$$\mathcal{A}'_f = \{\mathit{Automobile}(\mathit{toyota\_highlander}), \mathit{Automobile}(\mathit{nissan\_note})\}.$$

In knowledge exchange, the refined notion of query inseparability can be used to represent a more flexible knowledge exchange model, which allows additional individuals in the target KB. These 'anonymous' individuals are similar to nulls in the standard approaches to incomplete databases [25]. Thus, we say that a KB $\mathcal{K}_2$ with a relational signature Σ_2 is a universal CQ-solution with nulls for a KB $\mathcal{K}_1$ and a mapping specification $\mathcal{T}_{12}$ if $\mathcal{K}_1 \cup \mathcal{T}_{12} \equiv_{\Sigma_2, \mathsf{ind}(\mathcal{K}_1)} \mathcal{K}_2$ (here, the individuals in $\mathsf{ind}(\mathcal{K}_2) \setminus \mathsf{ind}(\mathcal{K}_1)$ play the role of nulls). To illustrate, we consider again the knowledge exchange example given above with the same Σ_e and $\mathcal{T}_{ae}$. Observe first that $\mathcal{K}_e$ is also a universal CQ-solution with nulls. On the other hand, there are universal CQ-solutions with nulls that are not universal CQ-solutions. To illustrate, let m_1 be a fresh individual name. Then $\mathcal{K}'_e = (\emptyset, \mathcal{A}'_e)$ is a universal CQ-solution with nulls for $\mathcal{K}_a$ and $\mathcal{T}_{ae}$, where

$$\begin{aligned}\mathcal{A}'_e = \{&\mathit{HybridCar}(\mathit{toyota\_highlander}), \mathit{Car}(\mathit{toyota\_highlander}),\\ &\mathit{hasMotor}(\mathit{toyota\_highlander}, m_1), \mathit{ElectricMotor}(m_1), \mathit{Motor}(m_1),\\ &\mathit{Car}(\mathit{nissan\_note}), \mathit{hasMotor}(\mathit{nissan\_note}, \mathit{hr15de}), \mathit{Motor}(\mathit{hr15de})\}.\end{aligned}$$

Intuitively, $\mathcal{A}'_e$ is a materialisation of all consequences of $\mathcal{K}_a \cup \mathcal{T}_{ae}$ in the relational signature Σ_e and, among individuals of $\mathcal{K}_a$, it clearly gives rise to the same answers to all CQs formulated in Σ_e (the additional individual, m_1, is not counted when comparing the CQ answers). The interested reader is referred to [23] for more explanations on the advantages of this notion.

We extend our algorithms deciding Σ-inseparability to algorithms deciding (Σ, Γ)-inseparability and investigate the data and combined complexity of the problem for KBs given in the same fragments of *Horn*-$\mathcal{ALCHI}$ as before. In contrast to Σ-query inseparability, which is P-complete for data complexity for all of those fragments, deciding (Σ, Γ)-query inseparability turns out to be NP-complete for data complexity. (In fact, it is NP-hard already for KBs without TBoxes since (Σ, Γ)-query inseparability is then equivalent to the problem of deciding the existence of a homomorphism from one relational structure to another, which is known to be NP-hard.) For combined complexity, (Σ, Γ)-query inseparability is exactly as hard as Σ-query inseparability whenever it is already NP-hard.

The remainder of the article is structured as follows. In Section 2, we introduce the syntax and semantics of the DLs considered in this article. In Section 3, we provide a model-theoretic characterisation of conjunctive query inseparability based on materialisations and introduce finite generating structures from which materialisations are obtained by unravelling. We also analyse our algorithms computing generating structures and their relevant properties, depending on the DLs considered. In Section 4, we develop games on generating structures and the corresponding algorithms for deciding inseparability, using which we obtain complexity upper bounds. Section 5 is devoted to proving matching lower complexity bounds. In Section 6, we refine Σ-inseparability by considering restricted sets of individuals in KBs and, in Section 7, we discuss related work and how our results can be (or have been) applied to solve open problems in knowledge exchange, TBox inseparability and for the comparison of OBDA (ontology-based data access) specifications. We conclude with a discussion of future work in Section 8.

2. *Horn-$\mathcal{ALCHI}$* and its Fragments

In this article, we investigate Σ- and (Σ, Γ)-query inseparability of KBs given in DLs that are Horn fragments[2] of $\mathcal{ALCHI}$. To define these DLs, we fix sequences of *individual names* a_i, *concept names* A_i, and *role names* P_i, for $i < \omega$. 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* are defined by the grammar

$$C \;::=\; A_i \;\mid\; \top \;\mid\; \bot \;\mid\; \neg C \;\mid\; C_1 \sqcap C_2 \;\mid\; C_1 \sqcup C_2 \;\mid\; \exists R.C \;\mid\; \forall R.C, \qquad (\mathcal{ALCI})$$

where R is a role. $\mathcal{ALC}$*-concepts* are those $\mathcal{ALCI}$-concepts that do not contain inverse roles. $\mathcal{ALCI}$*-TBoxes* and $\mathcal{ALC}$*-TBoxes* are finite sets of *concept inclusions* of the form

$$C_1 \sqsubseteq C_2,$$

where the C_i are $\mathcal{ALCI}$- or, respectively, $\mathcal{ALC}$-concepts. $\mathcal{ALCHI}$*-TBoxes* are finite sets of concept inclusions in $\mathcal{ALCI}$ and *role inclusions* of the form

$$R_1 \sqsubseteq R_2,$$

where the R_i are roles. $\mathcal{ALCH}$*-TBoxes* are $\mathcal{ALCHI}$-TBoxes that do not contain occurrences of inverse roles.

The DLs in the $\mathcal{EL}$ and *DL-Lite* families are sub-Boolean fragments of $\mathcal{ALCHI}$. $\mathcal{EL}$*-concepts* are defined by the grammar

$$C \;::=\; A_i \;\mid\; \top \;\mid\; C_1 \sqcap C_2 \;\mid\; \exists P_i.C. \qquad (\mathcal{EL})$$

In other words, they are $\mathcal{ALC}$-concepts without $\bot$, $\sqcup$, $\neg$ and $\forall P_i.C$. Note that $\mathcal{EL}$ does not have inverse roles. $\mathcal{EL}$*-TBoxes* are finite sets of concepts inclusions in $\mathcal{EL}$. $\mathcal{ELH}^{dr}_\bot$ is an extension of $\mathcal{EL}$ with $\bot$, role inclusions and domain and range restrictions. Thus, $\mathcal{ELH}^{dr}_\bot$*-concepts* are defined similarly to $\mathcal{EL}$-concepts but can also use $\bot$, and $\mathcal{ELH}^{dr}_\bot$-TBoxes consist of a finite number of $\mathcal{ELH}^{dr}_\bot$-concept inclusions, role inclusions (without inverse roles), and *range restrictions* of the form

$$\top \sqsubseteq \forall P_i.C$$

(domain restrictions are expressible by means of concept inclusions $\exists P_i.\top \sqsubseteq C$). Clearly, $\mathcal{EL}$ and $\mathcal{ELH}^{dr}_\bot$ are sub-languages of $\mathcal{ALC}$ and $\mathcal{ALCH}$, respectively.

Basic concepts in *DL-Lite* are defined by the following grammar:

$$B \;::=\; A_i \;\mid\; \top \;\mid\; \bot \;\mid\; \exists R.\top, \qquad (\textit{DL-Lite})$$

where R is a (possibly inverse) role. Existential quantifiers $\exists R.\top$ are called *unqualified*, and we usually write $\exists R$ instead of $\exists R.\top$. *DL-Lite$_{core}$-TBoxes* are finite sets of concept inclusions of the form

$$B_1 \sqsubseteq B_2 \qquad \text{and} \qquad B_1 \sqcap B_2 \sqsubseteq \bot,$$

where the B_i are basic concepts. *DL-Lite$_{horn}$-TBoxes* consist of a finite number of concept inclusions of the form

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

DL-Lite$^{\mathcal{H}}_{core}$- and *DL-Lite*$^{\mathcal{H}}_{horn}$*-TBoxes* contain, in addition, a finite number of role inclusions and role disjointness axioms of the form $R_1 \sqcap R_2 \sqsubseteq \bot$. Note that, unlike $\mathcal{EL}$ and $\mathcal{ELH}^{dr}_\bot$, the *DL-Lite* logics do have inverse roles.

To introduce the Horn fragments of the DLs with the Booleans operators, we require the following (standard) recursive definition [5, 26]. We say that a concept C occurs positively in C and, if C occurs positively (negatively) in C', then

[2] Strictly speaking, *DL-Lite*$^{\mathcal{H}}_{core}$ and *DL-Lite*$^{\mathcal{H}}_{horn}$ are not *fragments* of $\mathcal{ALCHI}$ because it does not have role disjointness constraints. However, these constraints play no essential part in our constructions, and the techniques we develop for $\mathcal{ALCHI}$ are also applicable to the logics in the *DL-Lite* family.

– C occurs positively (respectively, negatively) in $C' \sqcup D$, $C' \sqcap D$, $\exists R.C'$, $\forall R.C'$, $D \sqsubseteq C'$, and

– C occurs negatively (respectively, 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 $\mathcal{T}$. Clearly, the $\mathcal{EL}$- and *DL-Lite*-TBoxes are Horn by definition. For any other DL $\mathcal{L}$ (e.g., $\mathcal{ALCHI}$), only Horn $\mathcal{L}$-TBoxes are allowed in the DL *Horn-*$\mathcal{L}$.

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) $\mathcal{K} = (\mathcal{T}, \mathcal{A})$.

A *relational signature* is any non-empty finite set of concept and role names. An *individual signature* is a (possibly empty) finite set of individual names. We usually denote a relational signature by Σ, an individual signature by Γ, and sometimes call the pair (Σ, Γ) simply a *signature*. The relational signature of a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, which consists of the concept and role names occurring in $\mathcal{K}$, is denoted by $\mathsf{sig}(\mathcal{K})$. The individual signature of $\mathcal{K}$, comprising the individual names in $\mathcal{A}$, is denoted by $\mathsf{ind}(\mathcal{K})$. In this article, we are not interested in KBs with empty ABoxes, and so both $\mathsf{sig}(\mathcal{K})$ and $\mathsf{ind}(\mathcal{K})$ are non-empty by definition. By a Σ-concept, Σ-role, Σ-ABox, etc. we understand any concept, role, ABox, etc. all of whose concept and role names are taken from Σ.

Let (Σ, Γ) be a signature. In our interpretations, we adopt the *standard name assumption* in the sense that every individual name $a \in \Gamma$ is interpreted by itself. A (Σ, Γ)-*interpretation* is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}} \supseteq \Gamma$ is a non-empty set, the *domain* of $\mathcal{I}$, and $\cdot^{\mathcal{I}}$ is an *interpretation function* that assigns a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ to every concept name A and a binary relation $P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ to every role name P in such a way that $A^{\mathcal{I}} = \emptyset$ and $P^{\mathcal{I}} = \emptyset$, for any $A \notin \Sigma$ and $P \notin \Sigma$. (Note that only the individual names from Γ are interpreted in $\mathcal{I}$ and, although the list of individual names is countably infinite, $\Delta^{\mathcal{I}}$ may be finite. Note also that the concept and role names outside Σ are always interpreted as $\emptyset$.) When we use the terms 'interpretation', 'Σ-interpretation' or 'Γ-interpretation' without specifying a full signature, we mean a (Σ, Γ)-interpretation for some suitable (Σ, Γ); the same applies to other notions with the prefix (Σ, Γ) to be introduced below.

Roles and complex concepts are interpreted in $\mathcal{I}$ as follows:

$$
\begin{aligned}
(P_i^-)^{\mathcal{I}} &= \{ (v, u) \mid (u, v) \in P_i^{\mathcal{I}} \}, & \top^{\mathcal{I}} &= \Delta^{\mathcal{I}},\\
\bot^{\mathcal{I}} &= \emptyset, & (\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}},\\
(C_1 \sqcap C_2)^{\mathcal{I}} &= C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}} & (C_1 \sqcup C_2)^{\mathcal{I}} &= C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}},\\
(\exists R.C)^{\mathcal{I}} &= \{ u \mid (u, v) \in R^{\mathcal{I}} \text{ and } v \in C^{\mathcal{I}} \}, & (\forall R.C)^{\mathcal{I}} &= \{ u \mid v \in C^{\mathcal{I}} \text{ for all } (u, v) \in R^{\mathcal{I}} \}.
\end{aligned}
$$

For an inclusion or assertion α (whose individual names belong to Γ), we define the *truth-relation* $\mathcal{I} \models \alpha$ by taking:

$$
\begin{aligned}
\mathcal{I} &\models C_1 \sqsubseteq C_2 \quad \text{iff} \quad C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}, & \mathcal{I} &\models R_1 \sqsubseteq R_2 \quad \text{iff} \quad R_1^{\mathcal{I}} \subseteq R_2^{\mathcal{I}},\\
& & \mathcal{I} &\models R_1 \sqcap R_2 \sqsubseteq \bot \quad \text{iff} \quad R_1^{\mathcal{I}} \cap R_2^{\mathcal{I}} = \emptyset,\\
\mathcal{I} &\models A_k(a_i) \quad \text{iff} \quad a_i \in A_k^{\mathcal{I}}, & \mathcal{I} &\models P_k(a_i, a_j) \quad \text{iff} \quad (a_i, a_j) \in P_k^{\mathcal{I}}.
\end{aligned}
$$

Given a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, a Γ-interpretation $\mathcal{I}$ is called a *model* of $\mathcal{K}$ if $\mathsf{ind}(\mathcal{K}) \subseteq \Gamma$ and $\mathcal{I} \models \alpha$, for all $\alpha \in \mathcal{T} \cup \mathcal{A}$. In this case we write $\mathcal{I} \models \mathcal{K}$. We write $\mathcal{K} \models \alpha$, for an inclusion or assertion α that only uses individual names from $\mathsf{ind}(\mathcal{K})$, if $\mathcal{I} \models \alpha$ for all models $\mathcal{I}$ of $\mathcal{K}$. The notation $\mathcal{K} \models C(a)$, where C is any concept and $a \in \mathsf{ind}(\mathcal{K})$, should be understood in the same way. Finally, $\mathcal{K}$ is *consistent* if it has a model.

A *conjunctive query* (CQ) $\boldsymbol{q}(\boldsymbol{x})$ is a formula $\exists \boldsymbol{y}\, \varphi(\boldsymbol{x}, \boldsymbol{y})$, where φ is a conjunction of atoms of the form $A_k(z_1)$ or $P_k(z_1, z_2)$ with z_1, z_2 from $\boldsymbol{x}, \boldsymbol{y}$. Let $\mathcal{K}$ be a KB and $\boldsymbol{q}(\boldsymbol{x})$ a CQ. We call a tuple $\boldsymbol{a}$ of elements from $\mathsf{ind}(\mathcal{K})$ (of the same length as $\boldsymbol{x}$) a *certain answer* to $\boldsymbol{q}(\boldsymbol{x})$ over $\mathcal{K}$ if $\mathcal{I} \models \boldsymbol{q}(\boldsymbol{a})$ for all models $\mathcal{I}$ of $\mathcal{K}$ (understood as first-order structures). In this case we write $\mathcal{K} \models \boldsymbol{q}(\boldsymbol{a})$. For $\boldsymbol{q}$ without free variables, the answer to $\boldsymbol{q}$ is 'yes' if $\mathcal{K} \models \boldsymbol{q}$ and 'no' otherwise. We slightly abuse notation and write $\boldsymbol{a} \subseteq \Gamma$ to say that all elements of the tuple $\boldsymbol{a}$ are in Γ.

We remind the reader that, for combined complexity, the problem '$\mathcal{K} \models \boldsymbol{q}(\boldsymbol{a})$?' is NP-complete for the *DL-Lite* logics [6], $\mathcal{EL}$ and $\mathcal{ELH}_{\bot}^{dr}$ [27], and ExpTime-complete for the remaining Horn DLs introduced above [28]. For data complexity (with fixed $\mathcal{T}$ and $\boldsymbol{q}$), this problem is in AC^0 for the *DL-Lite* logics [6] and P-complete for the remaining DLs [27, 28].

3. Σ-Query Entailment, Materialisation and (Σ, Γ)-Homomorphism

We now define the central concepts of the article, Σ-query entailment and Σ-query inseparability, provide them with a semantic characterisation based on the notion of materialisation, and develop a theory of finitely generated materialisations.

Definition 1. Let $\mathcal{K}_1$ and $\mathcal{K}_2$ be KBs and Σ a relational signature. We say that $\mathcal{K}_1$ Σ-*query entails* $\mathcal{K}_2$ if

$$\mathcal{K}_2 \models \boldsymbol{q}(\boldsymbol{a}) \quad \text{implies} \quad \boldsymbol{a} \subseteq \mathsf{ind}(\mathcal{K}_1) \ \text{and} \ \mathcal{K}_1 \models \boldsymbol{q}(\boldsymbol{a}), \qquad \text{for all } \Sigma\text{-CQs } \boldsymbol{q}(\boldsymbol{x}) \text{ and all tuples } \boldsymbol{a} \subseteq \mathsf{ind}(\mathcal{K}_2).$$

Knowledge bases $\mathcal{K}_1$ and $\mathcal{K}_2$ are Σ-*query inseparable* if they Σ-query entail each other; in this case we write $\mathcal{K}_1 \equiv_\Sigma \mathcal{K}_2$.

We first quickly consider Σ-query entailment for the degenerate case when one of the involved KBs is inconsistent so that in the remainder of the article we can focus on consistent KBs only. Clearly, an inconsistent $\mathcal{K}_1$ Σ-query entails a KB $\mathcal{K}_2$ just in case $a \in \mathsf{ind}(\mathcal{K}_1)$ for all $a \in \mathsf{ind}(\mathcal{K}_2)$ with either $\mathcal{K}_2 \models A(a)$ or $\mathcal{K}_2 \models (\exists R)(a)$, for some $A \in \Sigma$ or Σ-role R. Now, suppose that $\mathcal{K}_1$ is consistent and $\mathcal{K}_2$ is inconsistent. Then $\mathcal{K}_1$ Σ-query entails $\mathcal{K}_2$ iff $\mathcal{K}_1 \models A(a)$ and $\mathcal{K}_1 \models P(a, b)$, for all concept and role names $A, P \in \Sigma$ and all $a, b \in \mathsf{ind}(\mathcal{K}_2)$. Thus, deciding Σ-query entailment in this case reduces to checking certain answers for all atomic Σ-CQs. A simple example showing that a consistent KB $\mathcal{K}_1$ can Σ-query entail an inconsistent KB $\mathcal{K}_2$ is given by $\mathcal{K}_1 = (\emptyset, \{A(a)\})$ and $\mathcal{K}_2 = (\{A \sqsubseteq \bot\}, \{A(a)\})$ with $\Sigma = \{A\}$. From now on we assume that all our KBs are *consistent*.

Definition 2. Let $\mathcal{K}$ be a KB. A $(\mathsf{sig}(\mathcal{K}), \mathsf{ind}(\mathcal{K}))$-interpretation $\mathcal{I}$ is called a *materialisation* of $\mathcal{K}$ if

$$\mathcal{K} \models \boldsymbol{q}(\boldsymbol{a}) \qquad \text{iff} \qquad \mathcal{I} \models \boldsymbol{q}(\boldsymbol{a}), \qquad \text{for all CQs } \boldsymbol{q}(\boldsymbol{x}) \text{ and all tuples } \boldsymbol{a} \subseteq \mathsf{ind}(\mathcal{K}).$$

We say that $\mathcal{K}$ is *materialisable* if it has a materialisation. (Note that we do not require a materialisation of $\mathcal{K}$ to be a *model* of $\mathcal{K}$.)

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

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

(It is to be emphasised that a Σ-role can be an inverse role even when we consider a language without role inverses.) We say that an element $u \in \Delta^{\mathcal{I}}$ is Σ-*participating in* $\mathcal{I}$ if $\boldsymbol{t}^{\mathcal{I}}_\Sigma(u) \neq \emptyset$ or $\boldsymbol{r}^{\mathcal{I}}_\Sigma(u, v) \neq \emptyset$, for some $v \in \Delta^{\mathcal{I}}$. The set of all individual names that are Σ-participating in $\mathcal{I}$ is denoted by $\mathsf{part}^{\mathcal{I}}_\Sigma$. Let $\mathcal{I}_i$ be Γ_i-interpretations, for $i = 1, 2$, such that $\Gamma \cap \mathsf{part}^{\mathcal{I}_1}_\Sigma \subseteq \Gamma_2$. A (Σ, Γ)-*homomorphism* h from $\mathcal{I}_1$ to $\mathcal{I}_2$ is a function $h \colon \Delta^{\mathcal{I}_1} \to \Delta^{\mathcal{I}_2}$ such that

– $h(a) = a$, for every $a \in \Gamma \cap \mathsf{part}^{\mathcal{I}_1}_\Sigma$,

– $\boldsymbol{t}^{\mathcal{I}_1}_\Sigma(u) \subseteq \boldsymbol{t}^{\mathcal{I}_2}_\Sigma(h(u))$ and $\boldsymbol{r}^{\mathcal{I}_1}_\Sigma(u, v) \subseteq \boldsymbol{r}^{\mathcal{I}_2}_\Sigma(h(u), h(v))$, for all $u, v \in \Delta^{\mathcal{I}_1}$.

Example 3. For $\Gamma_1 = \{a, b, c\}$, let $\mathcal{I}_1$ be a Γ_1-interpretation with $\Delta^{\mathcal{I}_1} = \{a, b, c\}$, $A^{\mathcal{I}_1} = \{a\}$, $B^{\mathcal{I}_1} = \{b\}$ and $C^{\mathcal{I}_1} = \{c\}$. If $\Sigma = \{A\}$ then $\mathsf{part}^{\mathcal{I}_1}_\Sigma = \{a\}$ as neither b nor c is Σ-participating in $\mathcal{I}_1$. For $\Gamma_2 = \{a, b, d\}$, let $\mathcal{I}_2$ be a Γ_2-interpretation with $\Delta^{\mathcal{I}_2} = \{a, b, d\}$, $A^{\mathcal{I}_2} = \{a\}$, $B^{\mathcal{I}_2} = \{d\}$ and $C^{\mathcal{I}_2} = \{b\}$. In this case, any map $h \colon \Delta^{\mathcal{I}_1} \to \Delta^{\mathcal{I}_2}$ with $h(a) = a$ is a $(\{A\}, \{a, b\})$-homomorphism from $\mathcal{I}_1$ to $\mathcal{I}_2$. However, there is no $(\{A, B\}, \{a, b\})$-homomorphism from $\mathcal{I}_1$ to $\mathcal{I}_2$ because $\mathsf{part}^{\mathcal{I}_1}_{\{A,B\}} = \{a, b\}$ but $\boldsymbol{t}^{\mathcal{I}_1}_{\{A,B\}}(b) \not\subseteq \boldsymbol{t}^{\mathcal{I}_2}_{\{A,B\}}(b)$.

We remind the reader of the following well-known link between certain answers to CQs and homomorphisms. Consider a CQ $\boldsymbol{q}(\boldsymbol{x}) = \exists \boldsymbol{y}\, \varphi(\boldsymbol{x}, \boldsymbol{y})$, a Γ'-interpretation $\mathcal{I}$, and a tuple $\boldsymbol{a} \subseteq \Gamma'$ of the same length as $\boldsymbol{x}$. Let Σ be the relational signature of $\boldsymbol{q}$, and let Γ be the set of individuals in $\boldsymbol{a}$. We can regard $\varphi(\boldsymbol{a}, \boldsymbol{y})$ as a (Σ, Γ)-interpretation $\mathcal{I}_{\varphi(\boldsymbol{a},\boldsymbol{y})}$ whose domain consists of the individuals in $\boldsymbol{a}$ and variables in $\boldsymbol{y}$, and $\mathcal{I}_{\varphi(\boldsymbol{a},\boldsymbol{y})} \models S(\boldsymbol{z})$ iff $S(\boldsymbol{z})$ is a conjunct of $\varphi(\boldsymbol{a}, \boldsymbol{y})$. In this case, we have $\mathcal{I} \models \boldsymbol{q}(\boldsymbol{a})$ iff there is a (Σ, Γ)-homomorphism from $\mathcal{I}_{\varphi(\boldsymbol{a},\boldsymbol{y})}$ to $\mathcal{I}$.

Suppose $\mathcal{I}_i$ is a materialisation of $\mathcal{K}_i$, for $i = 1, 2$. Since a composition of homomorphisms is again a homomorphism, if there is a $(\Sigma, \mathsf{ind}(\mathcal{K}_2))$-homomorphism from $\mathcal{I}_2$ to $\mathcal{I}_1$, then $\mathcal{K}_1$ Σ-query entails $\mathcal{K}_2$. The converse, however, does not necessarily hold, as shown by the following example.

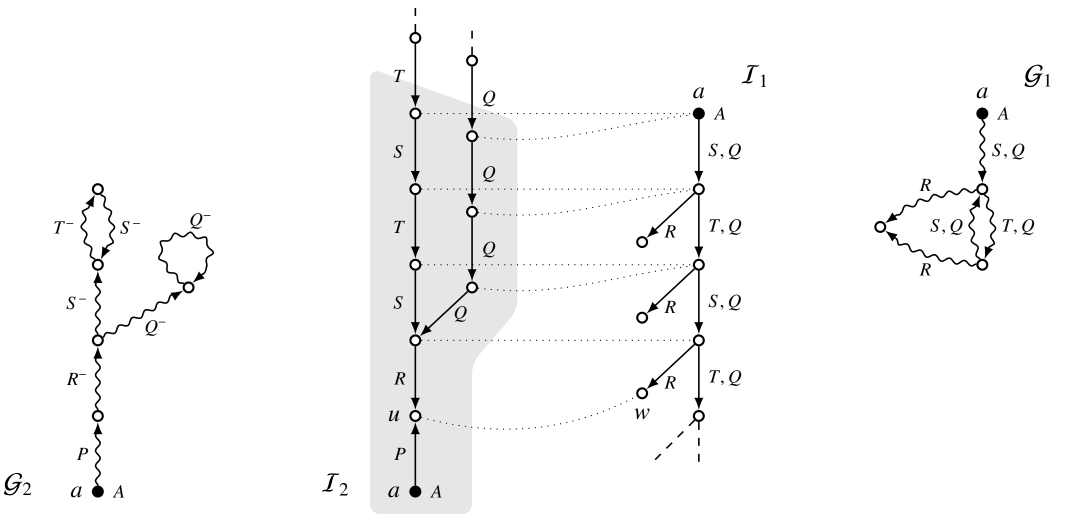


Figure 2: Materialisations $\mathcal{I}_2$ and $\mathcal{I}_1$ from Example 4 (dotted lines indicate a partial homomorphism) and their generating structures, $\mathcal{G}_2$ and $\mathcal{G}_1$.

Example 4. Consider the KBs $\mathcal{K}_i = (\mathcal{T}_i, \{A(a)\})$, for $i = 1, 2$, where

$$\mathcal{T}_1 = \{\, A \sqsubseteq \exists S,\ \ \exists S^- \sqsubseteq \exists T,\ \ \exists T^- \sqsubseteq \exists S,\ \ S \sqsubseteq Q,\ \ T \sqsubseteq Q,\ \ \exists Q^- \sqsubseteq \exists R \,\},$$
$$\mathcal{T}_2 = \{\, A \sqsubseteq \exists P,\ \ \exists P^- \sqsubseteq \exists R^-,\ \ \exists R \sqsubseteq \exists S^- \sqcap \exists Q^-,\ \ \exists Q \sqsubseteq \exists Q^-,\ \ \exists S \sqsubseteq \exists T^-,\ \ \exists T \sqsubseteq \exists S^- \,\}.$$

It is not hard to see (and it will be formally established below) that the interpretations $\mathcal{I}_1$ and $\mathcal{I}_2$ shown in Fig. 2 are materialisations of $\mathcal{K}_1$ and $\mathcal{K}_2$, respectively. Now, for $\Sigma = \{Q, R, S, T\}$, there is no $(\Sigma, \{a\})$-homomorphism from $\mathcal{I}_2$ to $\mathcal{I}_1$. Indeed, if we map u to, say, w then only the shaded part of $\mathcal{I}_2$ can be mapped $(\Sigma, \{a\})$-homomorphically to $\mathcal{I}_1$. On the other hand, $\mathcal{I}_2 \models \boldsymbol{q}(\boldsymbol{a})$ implies $\mathcal{I}_1 \models \boldsymbol{q}(\boldsymbol{a})$, for any Σ-CQ $\boldsymbol{q}(\boldsymbol{x})$, because any *finite* subinterpretation of $\mathcal{I}_2$ can be $(\Sigma, \{a\})$-homomorphically mapped to $\mathcal{I}_1$. This example motivates the following definitions.

A *subinterpretation* of a (Σ, Γ)-interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is a (Σ, Γ)-interpretation $\mathcal{I}' = (\Delta^{\mathcal{I}'}, \cdot^{\mathcal{I}'})$ with $\Delta^{\mathcal{I}'} \subseteq \Delta^{\mathcal{I}}$, $A^{\mathcal{I}'} = A^{\mathcal{I}} \cap \Delta^{\mathcal{I}'}$ and $P^{\mathcal{I}'} = P^{\mathcal{I}} \cap (\Delta^{\mathcal{I}'} \times \Delta^{\mathcal{I}'})$, for all concept and role names A and P. Now, given a signature (Σ, Γ), we say that an interpretation $\mathcal{I}_2$ is *finitely* (Σ, Γ)*-homomorphically embeddable* into an interpretation $\mathcal{I}_1$ if, for every *finite* subinterpretation $\mathcal{I}_2'$ of $\mathcal{I}_2$, there exists a (Σ, Γ)-homomorphism from $\mathcal{I}_2'$ to $\mathcal{I}_1$.

In the proof of the following criterion of Σ-query entailment, we regard any finite subinterpretation of $\mathcal{I}_2$ as a CQ whose variables are the elements of $\Delta^{\mathcal{I}_2}$, with $\mathsf{ind}(\mathcal{K}_2)$ being the answer variables.

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

Proof. $(\Rightarrow)$ Suppose $\mathcal{K}_1$ Σ-query entails $\mathcal{K}_2$. Let $\boldsymbol{a} = (a_1, \dots, a_n)$ be an enumeration of the individual names in $\mathsf{ind}(\mathcal{K}_2)$ that are Σ-participating in $\mathcal{I}_2$. Take any finite subinterpretation $\mathcal{I}_2'$ of $\mathcal{I}_2$ and let $u_1, \dots, u_{n+m}$ be an enumeration of those elements of $\Delta^{\mathcal{I}_2'}$ that are Σ-participating in $\mathcal{I}_2$ and such that $u_i = a_i$, for $i \leq n$. Consider a Σ-CQ

$$\boldsymbol{q}(x_1, \dots, x_n) = \exists x_{n+1} \dots \exists x_{n+m}\, \varphi(x_1, \dots, x_{n+m}), \quad \text{where} \quad \varphi(x_1, \dots, x_{n+m}) = \bigwedge_{\substack{i \leq n+m \\ A \in \boldsymbol{t}_\Sigma^{\mathcal{I}_2}(u_i)}} A(x_i) \;\wedge \bigwedge_{\substack{i,j \leq n+m \\ R \in \boldsymbol{r}_\Sigma^{\mathcal{I}_2}(u_i, u_j)}} R(x_i, x_j).$$

Since $\mathcal{I}_2 \models \varphi(u_1, \dots, u_{n+m})$, we have $\mathcal{I}_2 \models \boldsymbol{q}(\boldsymbol{a})$ and, since $\mathcal{I}_2$ is a materialisation, $\mathcal{K}_2 \models \boldsymbol{q}(\boldsymbol{a})$. As $\mathcal{K}_1$ Σ-query entails $\mathcal{K}_2$, we have $\boldsymbol{a} \subseteq \mathsf{ind}(\mathcal{K}_1)$ and $\mathcal{K}_1 \models \boldsymbol{q}(\boldsymbol{a})$. Since $\mathcal{I}_1$ is a materialisation, $\mathcal{I}_1 \models \boldsymbol{q}(\boldsymbol{a})$, and so $\mathcal{I}_1 \models \varphi(\boldsymbol{a}, v_{n+1}, \dots, v_{n+m})$, for some $v_{n+1}, \dots, v_{n+m} \in \Delta^{\mathcal{I}_1}$. Define a map $h\colon \Delta^{\mathcal{I}_2'} \to \Delta^{\mathcal{I}_1}$ by taking $h(a_i) = a_i$, for $i \leq n$, and $h(u_{n+i}) = v_{n+i}$, for $i \leq m$ (the rest of the domain of $\mathcal{I}_2'$ can be mapped arbitrarily as they are not Σ-participating in it). It can be readily seen that h is a (Σ, Γ)-homomorphism from $\mathcal{I}_2'$ to $\mathcal{I}_1$.

($\Leftarrow$) Suppose $\mathcal{I}_2$ is finitely $(\Sigma, \mathsf{ind}(\mathcal{K}_2))$-homomorphically embeddable into $\mathcal{I}_1$. Consider a Σ-CQ $\boldsymbol{q}(\boldsymbol{x}) = \exists \boldsymbol{y}\, \varphi(\boldsymbol{x}, \boldsymbol{y})$ and let $\mathcal{K}_2 \models \boldsymbol{q}(\boldsymbol{a})$, for some $\boldsymbol{a} \subseteq \mathsf{ind}(\mathcal{K}_2)$. Since $\mathcal{I}_2$ is a materialisation of $\mathcal{K}_2$, there is a tuple $\boldsymbol{u} = (u_1, \dots, u_m)$ of elements in $\Delta^{\mathcal{I}_2}$ such that $\mathcal{I}_2 \models \varphi(\boldsymbol{a}, \boldsymbol{u})$. Let $\mathcal{I}'_2$ be a subinterpretation of $\mathcal{I}_2$ with $\Delta^{\mathcal{I}'_2} = \mathsf{ind}(\mathcal{K}_2) \cup \{u_1, \dots, u_m\}$ and let h be a $(\Sigma, \mathsf{ind}(\mathcal{K}_2))$-homomorphism from $\mathcal{I}'_2$ to $\mathcal{I}_1$. Observe that each individual in $\boldsymbol{a}$ is Σ-participating in $\mathcal{I}'_2$, and so $h(a_i) = a_i$, for each a_i in $\boldsymbol{a}$. We also have $\mathcal{I}_1 \models \varphi(\boldsymbol{a}, h(u_1), \dots, h(u_m))$, whence $\boldsymbol{a} \subseteq \mathsf{ind}(\mathcal{K}_1)$ and $\mathcal{K}_1 \models \boldsymbol{q}(\boldsymbol{a})$. ❑

One problem with applying Theorem 5 is that materialisations are in general infinite for any of the DLs considered in this article. We address this problem by introducing finite representations of materialisations and showing that *Horn-$\mathcal{ALCHI}$* and all of its fragments defined above do have such finite representations.

Definition 6. Let $\mathcal{K}$ 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 some set Ω disjoint from $\mathsf{ind}(\mathcal{K})$,
- $(\Delta^{\mathcal{G}}, \cdot^{\mathcal{G}})$ is an interpretation with $P_i^{\mathcal{G}} \subseteq \mathsf{ind}(\mathcal{K}) \times \mathsf{ind}(\mathcal{K})$, for all role names P_i,
- $(\Delta^{\mathcal{G}}, \rightsquigarrow)$ is a directed graph (possibly containing loops) with nodes $\Delta^{\mathcal{G}}$ and arrows $\rightsquigarrow \subseteq \Delta^{\mathcal{G}} \times \Omega$, in which
  - every $w \rightsquigarrow w'$ is labelled with a set $(w, w')^{\mathcal{G}} \neq \emptyset$ of roles such that $(w_1, w')^{\mathcal{G}} = (w_2, w')^{\mathcal{G}}$ whenever $w_i \rightsquigarrow w'$, for $i = 1, 2$,
  - every $w \in \Omega$ is reachable by a path from $\mathsf{ind}(\mathcal{K})$,

  where by a *path*, σ, we mean any sequence $w_0 \cdots w_n$ with $w_0 \in \mathsf{ind}(\mathcal{K})$ and $w_i \rightsquigarrow w_{i+1}$ for $i < n$.

The *unravelling* $\mathcal{M}$ of $\mathcal{G}$ is a $(\mathsf{sig}(\mathcal{K}), \mathsf{ind}(\mathcal{K}))$-interpretation $(\Delta^{\mathcal{M}}, \cdot^{\mathcal{M}})$ such that

$$\begin{aligned}
&\Delta^{\mathcal{M}} \text{ is the set of paths in } \mathcal{G}, \\
&A^{\mathcal{M}} = \{\, \sigma \mid \mathsf{tail}(\sigma) \in A^{\mathcal{G}} \,\}, && \text{for each concept name } A, \\
&P^{\mathcal{M}} = P^{\mathcal{G}} \cup \{\, (\sigma, \sigma w) \mid \mathsf{tail}(\sigma) \rightsquigarrow w,\ P \in (\mathsf{tail}(\sigma), w)^{\mathcal{G}} \,\} \\
&\qquad\quad \cup \{\, (\sigma w, \sigma) \mid \mathsf{tail}(\sigma) \rightsquigarrow w,\ P^- \in (\mathsf{tail}(\sigma), w)^{\mathcal{G}} \,\}, && \text{for each role name } P,
\end{aligned}$$

where $\mathsf{tail}(\sigma)$ is the last element of a path σ. We call $\mathcal{G}$ a *generating structure for* $\mathcal{K}$ if its unravelling is a materialisation of $\mathcal{K}$. We say that a DL $\mathcal{L}$ has *finitely generated materialisations* if every $\mathcal{L}$-KB has a generating structure.

For instance, the materialisations $\mathcal{I}_2$ and $\mathcal{I}_1$ from Example 4 are isomorphic to the unravellings of the structures $\mathcal{G}_2$ and $\mathcal{G}_1$ in Fig. 2, respectively, and so $\mathcal{G}_i$ is a generating structure for the KB $\mathcal{K}_i$ from that example, for $i = 1, 2$.

To construct generating structures for KBs, we first transform their TBoxes into normal form [29]. Let $\mathcal{L}$ be any of our DLs. An $\mathcal{L}$-TBox is said to be in *normal form* if its inclusions are of the following form:

$$\begin{aligned}
A_1 &\sqsubseteq A_2, & \top &\sqsubseteq A, \\
A_1 &\sqsubseteq \forall R.A_2, & \top &\sqsubseteq \forall R.A, \\
\exists R.C &\sqsubseteq A, & A &\sqsubseteq \exists R.C, \\
A_1 \sqcap A_2 &\sqsubseteq A, & R_1 &\sqsubseteq R_2,
\end{aligned}$$

where A, A_1, A_2 are concept names, C is a concept name or $\top$, and R, R_1, R_2 are roles. To describe the relationship between a TBox and its transformation into normal form, we introduce the notion of model inseparability. Let (Σ, Γ) be a signature. We say that Γ-interpretations $\mathcal{I}_1$ and $\mathcal{I}_2$ *coincide on* Σ if $\Delta^{\mathcal{I}_1} = \Delta^{\mathcal{I}_2}$ and $S^{\mathcal{I}_1} = S^{\mathcal{I}_2}$, for all $S \in \Sigma$; in this case we write $\mathcal{I}_1 =_\Sigma \mathcal{I}_2$. KBs $\mathcal{K}_1$ and $\mathcal{K}_2$ with $\mathsf{ind}(\mathcal{K}_1) = \mathsf{ind}(\mathcal{K}_2)$ are called Σ-*model inseparable* if, for every model $\mathcal{I}_1$ of $\mathcal{K}_1$, there exists a model $\mathcal{I}_2$ of $\mathcal{K}_2$ such that $\mathcal{I}_2 =_\Sigma \mathcal{I}_1$, and vice versa. The following was shown in [29, 28, 22]:

Theorem 7. *Let $\mathcal{L}$ be any of our DLs. Given a consistent $\mathcal{L}$-KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, one can construct in polynomial time an $\mathcal{L}$-KB $\mathcal{K} = (\mathcal{T}', \mathcal{A})$ in normal form such that $\mathcal{K}$ and $\mathcal{K}'$ are $\mathsf{sig}(\mathcal{T})$-model inseparable.*

(Note that the 'negative' axioms of the form $A \sqsubseteq \bot$, $A_1 \sqcap A_2 \sqsubseteq \bot$ and $R_1 \sqcap R_2 \sqsubseteq \bot$ can be removed from a TBox if the knowledge base is known to be consistent.)

We show now how to define the generating structures. Suppose we are given a (consistent) KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ with a *Horn-$\mathcal{ALCHI}$* TBox $\mathcal{T}$ in normal form. For a role R, the *equivalence class* $[R]$ *of* R *with respect to* $\mathcal{T}$ is defined by taking

$$[R] = \{ S \mid \mathcal{T} \models R \sqsubseteq S \text{ and } \mathcal{T} \models S \sqsubseteq R \}.$$

Denote by $\mathsf{con}(\mathcal{T})$ the set of

- concepts of the form $\top$, A and $\exists R.A$ that occur in $\mathcal{T}$, as well as
- concepts of the form $\exists R^-.C$ such that $\mathcal{T}$ contains $C \sqsubseteq \forall R.A$.

The $\mathcal{T}$*-type of* $u \in \Delta^\mathcal{I}$ *in* $\mathcal{I}$ is the set $\boldsymbol{\tau}^\mathcal{I}_\mathcal{T}(u) = \{ C \in \mathsf{con}(\mathcal{T}) \mid u \in C^\mathcal{I} \}$. We say that $\boldsymbol{\tau} \subseteq \mathsf{con}(\mathcal{T})$ is a $\mathcal{T}$*-type* if there exists a model $\mathcal{I}$ of $\mathcal{T}$ such that $\boldsymbol{\tau} = \boldsymbol{\tau}^\mathcal{I}_\mathcal{T}(u)$, for some $u \in \Delta^\mathcal{I}$. Denote by $\mathsf{type}(\mathcal{T})$ the set of all $\mathcal{T}$-types. It is well-known [30] that $\mathsf{type}(\mathcal{T})$ can be computed in exponential time in $|\mathcal{T}|$. We can order $\mathcal{T}$-types by the set-theoretic inclusion $\subseteq$. Sometimes we use $\boldsymbol{\tau}$ in concepts (say, $\exists R.\boldsymbol{\tau}$), in which case it should be understood as an abbreviation for $\bigsqcap_{C \in \boldsymbol{\tau}} C$.

Now, we define the *generating relation* $\rightsquigarrow$ on the set comprising $\mathsf{ind}(\mathcal{K})$ and $\Omega_\mathcal{T}$, which is the set of *all* pairs of the form $([R], \boldsymbol{\tau})$, for a role R in $\mathcal{T}$ and $\boldsymbol{\tau} \in \mathsf{type}(\mathcal{T})$. For $a \in \mathsf{ind}(\mathcal{K})$ and $([R_1], \boldsymbol{\tau}_1), ([R_2], \boldsymbol{\tau}_2) \in \Omega_\mathcal{T}$, we set

$$\begin{aligned} a \rightsquigarrow ([R_2], \boldsymbol{\tau}_2) \quad &\text{iff} \quad \boldsymbol{\tau}_2 \text{ is a } \subseteq\text{-maximal } \mathcal{T}\text{-type such that } \mathcal{K} \models (\exists R_2.\boldsymbol{\tau}_2)(a) \quad \text{and} \\ &\qquad \mathcal{K} \not\models R_2(a,b), \text{ for any } b \in \mathsf{ind}(\mathcal{K}) \text{ with } \boldsymbol{\tau}_2 \subseteq \{ C \in \mathsf{con}(\mathcal{T}) \mid \mathcal{K} \models C(b) \}; \\ ([R_1], \boldsymbol{\tau}_1) \rightsquigarrow ([R_2], \boldsymbol{\tau}_2) \quad &\text{iff} \quad \boldsymbol{\tau}_2 \text{ is a } \subseteq\text{-maximal } \mathcal{T}\text{-type such that } \mathcal{T} \models \boldsymbol{\tau}_1 \sqsubseteq \exists R_2.\boldsymbol{\tau}_2. \end{aligned}$$

The *generating structure* $\mathcal{G} = (\Delta^\mathcal{G}, \cdot^\mathcal{G}, \rightsquigarrow)$ is defined as follows. Let $\Omega \subseteq \Omega_\mathcal{T}$ be the set of all w such that there are $a \in \mathsf{ind}(\mathcal{K})$ and $w_1, \dots, w_n \in \Omega_\mathcal{T}$ with $a \rightsquigarrow w_1 \rightsquigarrow \cdots \rightsquigarrow w_n = w$; in other words, Ω is the subset of $\Omega_\mathcal{T}$ that is reachable from $\mathsf{ind}(\mathcal{K})$ via $\rightsquigarrow$-arrows. Thus, $\Delta^\mathcal{G} = \mathsf{ind}(\mathcal{K}) \cup \Omega$. (The restriction of $\rightsquigarrow$ to $\Delta^\mathcal{G}$ will also be denoted by $\rightsquigarrow$.) Second, the interpretation function $\cdot^\mathcal{G}$ and the labelling of the graph $(\Delta^\mathcal{G}, \rightsquigarrow)$ are defined by setting

$$\begin{aligned} A^\mathcal{G} &= \{ a \in \mathsf{ind}(\mathcal{K}) \mid \mathcal{K} \models A(a) \} \ \cup \ \{ ([R], \boldsymbol{\tau}) \in \Omega \mid A \in \boldsymbol{\tau} \}, \\ P^\mathcal{G} &= \{ (a,b) \mid R(a,b) \in \mathcal{A} \text{ and } \mathcal{T} \models R \sqsubseteq P \}, \\ (w, w')^\mathcal{G} &= \{ S \mid \mathcal{T} \models R \sqsubseteq S \}, \text{ for every } w \rightsquigarrow w' \text{ with } w = ([R], \boldsymbol{\tau}) \end{aligned}$$

(here we assume that $P^-(b,a) \in \mathcal{A}$ if $P(a,b) \in \mathcal{A}$). In order to show that the constructed $\mathcal{G} = (\Delta^\mathcal{G}, \cdot^\mathcal{G}, \rightsquigarrow)$ is indeed a generating structure for $\mathcal{K}$, we need to establish that its unravelling is a materialisation.

Theorem 8. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a (consistent) KB with a Horn-$\mathcal{ALCHI}$ TBox in normal form. Let $\mathcal{G}$ be the structure defined above. Then the unravelling $\mathcal{M}$ of $\mathcal{G}$ is a materialisation of $\mathcal{K}$, and $\mathcal{G}$ is a generating structure for $\mathcal{K}$.*

Proof. We require two lemmas. The proof of the first one is routine and can be found in Appendix A:

Lemma 9. *$\mathcal{M}$ is a model of $\mathcal{K}$. Moreover,*

- $\boldsymbol{\tau}^\mathcal{M}_\mathcal{T}(a) = \{ C \in \mathsf{con}(\mathcal{T}) \mid \mathcal{K} \models C(a) \}$, *for all* $a \in \mathsf{ind}(\mathcal{K})$*;*
- $\boldsymbol{\tau}^\mathcal{M}_\mathcal{T}(\sigma) = \boldsymbol{\tau}$, *for all* $\sigma \in \Delta^\mathcal{M}$ *with* $\mathsf{tail}(\sigma) = ([R], \boldsymbol{\tau})$.

The second lemma says that $\mathcal{M}$ is a universal model of $\mathcal{K}$ in the following sense:

Lemma 10. *For every model $\mathcal{I}$ of $\mathcal{K}$, there exists a $(\mathsf{sig}(\mathcal{K}), \mathsf{ind}(\mathcal{K}))$-homomorphism from $\mathcal{M}$ to $\mathcal{I}$.*

Proof. Let $\Sigma = \mathsf{sig}(\mathcal{K})$ and $\Gamma = \mathsf{ind}(\mathcal{K})$. By induction on the length of $\sigma \in \Delta^{\mathcal{M}}$, we define a function $h\colon \Delta^{\mathcal{M}} \to \Delta^{\mathcal{I}}$ which satisfies the following properties implying that h is a (Σ, Γ)-homomorphism:

$$h(a) = a, \qquad \text{for } a \in \Gamma, \tag{1}$$

$$\boldsymbol{\tau}^{\mathcal{M}}_{\mathcal{T}}(\sigma) \subseteq \boldsymbol{\tau}^{\mathcal{I}}_{\mathcal{T}}(h(\sigma)), \qquad \text{for } \sigma \in \Delta^{\mathcal{M}}, \tag{2}$$

$$\boldsymbol{r}^{\mathcal{M}}_{\Sigma}(\sigma, \sigma') \subseteq \boldsymbol{r}^{\mathcal{I}}_{\Sigma}(h(\sigma), h(\sigma')), \qquad \text{for } \sigma, \sigma' \in \Delta^{\mathcal{M}}. \tag{3}$$

(Note that (2) refers to the full $\mathcal{T}$-types comprising concepts of the form $\top$, A and $\exists R.B$ rather than the atomic Σ-types $\boldsymbol{t}$ containing only concept names.)

First, for each $a \in \Gamma$, we set $h(a) = a$ in accordance with (1). Conditions (2) and (3) for $\sigma, \sigma' \in \Gamma$ follow from Lemma 9, the fact that $\mathcal{I}$ is a model of $\mathcal{K}$, and the construction of $\mathcal{M}$.

Suppose now that $h(\sigma)$ has already been defined for $\sigma \cdot ([S], \boldsymbol{\tau}) \in \Delta^{\mathcal{M}}$. By the construction of $\mathcal{M}$, it follows that $\mathcal{K} \models (\exists S.\boldsymbol{\tau})(a)$ if $\sigma = a$, or $\mathcal{T} \models \boldsymbol{\tau}' \sqsubseteq \exists S.\boldsymbol{\tau}$ if $\mathsf{tail}(\sigma) = ([S'], \boldsymbol{\tau}')$. Since $\mathcal{I} \models \mathcal{K}$, by Lemma 9 and the induction hypothesis—$\boldsymbol{\tau}^{\mathcal{M}}_{\mathcal{T}}(\sigma) \subseteq \boldsymbol{\tau}^{\mathcal{I}}_{\mathcal{T}}(h(\sigma))$—it follows that there exists $z \in \Delta^{\mathcal{I}}$ such that $S \in \boldsymbol{r}^{\mathcal{I}}_{\Sigma}(h(\sigma), z)$ and $\boldsymbol{\tau} \subseteq \boldsymbol{\tau}^{\mathcal{I}}_{\mathcal{T}}(z)$. We set $h(\sigma \cdot ([S], \boldsymbol{\tau})) = z$ and show that (2) and (3) hold. By Lemma 9, we have $\boldsymbol{\tau}^{\mathcal{M}}_{\mathcal{T}}(\sigma \cdot ([S], \boldsymbol{\tau})) = \boldsymbol{\tau}$, whence (2). Next, observe that $R \in \boldsymbol{r}^{\mathcal{M}}_{\Sigma}(\sigma, \sigma \cdot ([S], \boldsymbol{\tau}))$ follows from $\mathcal{T} \models S \sqsubseteq R$, and since $\mathcal{I}$ is a model of $\mathcal{T}$, we obtain $R \in \boldsymbol{r}^{\mathcal{I}}_{\Sigma}(h(\sigma), z)$, thus proving (3). ❑

We are now in a position to complete the proof of Theorem 8. We show that $\mathcal{K} \models \boldsymbol{q}(\boldsymbol{a})$ if and only if $\mathcal{M} \models \boldsymbol{q}(\boldsymbol{a})$, for any CQ $\boldsymbol{q}(\boldsymbol{x}) = \exists \boldsymbol{y}\, \varphi(\boldsymbol{x}, \boldsymbol{y})$ and $\boldsymbol{a} \subseteq \mathsf{ind}(\mathcal{K})$. If $\mathcal{K} \models \boldsymbol{q}(\boldsymbol{a})$ then, by Lemma 9, $\mathcal{M} \models \boldsymbol{q}(\boldsymbol{a})$. Conversely, suppose $\mathcal{M} \models \boldsymbol{q}(\boldsymbol{a})$. Then there exist a tuple $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_m)$ of elements in $\Delta^{\mathcal{M}}$ such that $\mathcal{M} \models \varphi(\boldsymbol{a}, \boldsymbol{\sigma})$. Let $\mathcal{I}$ be any model of $\mathcal{K}$. By Lemma 10, there exists a $(\mathsf{sig}(\mathcal{K}), \mathsf{ind}(\mathcal{K}))$-homomorphism h from $\mathcal{M}$ to $\mathcal{I}$. But then we have $\mathcal{I} \models \varphi(\boldsymbol{a}, h(\sigma_1), \dots, h(\sigma_m))$, and so $\mathcal{I} \models \boldsymbol{q}(\boldsymbol{a})$. ❑

Note that the generating structures $\mathcal{G} = (\Delta^{\mathcal{G}}, \cdot^{\mathcal{G}}, \rightsquigarrow)$ of KBs $\mathcal{K}$ with *Horn-$\mathcal{ALCHI}$*, *Horn-$\mathcal{ALCI}$*, *Horn-$\mathcal{ALCH}$* and *Horn-$\mathcal{ALC}$* TBoxes can contain *exponentially many* (in $|\mathcal{T}|$) elements in Ω (remember that $\Delta^{\mathcal{G}} = \mathsf{ind}(\mathcal{K}) \cup \Omega$); cf. Section 5. Note also that if the TBox in $\mathcal{K}$ is in *Horn-$\mathcal{ALCH}$* (or one of its fragments *Horn-$\mathcal{ALC}$*, $\mathcal{ELH}^{dr}_{\bot}$ or $\mathcal{EL}$) then it contains no inverse roles, and so the labels $(w, w')^{\mathcal{G}}$ on arrows $w \rightsquigarrow w'$ of the generating structure do not contain inverse roles either. We call such generating structures *forward*.

The generating structures of KBs with $\mathcal{ELH}^{dr}_{\bot}$ and $\mathcal{EL}$ TBoxes $\mathcal{T}$ contain *polynomially many* elements in Ω. Indeed, for every element $([R], \boldsymbol{\tau}) \in \Omega$, we can find a single concept $\exists R.A$ in $\mathcal{T}$ such that

$$\boldsymbol{\tau} \;=\; \{\, C \in \mathsf{con}(\mathcal{T}) \mid \mathcal{T} \models A \sqcap \bigsqcap_{\substack{\mathcal{T} \models R \sqsubseteq S \\ \top \sqsubseteq \forall S.B \text{ in } \mathcal{T}}} B \sqsubseteq C \,\}.$$

(This is not the case for *Horn-$\mathcal{ALC}$* because of axioms of the form $A_1 \sqsubseteq \forall R.A_2$ with $A_1 \neq \top$.) We remark that the generating structures for $\mathcal{EL}$ defined above were initially represented as pairs of functions by Brandt [31] and later called the canonical models; see, e.g., [32]. We prefer the term 'generating structure' to avoid confusion with the possibly infinite canonical model (materialisation).

Finally, the generating structures for KBs with *DL-Lite* TBoxes $\mathcal{T}$ also contain polynomially many elements in Ω because every $([R], \boldsymbol{\tau}) \in \Omega$ is determined by the role R:

$$\boldsymbol{\tau} \;=\; \{\, C \in \mathsf{con}(\mathcal{T}) \mid \mathcal{T} \models \exists R^- \sqsubseteq C \,\}.$$

Observe that if $\mathcal{T}$ does not contain role inclusions (which is the case for *DL-Lite$_{core}$* and *DL-Lite$_{horn}$* TBoxes) then, for any w and R, there is at most one w' such that $w \rightsquigarrow w'$ and $R \in (w, w')^{\mathcal{G}}$. Generating structures with this property will be called *functional*. We summarise these observations in the following theorem:

Theorem 11. *Horn-$\mathcal{ALCHI}$ and all of its fragments defined above have finitely generated materialisations. Furthermore, there is a polynomial p such that*

(*i*) *a generating structure $\mathcal{G}$ for any* Horn-$\mathcal{ALCHI}$ *KB $(\mathcal{T}, \mathcal{A})$ can be constructed in time $|\mathcal{A}| \cdot 2^{p(|\mathcal{T}|)}$;*

(*ii*) *a forward generating structure $\mathcal{G}$ for any* Horn-$\mathcal{ALCH}$ *KB $(\mathcal{T}, \mathcal{A})$ can be constructed in time $|\mathcal{A}| \cdot 2^{p(|\mathcal{T}|)}$;*

(*iii*) *a forward generating structure* $\mathcal{G}$ *for any* $\mathcal{ELH}^{dr}_{\bot}$ *KB* $(\mathcal{T}, \mathcal{A})$ *can be constructed in time* $|\mathcal{A}| \cdot p(|\mathcal{T}|)$;

(*iv*) *a generating structure* $\mathcal{G}$ *for any DL-Lite*$^{\mathcal{H}}_{horn}$ *KB* $(\mathcal{T}, \mathcal{A})$ *can be constructed in time* $|\mathcal{A}| \cdot p(|\mathcal{T}|)$;

(*v*) *a functional generating structure* $\mathcal{G}$ *for any DL-Lite*$_{horn}$ *KB* $(\mathcal{T}, \mathcal{A})$ *can be constructed in time* $|\mathcal{A}| \cdot p(|\mathcal{T}|)$.

As a final remark, we note that the generating structures $\mathcal{G} = (\Delta^{\mathcal{G}}, \cdot^{\mathcal{G}}, \rightsquigarrow)$ defined above can often be simplified. For example, in the case of *DL-Lite* KBs, we can impose the following additional restrictions on the generating relation $\rightsquigarrow$:

(lite$_1$) if $u \rightsquigarrow ([R], \boldsymbol{\tau})$ then $[R]$ is $\leq_{\mathcal{T}}$-minimal, where $[S] \leq_{\mathcal{T}} [T]$ iff $\mathcal{T} \models S \sqsubseteq T$;

(lite$_2$) if $([R_1], \boldsymbol{\tau}_1) \rightsquigarrow ([R_2], \boldsymbol{\tau}_2)$ then $[R_2^-] \neq [R_1]$.

It is easily seen that these simplifications do not affect the proof of Theorem 8 (the branches of the unravelling that are pruned as a result of these restrictions can be homomorphically mapped to other branches; for a more detailed argument, see the proof of Theorem 5 in the full version of [24]). The generating structure $\mathcal{G}_1$ in Fig. 2 as well as the generating structures in all our examples from Section 4 are constructed with these extra restrictions in mind.

So far we have only considered Σ-query entailment because Σ-query inseparability can be reduced to two Σ-query entailment checks. The following result shows that, conversely, one can reduce Σ-query entailment in LogSpace to Σ-query inseparability, for all DLs considered in this article except *DL-Lite*$_{core}$ and *DL-Lite*$_{horn}$.[3]

Theorem 12. *Let* $\mathcal{L}$ *be any of our DLs containing* $\mathcal{EL}$ *or having role inclusions. Then* Σ*-query entailment of consistent* $\mathcal{L}$*-KBs is* LogSpace*-reducible to* Σ*-query inseparability of* $\mathcal{L}$*-KBs.*

The proof of Theorem 12 is given in Appendix A and is based on the notions and results introduced in this section: the materialisations of KBs constructed to prove Theorem 11, the normal form of Theorem 7, and the semantic characterisation of Σ-query entailment given in Theorem 5. The underlying idea is to construct modifications $\mathcal{K}_1'$ and $\mathcal{K}_2'$ of the given KBs $\mathcal{K}_1$ and $\mathcal{K}_2$ such that $\mathcal{K}_1$ Σ-query entails $\mathcal{K}_2$ iff $\mathcal{K}_1$ and $\mathcal{K}_1' \cup \mathcal{K}_2'$ are Σ-query inseparable. Note that modifications of $\mathcal{K}_1$ and $\mathcal{K}_2$ are, in general, necessary: let $\mathcal{K}_1 = (\emptyset, \{A(a)\})$, $\mathcal{K}_2 = (\{A \sqsubseteq B\}, \{C(a)\})$ and $\Sigma = \{A, B\}$; then $\mathcal{K}_1$ Σ-query entails $\mathcal{K}_2$ but $\mathcal{K}_1$ does not Σ-query entail $\mathcal{K}_1 \cup \mathcal{K}_2$ since $\mathcal{K}_1 \cup \mathcal{K}_2 \models B(a)$.

4. Finite Σ-homomorphic Embeddability by Games

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

To explain the underlying intuition, we first reformulate the definition of finite Σ-homomorphic embedding in game-theoretic terms. Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be the materialisations obtained by unravelling finite generating structures $\mathcal{G}_1$ and $\mathcal{G}_2$ for (consistent) KBs $\mathcal{K}_1$ and $\mathcal{K}_2$, respectively. We assume that the ABox part of $\mathcal{M}_2$ is $(\Sigma, \mathsf{ind}(\mathcal{K}_2))$-homomorphically embeddable into the ABox part of $\mathcal{M}_1$, that is, the following condition holds:

(abox) $\mathsf{ind}(\mathcal{K}_1) \supseteq \mathsf{ind}(\mathcal{K}_2) \cap \mathsf{part}^{\mathcal{M}_2}_{\Sigma}$ and $\boldsymbol{t}^{\mathcal{M}_2}_{\Sigma}(a) \subseteq \boldsymbol{t}^{\mathcal{M}_1}_{\Sigma}(a)$ and $\boldsymbol{r}^{\mathcal{M}_2}_{\Sigma}(a, b) \subseteq \boldsymbol{r}^{\mathcal{M}_1}_{\Sigma}(a, b)$, for any $a, b \in \mathsf{ind}(\mathcal{K}_2) \cap \mathsf{part}^{\mathcal{M}_2}_{\Sigma}$.

The game is played by two players, player 1 and player 2, starting from some initial state $(\pi_0 \mapsto \sigma_0)$, where $\pi_0 \in \Delta^{\mathcal{M}_2}$ and $\sigma_0 \in \Delta^{\mathcal{M}_1}$. Intuitively, $(\pi_0 \mapsto \sigma_0)$ means that 'π_0 is to be Σ-homomorphically mapped onto σ_0'. Player 2 wants to demonstrate that there is no Σ-homomorphism from (a finite subinterpretation of) $\mathcal{M}_2$ into $\mathcal{M}_1$ with π_0 mapped onto σ_0: in each round $i > 0$ of the game, player 2 challenges player 1 with some π_i such that $\boldsymbol{r}^{\mathcal{M}_2}_{\Sigma}(\pi_{i-1}, \pi_i) \neq \emptyset$. Player 1, whose aim is to show that a Σ-homomorphism exists, has to respond with some σ_i such that the already constructed partial Σ-homomorphism can be extended with $\pi_i \mapsto \sigma_i$. It is easy to see that if, for any $\pi_0 \in \Delta^{\mathcal{M}_2}$, there exists $\sigma_0 \in \Delta^{\mathcal{M}_1}$ such that

[3] Note that, by Theorems 31 and 30, Σ-query entailment and inseparability are P-complete for *DL-Lite*$_{core}$ and *DL-Lite*$_{horn}$ in both combined and data complexity. *DL-Lite*$_{core}$ and *DL-Lite*$_{horn}$ are omitted from Theorem 12 since we have not found a direct LogSpace-reduction of Σ-query entailment to Σ-query inseparability.

– in each round $i < \omega$ of the game starting from the state $(\pi_0 \mapsto \sigma_0)$, player 1 can find a response to every challenge of player 2 (if any)

then there exists a Σ-homomorphism from $\mathcal{M}_2$ into $\mathcal{M}_1$, and the other way round. That $\mathcal{M}_2$ is *finitely* Σ-homomorphically embeddable into $\mathcal{M}_1$ is equivalent to the following condition: for any $\pi_0 \in \Delta^{\mathcal{M}_2}$ and any $n < \omega$, there exists $\sigma_0^n \in \Delta^{\mathcal{M}_1}$ such that,

– in each round $i < n$ of the game starting from $(\pi_0 \mapsto \sigma_0^n)$, player 1 has a response to every challenge of player 2.

It is also readily seen that instead of $\mathcal{M}_2$ player 2 can operate on its generating structure $\mathcal{G}_2$ and challenge player 1 with $\mathsf{tail}(\pi_{i-1}) \rightsquigarrow \mathsf{tail}(\pi_i)$.

To define our games formally, we require some notation. For a generating structure $\mathcal{G}$ for $\mathcal{K}$ and a relational signature Σ, the Σ-*types* $\boldsymbol{t}_\Sigma^{\mathcal{G}}(w)$ and $\boldsymbol{r}_\Sigma^{\mathcal{G}}(w, w')$ of $w, w' \in \Delta^{\mathcal{G}}$ are defined by:

$$\boldsymbol{t}_\Sigma^{\mathcal{G}}(w) = \{\, \Sigma\text{-concept name } A \mid w \in A^{\mathcal{G}} \,\}, \qquad \boldsymbol{r}_\Sigma^{\mathcal{G}}(w, w') = \begin{cases} \{\, \Sigma\text{-role } R \mid (w, w') \in R^{\mathcal{G}} \,\}, & \text{if } w, w' \in \mathsf{ind}(\mathcal{K}),\\ \{\, \Sigma\text{-role } R \mid R \in (w, w')^{\mathcal{G}} \,\}, & \text{if } w \rightsquigarrow w',\\ \emptyset, & \text{otherwise,} \end{cases}$$

where $(P^-)^{\mathcal{G}}$ is the inverse of $P^{\mathcal{G}}$. We also define $\bar{\boldsymbol{r}}_\Sigma^{\mathcal{G}}(w, w')$ to contain the inverses of the roles in $\boldsymbol{r}_\Sigma^{\mathcal{G}}(w, w')$. Note that $\bar{\boldsymbol{r}}^{\mathcal{G}}(a, b) = \boldsymbol{r}^{\mathcal{G}}(b, a)$, for $a, b \in \mathsf{ind}(\mathcal{K})$ but, in general, $\bar{\boldsymbol{r}}_\Sigma^{\mathcal{G}}(w, w')$ is not the same as $\boldsymbol{r}_\Sigma^{\mathcal{G}}(w', w)$ as shown by the T^-, S^--cycle in Fig. 2. We write $w \rightsquigarrow^\Sigma w'$ if $w \rightsquigarrow w'$ and $\boldsymbol{r}_\Sigma^{\mathcal{G}}(w, w') \neq \emptyset$.

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 relational 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 unravelling; $\mathcal{G}_i^\Sigma$ and $\mathcal{M}_i^\Sigma$ denote the restrictions of $\mathcal{G}_i$ and $\mathcal{M}_i$ to Σ. We first formalise the game described above and played on the finite generating structure $\mathcal{G}_2^\Sigma$ and the possibly infinite materialisation $\mathcal{M}_1^\Sigma$.

4.1. Infinite Game $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$

The *states* of this game are of the form $\mathfrak{s}_i = (u_i \mapsto \sigma_i)$, for $i \geq 0$, $u_i \in \Delta^{\mathcal{G}_2}$ and $\sigma_i \in \Delta^{\mathcal{M}_1}$, such that

(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

(s$_0$) $\sigma_0 = u_0$ in case $u_0 \in \mathsf{ind}(\mathcal{K}_2) \cap \mathsf{part}_\Sigma^{\mathcal{M}_2}$.

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 from $\mathsf{ind}(\mathcal{K}_2)$; 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. (Thus, player 2 loses if he has no challenge, while player 1 loses if he has no response.)

The following theorem gives a game-theoretic flavour to the criterion of Theorem 5.

Theorem 13. (*i*) $\mathcal{M}_2$ *is* finitely Σ-*homomorphically embeddable into* $\mathcal{M}_1$ *if and only if* **(abox)** *and the following condition hold*:

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

(*ii*) *There exists a Σ-homomorphism from $\mathcal{M}_2$ to $\mathcal{M}_1$ if and only if* **(abox)** *and the following condition hold*:

(ω-win) *for any $u_0 \in \Delta^{\mathcal{G}_2}$, there is $\sigma_0 \in \Delta^{\mathcal{M}_1}$ such that player 1 has an ω-winning strategy in the game $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ starting from $(u_0 \mapsto \sigma_0)$.*

Proof. We only prove (*i*) and leave (*ii*) to the reader.

($\Rightarrow$) Suppose $\mathcal{M}_2$ is finitely Σ-homomorphically embeddable into $\mathcal{M}_1$. Then **(abox)** holds by the definition of Σ-homomorphism. To show that **(win)** holds, suppose $u_0 \in \Delta^{\mathcal{G}_2}$ and $n < \omega$ are given. Take a finite subinterpretation $\mathcal{M}_{02}$ of $\mathcal{M}_2$ that contains σu_0, for some (say, the shortest) word σ, and all those elements of $\mathcal{M}_2$ whose distance from σu_0 does not exceed n ($\mathcal{M}_{02}$ also contains all individual names of $\mathcal{M}_2$). Let $h\colon \mathcal{M}_{02} \to \mathcal{M}_1$ be a $(\Sigma, \mathsf{ind}(\mathcal{K}_2))$-homomorphism. Take $\sigma_0 = h(\sigma u_0)$. Clearly, u_0 and σ_0 satisfy **(s_0)** and **(s_1)**. We show that player 1 has an n-winning strategy in the game $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ starting from $(u_0 \mapsto \sigma_0)$. Suppose player 2 picks $u_0 \rightsquigarrow^\Sigma_2 u_1$. Then $\sigma u_0 u_1$ is an element of $\mathcal{M}_{02}$, and player 1 responds with $\sigma_1 = h(\sigma u_0 u_1)$. Conditions **($s_1$)** and **($s_2$)** hold because h is a Σ-homomorphism. In the same way player 1 uses h to respond to all challenges of player 2 in any round $k < n$ of the game $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$.

($\Leftarrow$) Let $\mathcal{M}_{02}$ be a finite subinterpretation of $\mathcal{M}_2$. We enumerate elements of the domain of $\mathcal{M}_{02}$ in such a way that σ appears in the list before σ' whenever $\sigma' = \sigma u$, for some u. We define, by induction, a $(\Sigma, \mathsf{ind}(\mathcal{K}_2))$-homomorphism $h\colon \mathcal{M}_{02} \to \mathcal{M}_1$ as follows. Let n be the number of elements in the domain of $\mathcal{M}_{02}$. Pick the first (in the order described above) element σ that has not been mapped to $\mathcal{M}_1$ yet. There are two possible options.

– Suppose first that there is no $\sigma_0 \in \Delta^{\mathcal{M}_{02}}$ such that $\sigma = \sigma_0 u$ and $\mathsf{tail}(\sigma_0) \rightsquigarrow^2_\Sigma u$, for some u. Then, by **(win)**, there is $\sigma' \in \Delta^{\mathcal{M}_1}$ such that player 1 has an n-winning strategy in the game $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ starting from $(\mathsf{tail}(\sigma) \mapsto \sigma')$. We set $h(\sigma) = \sigma'$. Note that if $\sigma = a$, for some $a \in \mathsf{ind}(\mathcal{K}_2) \cap \mathsf{part}^{\mathcal{M}_2}_\Sigma$, then, by **($s_0$)**, $h(a) = a$.

– Otherwise, we consider the *longest* sequence $u_1, \dots, u_k$, $k \geq 1$, such that $\mathsf{tail}(\sigma_0) \rightsquigarrow^\Sigma_2 u_1 \rightsquigarrow^\Sigma_2 \cdots \rightsquigarrow^\Sigma_2 u_k$ and $\sigma_m = \sigma_0 u_1 \cdots u_m \in \Delta^{\mathcal{M}_{02}}$, for all $m < k$, with $\sigma = \sigma_k$. By the definition of the order, $\sigma_0, \dots, \sigma_{k-1}$ have already been mapped by h. By construction and **(win)**, player 1 has an n-winning strategy from $(\mathsf{tail}(\sigma_0) \mapsto h(\sigma_0))$. Therefore, player 1 has a response σ' to the challenge $\mathsf{tail}(\sigma_{k-1}) \rightsquigarrow^\Sigma_2 \mathsf{tail}(\sigma_k)$. So, we set $h(\sigma) = \sigma'$.

It is readily seen that, by **(abox)**, **(s_1)** and **(s_2)**, the constructed h is a $(\Sigma, \mathsf{ind}(\mathcal{K}_2))$-homomorphism from $\mathcal{M}_{02}$ to $\mathcal{M}_1$. ❑

Example 14. Consider $\mathcal{G}^\Sigma_2$ and $\mathcal{M}^\Sigma_1$ shown in Fig. 3a, where $\Sigma = \{Q, R\}$. An n-winning strategy for player 1 in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ starting from $(a \mapsto a)$ is shown by dotted lines with the rounds of the game indicated by the numbers on the dotted lines. In the state $(a \mapsto a)$, player 2 has two possible challenges: $a \rightsquigarrow^\Sigma_2 u$ and $a \rightsquigarrow^\Sigma_2 u'$. In response to the former, player 1 maps u to a and the successive challenges to the elements of the chain that begins with RQ (indicated by indices $1, 2, \dots$). In response to the latter challenge, player 1 maps u' and all the successive challenges to the same element a (indices $1', 2', \dots$). Note that in all but the starting state, player 2 has only one possible challenge.

Example 15. Consider now $\mathcal{G}^\Sigma_2$ and $\mathcal{M}^\Sigma_1$ in Fig. 3b, where $\Sigma = \{Q, R, S, T\}$ (see also Example 4). A 4-winning strategy for player 1 in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ starting from $(u_0 \mapsto \sigma_4)$ is shown in Fig. 3b by dotted lines (again, rounds of the game are indicated by the numbers). In contrast to Example 14, where player 1 either stays in the ABox or always moves away from it, the winning strategy for player 1 now is to move in the opposite direction, towards the ABox. (Note that in round 2, player 2 has two possible challenges, $u_1 \rightsquigarrow^\Sigma_2 u_2$ and $u_1 \rightsquigarrow^\Sigma_2 v$.) In fact, for any $n > 0$, player 1 has an n-winning strategy starting from any $(u_0 \mapsto \sigma_m)$ provided that m is even and $m \geq n$.

The criterion of Theorem 13 does not seem to be a big improvement on Theorem 5 as we still have to deal with an infinite materialisation. Note that, for some DLs such as $\mathcal{EL}$, *Horn-$\mathcal{ALC}$* and *DL-Lite$_{horn}$*, it is enough to play the same game as defined above but on the *finite* generating structures $\mathcal{G}_2$ and $\mathcal{G}_1$. We denote this naïve reformulation of $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$—in which σ_i and $\mathcal{M}_1$ are replaced with w_i and $\mathcal{G}_1$, respectively—by $G^n_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ and invite the reader to prove that, in the case of, say *DL-Lite$_{horn}$*, Theorem 13 will continue to hold if we replace **(win)** with the following condition, which can be checked in polynomial time in $O(|\mathcal{G}_2| \times |\mathcal{G}_1|)$: for any $u_0 \in \Delta^{\mathcal{G}_2}$, there exists $w_0 \in \Delta^{\mathcal{G}_1}$ such that player 1 has an ω-winning strategy in the game $G^n_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ starting from $(u_0 \mapsto w_0)$. (We shall obtain this result later as a consequence of a more general theorem.) Unfortunately, the existence of an ω-winning strategy in this naïve game does not imply Σ-homomorphic embeddability of $\mathcal{M}_2$ into $\mathcal{M}_1$ for DLs such as *DL-Lite$^{\mathcal{H}}_{core}$* or *Horn-$\mathcal{ALCI}$*.

In the remainder of this section, we show that condition **(win)** in the infinite game $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ can be checked by analysing a much more complex game on the generating structures $\mathcal{G}_2$ and $\mathcal{G}_1$. We consider four types of strategies in

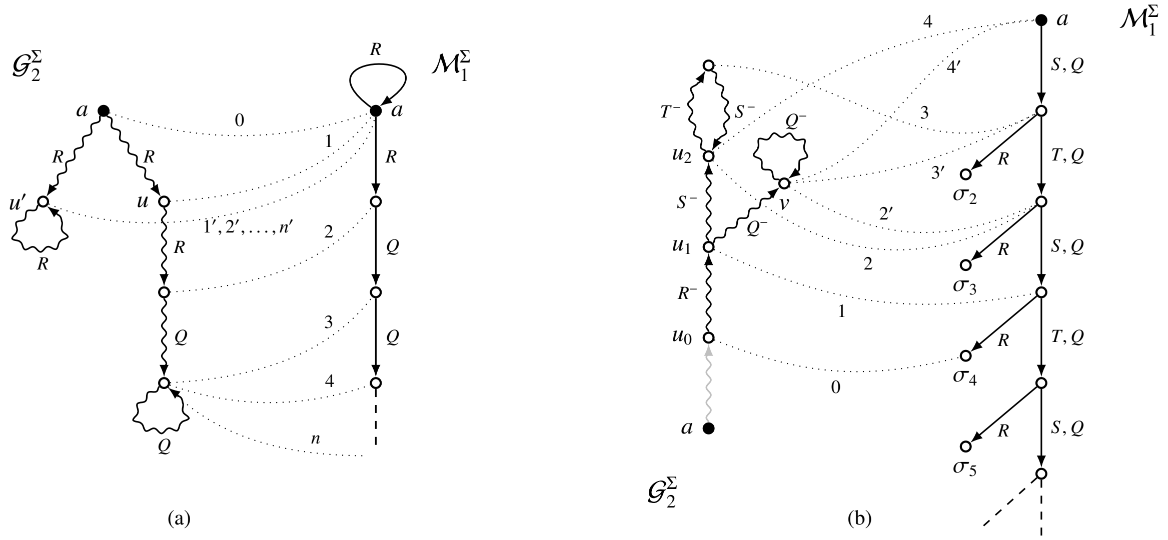


Figure 3: (a) Example 14: n-winning strategy in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ from $(a \mapsto a)$. (b) Example 15: 4-winning strategy in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ from $(u_0 \mapsto \sigma_4)$.

$G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$: forward, backward, start-bounded and general. For each strategy type, θ, we define a game $G^\theta_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ such that the following conditions are equivalent:

(win-θ) for any $u_0 \in \Delta^{\mathcal{G}_2}$ and $n < \omega$, there is $\sigma_0^n \in \Delta^{\mathcal{M}_1}$ such that player 1 has an n-winning θ-strategy in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ starting from $(u_0 \mapsto \sigma_0^n)$;

(ω-win$^\theta$) for any $u_0 \in \Delta^{\mathcal{G}_2}$, player 1 has an ω-winning strategy in $G^\theta_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ starting from some state depending on u_0 and θ.

We begin by considering 'forward' winning strategies (such as in Example 14) that are sufficient for the DLs without inverse roles.

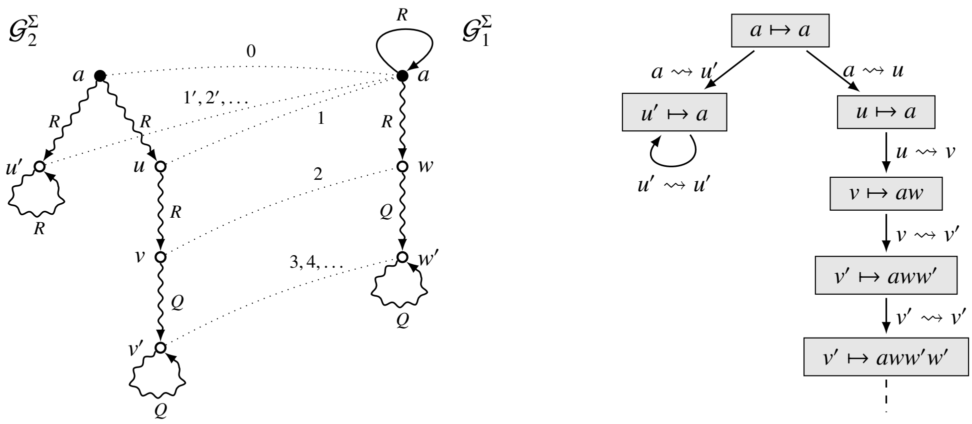
4.2. *Forward Strategy and Game $G^f_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$*

We say that a λ-strategy ($\lambda \le \omega$) 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 $\sigma_i = \sigma_{i-1}w$, for some $w \in \Delta^{\mathcal{G}_1}$. For instance, if the generating structures $\mathcal{G}_i$, $i = 1, 2$, are forward then *every* strategy in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ is forward, and so **(win)** coincides with **(win-*f*)**. By Theorem 11 (ii) and (iii), this is the case for *Horn-$\mathcal{ALCH}$*, *Horn-$\mathcal{ALC}$*, $\mathcal{ELH}^{dr}_\bot$ and $\mathcal{EL}$.

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 a λ-winning strategy in the *game* $G^f_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ whose states, initial states, challenges of player 2 and responses of player 1 are defined in the table below:

forward game $G^f_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$	
states, $i \ge 0$	$(u_i \mapsto w_i)$ with $u_i \in \Delta^{\mathcal{G}_2}$, $w_i \in \Delta^{\mathcal{G}_1}$ and $\boldsymbol{t}^{\mathcal{G}_2}_\Sigma(u_i) \subseteq \boldsymbol{t}^{\mathcal{G}_1}_\Sigma(w_i)$
initial state	$(u_0 \mapsto w_0)$ such that $w_0 = u_0$ in case $u_0 \in \mathsf{ind}(\mathcal{K}_2) \cap \mathsf{part}^{\mathcal{M}_2}_\Sigma$
challenges, $i > 0$	$u_{i-1} \rightsquigarrow^\Sigma_2 u_i$
responses, $i > 0$	w_i such that either $w_{i-1} \rightsquigarrow_1 w_i$ or $w_{i-1}, w_i \in \mathsf{ind}(\mathcal{K}_1)$ and $\boldsymbol{r}^{\mathcal{G}_2}_\Sigma(u_{i-1}, u_i) \subseteq \boldsymbol{r}^{\mathcal{G}_1}_\Sigma(w_{i-1}, w_i)$

(Note again that of all u_i only u_0 may belong to $\mathsf{ind}(\mathcal{K}_2)$.)



(a) (b)

Figure 4: The forward game $G^f_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ from $(a \mapsto a)$ in Example 16: (a) an ω-winning strategy for player 1; (b) the infinite graph $\mathfrak{T}$ for extracting ω-winning strategies.

Example 16. Consider $\mathcal{G}_2^\Sigma$ and $\mathcal{G}_1^\Sigma$ shown in Fig. 4a, where $\mathcal{G}_1^\Sigma$ is a generating structure that can be unravelled into $\mathcal{M}_1^\Sigma$ from Example 14. It is not hard to see that, for any $u_0 \in \Delta^{\mathcal{G}_2}$, there is $w_0 \in \Delta^{\mathcal{G}_1}$ such that player 1 has an ω-winning strategy in $G^f_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ starting from $(u_0 \mapsto w_0)$. Such a strategy starting from $(a \mapsto a)$ is depicted by dotted lines.

The reader may find more elegant proofs of the following lemma. However, the constructions we use will be required for the proofs of other lemmas, in particular, a more general Lemma 28.

Lemma 17. *Conditions* **(win-*f*)** *and* **(ω-win^f)** *are equivalent. More precisely, for any $u_0 \in \Delta^{\mathcal{G}_2}$ and $\sigma_0 \in \Delta^{\mathcal{M}_1}$, the following are equivalent*:

(a) *player 1 has an ω-winning forward strategy in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ starting from $(u_0 \mapsto \sigma_0)$;*

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

(c) *player 1 has an ω-winning strategy in $G^f_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ starting from $(u_0 \mapsto \mathsf{tail}(\sigma_0))$.*

Proof. (a) $\Rightarrow$ (b) is trivial.

(b) $\Rightarrow$ (c) We construct a (possibly infinite) directed graph $\mathfrak{T}$ whose nodes are of the form $(u \mapsto \delta)$, where $u \in \Delta^{\mathcal{G}_2}$ and δ is a suffix of some element in $\Delta^{\mathcal{M}_1}$, and whose arrows are labelled with $u \rightsquigarrow_2^\Sigma u'$ so that the following conditions hold:

(1) $\mathfrak{T}$ contains an *initial node* $(u_0 \mapsto \mathsf{tail}(\sigma_0))$;

(2) $\boldsymbol{t}_\Sigma^{\mathcal{G}_2}(u) \subseteq \boldsymbol{t}_\Sigma^{\mathcal{G}_1}(\mathsf{tail}(\delta))$, for every node $(u \mapsto \delta)$ in $\mathfrak{T}$;

(3) for any $u \rightsquigarrow_2^\Sigma u'$, every node $(u \mapsto \delta)$ in $\mathfrak{T}$ has exactly one $(u \rightsquigarrow_2^\Sigma u')$-successor, which can be of the following forms:

  (3.1) $(u' \mapsto \delta w')$ if $\mathsf{tail}(\delta) = w$, $w \rightsquigarrow_1 w'$ and $\boldsymbol{r}_\Sigma^{\mathcal{G}_2}(u, u') \subseteq \boldsymbol{r}_\Sigma^{\mathcal{G}_1}(w, w')$;

  (3.2) $(u' \mapsto b)$ if $\delta = a \in \mathsf{ind}(\mathcal{K}_1)$, $b \in \mathsf{ind}(\mathcal{K}_1)$ and $\boldsymbol{r}_\Sigma^{\mathcal{G}_2}(u, u') \subseteq \boldsymbol{r}_\Sigma^{\mathcal{G}_1}(a, b)$.

(The infinite graph $\mathfrak{T}$ for the winning strategy in Example 16 is depicted in Fig. 4b.)

Such a graph $\mathfrak{T}$ (if it exists) gives rise to the required ω-winning strategy for player 1 in $G^f_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$. Indeed, consider the function s mapping the nodes of $\mathfrak{T}$ to states in the game $G^f_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ and defined by taking

$$\mathsf{s}(u \mapsto \delta) = (u \mapsto \mathsf{tail}(\delta));$$

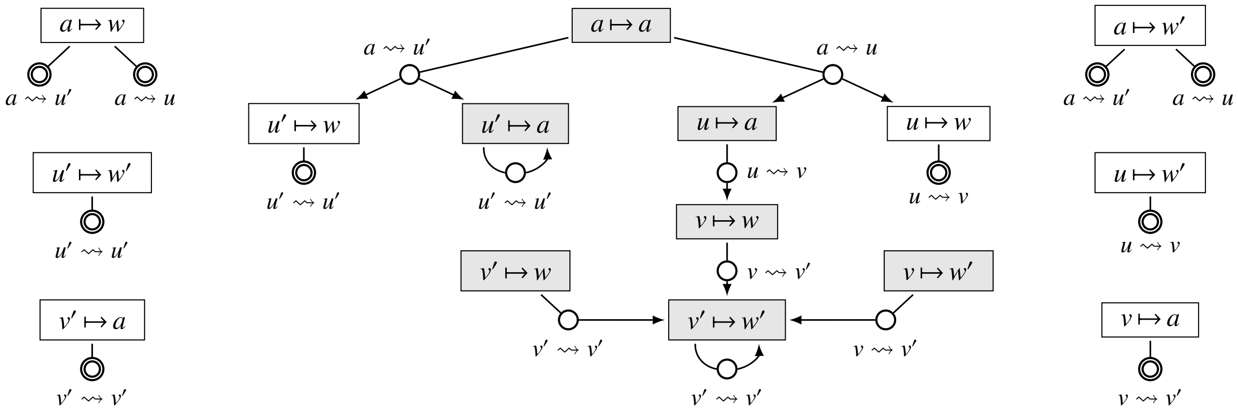


Figure 5: The full graph of the game $G^f_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ in Example 14.

in particular, the initial node $\mathfrak{n}_0$ of $\mathfrak{T}$ is mapped to the starting state: $\mathsf{s}(\mathfrak{n}_0) = (u_0 \mapsto \mathsf{tail}(\sigma_0))$. Now, when challenged by player 2 with $u \rightsquigarrow^\Sigma_2 u'$ in a state $\mathsf{s}(\mathfrak{n})$, player 1 picks a unique $u \rightsquigarrow^\Sigma_2 u'$-successor $\mathfrak{n}'$ of any $\mathfrak{r}$ in $\mathfrak{T}$ such that $\mathsf{s}(\mathfrak{r}) = \mathsf{s}(\mathfrak{n})$, and responds to the challenge with $\mathsf{s}(\mathfrak{n}')$. Note that although nodes are not uniquely determined by the states, *any* choice of $\mathfrak{r}$ as above results in an ω-winning strategy for player 1.

We now show that $\mathfrak{T}$ exists. Let $\mathbb{S}_0$ be the given set of n-winning forward strategies for player 1 in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ starting from $(u_0 \mapsto \sigma_0)$. Let $w_0 = \mathsf{tail}(\sigma_0)$. Define $\mathfrak{T}_0$ to be the graph with the single initial node $(u_0 \mapsto w_0)$. Clearly it satisfies (1) and (2) above. If it also satisfies (3), then we are done. Otherwise, we take all the challenges $u_0 \rightsquigarrow^\Sigma_2 u^1_1, \dots, u_0 \rightsquigarrow^\Sigma_2 u^k_1$ by player 2 and use the pigeonhole principle and the fact that the number of roles in $\mathcal{K}_1$ is finite to find $w^1_1, \dots, w^k_1 \in \Delta^{\mathcal{G}_1}$ and a subset $\mathbb{S}_1 \subseteq \mathbb{S}_0$ such that, for any challenge $u_0 \rightsquigarrow^\Sigma_2 u^i_1$, every strategy $\mathcal{S} \in \mathbb{S}_1$ gives a response $(u^i_1 \mapsto \sigma^i_1)$ with $\mathsf{tail}(\sigma^i_1) = w^i_1$. If $w^i_1 \in \mathsf{ind}(\mathcal{K}_1)$ then we add to $\mathfrak{T}_0$ the node $(u^i_1 \mapsto w^i_1)$; and if $w^i_1 \notin \mathsf{ind}(\mathcal{K}_1)$ then we add to $\mathfrak{T}_0$ the node $(u^i_1 \mapsto w_0 w^i_1)$; we also add a $u_0 \rightsquigarrow^\Sigma_2 u^i_1$ arc connecting $(u_0 \mapsto w_0)$ with the newly introduced node. This gives us the graph $\mathfrak{T}_1$. We proceed in the same way and construct a sequence of directed graphs $\mathfrak{T}_0 \subseteq \mathfrak{T}_1 \subseteq \dots$ until we either reach some $\mathfrak{T}_k$ satisfying (1)–(3) or obtain an infinite sequence and take $\mathfrak{T} = \bigcup_{k<\omega} \mathfrak{T}_k$, which obviously satisfies (1)–(3).

$(c) \Rightarrow (a)$ The given ω-winning strategy in $G^f_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ starting from $(u_0 \mapsto \mathsf{tail}(\sigma_0))$ is mirrored by an obvious ω-winning forward strategy in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ starting from $(u_0 \mapsto \sigma_0)$. ❑

Example 18. Consider again $\mathcal{G}^\Sigma_2$ and $\mathcal{G}^\Sigma_1$ in Fig. 4a. Figure 5 depicts the full graph of the game $G^f_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$, in which rectangles represent the states and circles the challenges of player 2. Note that it contains two dead-ends reachable from $(a \mapsto a)$—the challenges $u' \rightsquigarrow^\Sigma_2 u'$ in the state $(u' \mapsto w)$ and $u \rightsquigarrow^\Sigma_2 v$ in $(u \mapsto w)$ of player 2 to which player 1 has no response (the dead-ends are indicated by double circles). The ω-winning strategy for player 1 in this graph is, therefore, to avoid these dead-ends; it is indicated by the shaded states.

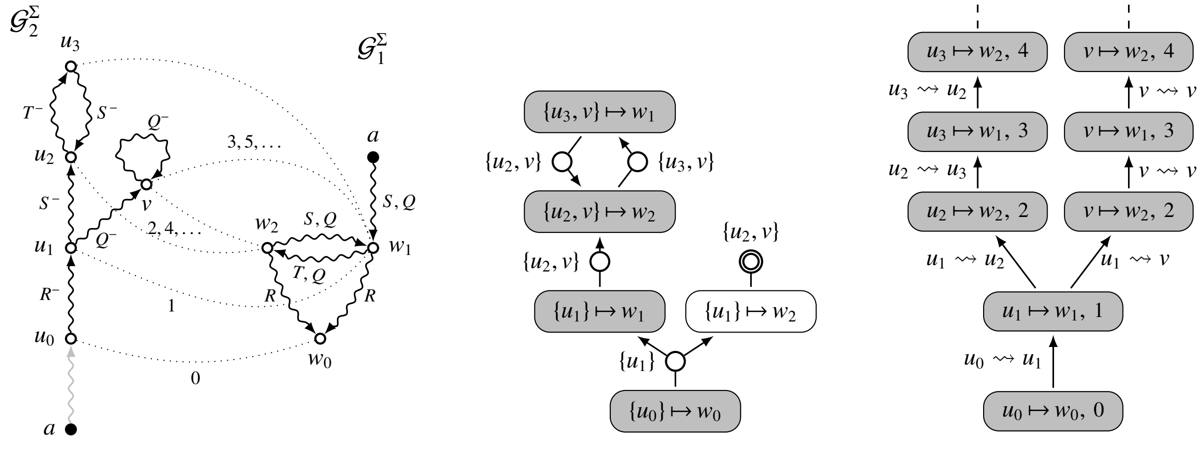
Consider now the full graph of the game $G^f_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ constructed similarly to the graph in Fig. 5. It is easy to see that player 1 has an ω-winning strategy in $G^f_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ starting from a state $\mathfrak{s}$ if and only if player 2 does not have a winning strategy in the *reachability game* (on the full graph) the aim of which is to reach a dead-end (where player 1 does not have a response) starting from $\mathfrak{s}$. As is well-known [33], the (non-)existence of such a strategy for player 2 can be checked in polynomial time in the size $O(|\mathcal{G}_2| \times |\mathcal{G}_1|)$ of the game graph. In view of Theorem 11 (ii) and (iii), we then obtain:

Theorem 19. *For combined complexity, checking Σ-query entailment is in* P *for $\mathcal{EL}$ and $\mathcal{ELH}^{dr}_\bot$ 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 15 can be described as 'backward'.

4.3. 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}



(a) (b) (c)

Figure 6: The backward game $G^b_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ from $(\{u_0\} \mapsto w_0)$ in Example 20: (a) an ω-winning strategy for player 1; (b) a fragment of the full game graph; (c) the infinite tree $\mathfrak{T}$ for extracting ω-winning strategies.

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 15. That is why the states of the game $G^b_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ are of the form $\mathfrak{s}_i = (\Xi_i \mapsto w_i)$, where Ξ_i is a *non-empty subset* of $\Delta^{\mathcal{G}_2}$ and $w_i \in \Delta^{\mathcal{G}_1}$. 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 $w_i \in \Delta^{\mathcal{G}_1}$ such that $w_i \rightsquigarrow_1 w_{i-1}$. More formally, the states, challenges of player 2 and responses by player 1 are defined as follows:

backward game $G^b_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$	
states, $i \geq 0$	$(\Xi_i \mapsto w_i)$ with $\Xi_i \subseteq \Delta^{\mathcal{G}_2}$, $\Xi_i \neq \emptyset$, $w_i \in \Delta^{\mathcal{G}_1}$ and $\boldsymbol{t}^{\mathcal{G}_2}_\Sigma(u) \subseteq \boldsymbol{t}^{\mathcal{G}_1}_\Sigma(w_i)$, for all $u \in \Xi_i$
initial state	$(\{u_0\} \mapsto w_0)$ such that $w_0 = u_0$ in case $u_0 \in \mathsf{ind}(\mathcal{K}_2) \cap \mathsf{part}^{\mathcal{M}_2}_\Sigma$
challenges, $i > 0$	$\Xi_i = \Xi^{\rightsquigarrow}_{i-1}$ provided that $\Xi_i \neq \emptyset$
responses, $i > 0$	w_i such that $w_i \rightsquigarrow_1 w_{i-1}$ and $\boldsymbol{r}^{\mathcal{G}_2}_\Sigma(u, v) \subseteq \bar{\boldsymbol{r}}^{\mathcal{G}_1}_\Sigma(w_i, w_{i-1})$, for all $u \in \Xi_{i-1}$ and $v \in \Xi_i$

(Note that, by definition, Ξ_0 is a singleton and the sets Ξ_i, for $i > 0$, contain no individuals from $\mathsf{ind}(\mathcal{K}_2)$.)

Example 20. Figure 6a shows an ω-winning strategy for player 1 in $G^b_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ starting from $(\{u_0\} \mapsto w_0)$, where $\mathcal{G}_1$ is a generating structure that can be unravelled into $\mathcal{M}_1$ in Example 15. Figure 6b presents the corresponding fragment of the full game graph (shaded nodes form an ω-winning strategy and the non-shaded node leads to a dead-end, where player 1 loses).

Lemma 21. *Conditions* **(win-*b*)** *and* **(ω-win^b)** *are equivalent. More precisely, for any* $u_0 \in \Delta^{\mathcal{G}_2}$ *and* $w_0 \in \Delta^{\mathcal{G}_1}$*, the following are equivalent*:

(*a*) *for every* $n < \omega$*, there is* $\sigma^n_0 \in \Delta^{\mathcal{M}_1}$ *with* $\mathsf{tail}(\sigma^n_0) = w_0$ *such that player 1 has an n-winning backward strategy in* $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ *starting from* $(u_0 \mapsto \sigma^n_0)$;

(*b*) *player 1 has an* ω*-winning strategy in* $G^b_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ *starting from* $(\{u_0\} \mapsto w_0)$.

Proof. (a) $\Rightarrow$ (b) We begin by constructing a possibly infinite directed *tree* $\mathfrak{T}$ with nodes of the form $(u \mapsto w, i)$, where $u \in \Delta^{\mathcal{G}_2}$, $w \in \Delta^{\mathcal{G}_1}$ and $0 \leq i < \omega$, whose arrows are labelled with $u \rightsquigarrow_2^\Sigma u'$ so that the following conditions hold:

(1) the root of $\mathfrak{T}$ is of the form $(u_0 \mapsto w_0, 0)$;

(2) $\boldsymbol{t}_\Sigma^{\mathcal{G}_2}(u) \subseteq \boldsymbol{t}_\Sigma^{\mathcal{G}_1}(w)$, for every node $(u \mapsto w, i)$ in $\mathfrak{T}$;

(3) for any $u \rightsquigarrow_2^\Sigma u'$, every node $(u \mapsto w, i)$ in $\mathfrak{T}$ has exactly one $(u \rightsquigarrow_2^\Sigma u')$-successor in $\mathfrak{T}$, which is of the form $(u' \mapsto w', i+1)$ and satisfies $w' \rightsquigarrow_1 w$ and $\boldsymbol{r}_\Sigma^{\mathcal{G}_2}(u, u') \subseteq \bar{\boldsymbol{r}}_\Sigma^{\mathcal{G}_1}(w', w)$.

(4) for any nodes $(u \mapsto w, i)$ and $(u' \mapsto w', i)$ in $\mathfrak{T}$, we have $w = w'$.

(The infinite tree $\mathfrak{T}$ for the winning strategy in Example 20 is depicted in Fig. 6c.)

Such a tree $\mathfrak{T}$ defines an ω-winning strategy for player 1 in $G_\Sigma^b(\mathcal{G}_2, \mathcal{G}_1)$ starting from $(\{u_0\} \mapsto w_0)$. In detail, let $w_0, w_1, \dots$ be the longest (and so possibly infinite) sequence of elements of $\Delta^{\mathcal{G}_1}$ such that, for each w_i, there exists u with $(u \mapsto w_i, i)$ a node in $\mathfrak{T}$. Note that, by (4), every w_i (if it exists) is uniquely determined. We set

$$\Xi_i = \{\, u \mid (u \mapsto w_i, i) \in \mathfrak{T} \,\}$$

and observe that $\Xi_0 = \{u_0\}$ and $\Xi_i = \Xi_{i-1}^{\rightsquigarrow}$ and $\Xi_i \neq \emptyset$, for all $i > 0$. Take the maximal $m < \omega$ such that w_m exists and $w_i \neq w_m$ for all $i < m$ (in other words, w_m is the first repeating element in the sequence). Now the strategy of player 1 is as follows: when challenged by player 2 with some $u \rightsquigarrow_2^\Sigma u'$ in state $(\Xi_i \mapsto w_i)$ with $i \leq m$, player 1 responds with w_{i+1} if $i < m$ and with the uniquely determined w_k, for $k \leq m$ and $w_k = w_{m+1}$, if $i = m$.

We now show that $\mathfrak{T}$ exists. Let $\mathbb{S}_0$ be the given set of n-winning backward strategies for player 1 in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ starting from $(u_0 \mapsto \sigma_0^n)$, for $\sigma_0^n \in \Delta^{\mathcal{M}_1}$ with $\mathsf{tail}(\sigma_0^n) = w_0$. Define $\mathfrak{T}_0$ to be the tree with the single node $(u_0 \mapsto w_0, 0)$. Clearly, it satisfies (1), (2) and (4). If it also satisfies (3), then we are done. Otherwise, we take a challenge $u_0 \rightsquigarrow_2^\Sigma u_1$ by player 2 and use the pigeonhole principle to find $w_1 \in \Delta^{\mathcal{G}_1}$ and a subset $\mathbb{S}_1 \subseteq \mathbb{S}_0$ such that, for any challenge $u_0 \rightsquigarrow_2^\Sigma u'$, every strategy $\mathcal{S} \in \mathbb{S}_1$ gives a response $(u' \mapsto \sigma')$ with $\mathsf{tail}(\sigma') = w_1$. We add to $\mathfrak{T}_0$ the nodes $(u' \mapsto w_1, 1)$, for any challenge $u_0 \rightsquigarrow_2^\Sigma u'$. We also add a $u_0 \rightsquigarrow_2^\Sigma u'$ arc connecting $(u_0 \mapsto w_0, 0)$ with the newly introduced nodes. This gives us the tree $\mathfrak{T}_1$ satisfying (1), (2) and (4). We proceed in this way and construct a sequence of trees $\mathfrak{T}_0 \subseteq \mathfrak{T}_1 \subseteq \dots$ until we either reach some $\mathfrak{T}_k$ satisfying (1)–(4) or obtain an infinite sequence and take $\mathfrak{T} = \bigcup_{k<\omega} \mathfrak{T}_k$, which obviously satisfies (1)–(4).

(b) $\Rightarrow$ (a) Suppose player 1 has an ω-winning strategy $\mathcal{S}$ starting from $(\{u_0\} \mapsto w_0)$ in the game $G_\Sigma^b(\mathcal{G}_2, \mathcal{G}_1)$ and let $n < \omega$. Recall that, for each state $(\Xi \mapsto w)$, there is (at most) one challenge $\Xi' = \Xi^{\rightsquigarrow}$. Thus, the first n rounds of a play according to $\mathcal{S}$ starting from $(\{u_0\} \mapsto w_0)$ are given by a sequence $(\Xi_0 \mapsto w_0), (\Xi_1 \mapsto w_1), \dots, (\Xi_k \mapsto w_k)$, where $\Xi_0 = \{u_0\}$ and either $k = n$ or $k < n$ and $\Xi_k^{\rightsquigarrow} = \emptyset$. Take any $\sigma \in \Delta^{\mathcal{M}_1}$ with $\mathsf{tail}(\sigma) = w_k$ and let $\sigma_0^n = \sigma w_{k-1} \cdots w_0$. Clearly, player 1 has an n-winning backward strategy in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ starting from $(u_0 \mapsto \sigma_0^n)$. ❑

Although Lemmas 17 and 21 look similar, the game $G_\Sigma^b(\mathcal{G}_2, \mathcal{G}_1)$ turns out to be more complex than $G_\Sigma^f(\mathcal{G}_2, \mathcal{G}_1)$ because the full game graph can be exponential in the size of $\Delta^{\mathcal{G}_2} \setminus \mathsf{ind}(\mathcal{K}_2)$. In fact, we have the following:

Lemma 22. *Checking whether player 1 has an ω-winning strategy in $G_\Sigma^b(\mathcal{G}_2, \mathcal{G}_1)$ 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 $p_1, \dots p_k$ or a negation of such a variable.

Let $N_1, \dots, N_k$ be the first k prime numbers (observe that $1 < N_k \leq k^2$). We take a role name R, a role name C_i, for each clause c_i in φ, and a role name $S_{j\ell}$, for each $1 \leq j \leq k$ and $1 \leq \ell \leq N_j$. Now we define a KB $\mathcal{K}_2 = (\mathcal{T}_2, \{A(a)\})$, where $\mathcal{T}_2$ contains $A \sqsubseteq \exists R$, the following inclusions, for $1 \leq j \leq k$ and $1 \leq \ell < N_j$,

$$\exists R^- \sqsubseteq \exists S_{j1}, \qquad \exists S_{j\ell}^- \sqsubseteq \exists S_{j\ell+1}, \qquad \exists S_{jN_j}^- \sqsubseteq \exists S_{j1},$$

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

$S_{j1} \sqsubseteq C_i$, if p_j is a literal of c_i,

$S_{j2} \sqsubseteq C_i$, if $\neg p_j$ is a literal of c_i.

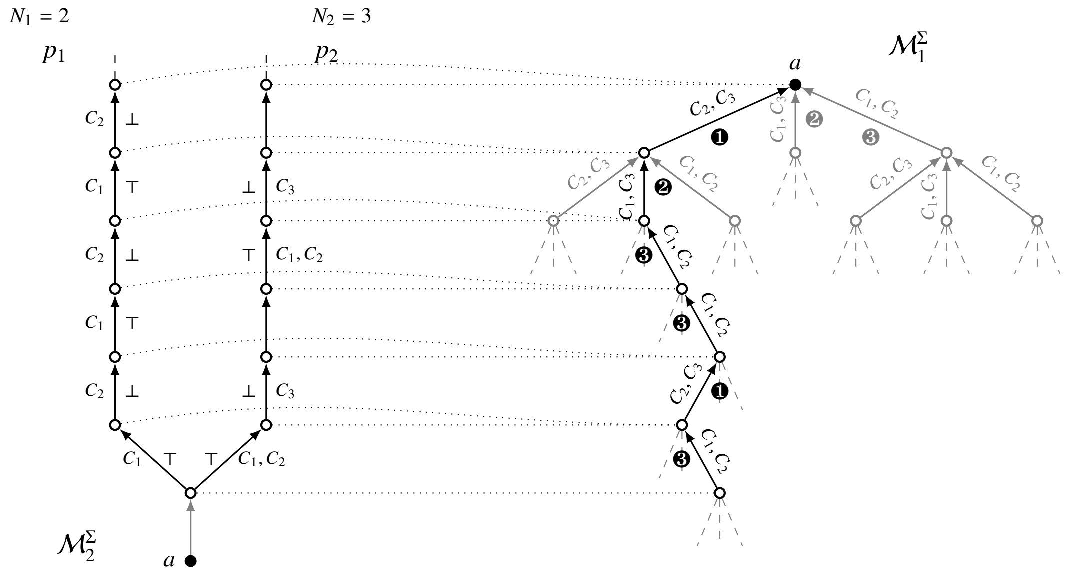


Figure 7: $\mathcal{M}_2^\Sigma$ and $\mathcal{M}_1^\Sigma$ for $\varphi = c_1 \wedge c_2 \wedge c_3$, where $c_1 = p_1 \vee p_2$, $c_2 = \neg p_1 \vee p_2$ and $c_3 = \neg p_2$. The $\top/\bot$ symbols on the arrows of $\mathcal{M}_2^\Sigma$ indicate the truth value of the respective variable. Only one branch of $\mathcal{M}_1^\Sigma$ is shown in full detail, with the index of the missing role C_i in the black circle next to the arrow.

Intuitively, $\mathcal{M}_2$ is a tree with k branches having a common root arrow R. The jth branch is obtained by unravelling a loop of N_j arrows $S_{j1}, \dots, S_{jN_j}$: the first arrow, S_{j1}, corresponds to p_j being true (under an assignment) and the second arrow, S_{j2}, to p_j being false (other arrows do not encode truth values). Therefore, $N_1 \times N_2 \times \cdots \times N_k$ layers (the layer i consists of all arrows from points at distance i from the root) contain representations of all possible assignments to $p_1, \dots, p_k$ (for $k = 2$, see Fig. 7 on the left). The last two types of role inclusions make sure that the roles $C_1, \dots, C_m$, which constitute the signature Σ, mark those assignments under which φ is true.

We now take $\mathcal{K}_1 = (\mathcal{T}_1, \{A(a)\})$, where $\mathcal{T}_1$ contains the following inclusions:

$$A \sqsubseteq \exists T_i \quad \text{and} \quad \exists T_i^- \sqsubseteq A, \qquad \text{for } 1 \leq i \leq m$$
$$T_i \sqsubseteq C_{i'}^-, \qquad \text{for } 1 \leq i \neq i' \leq m.$$

In $\mathcal{M}_1$, the path from each point to the root contains arrows that are labelled by all of $C_1, \dots, C_m$ but one (for $m = 3$, see Fig. 7 on the right). Note that the C_i arrows point towards the root, in the opposite direction to the C_i arrows of $\mathcal{M}_2$. Thus, there is a finite $(\Sigma, \{a\})$-homomorphism from $\mathcal{M}_2$ into $\mathcal{M}_1$ if and only if one of the clauses is false under each of the assignments (that is, iff φ is unsatisfiable).

The generating structure $\mathcal{G}_1$ is essentially a set of loops each of which is missing precisely one of the C_i. Thus, the responses of player 1 correspond to the choices of the missing C_i. The challenges by player 2, on the other hand, correspond to the subsets of $C_1, \dots, C_m$ in the layers of $\mathcal{M}_2$, the number of which may be exponential in k. So 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 repeats itself after $N_1 \times \cdots \times N_k$ steps. ❑

A general 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.

4.4. 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 $(u_0 \mapsto \sigma_0)$ is called *start-bounded* if it never leads to a state $(u_i \mapsto \sigma_i)$ such that $\sigma_0 = \sigma_i w$, for some $w \in \Delta^{\mathcal{G}_1}$ 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 23. The strategy starting from $(u_2 \mapsto \sigma_0)$ and shown in Fig. 8a by dotted lines is start-bounded, with the numbers indicating the rounds of the game: the responses $\sigma_0, \sigma_1, \sigma_2$ of player 1 move away from the ABox, after which player 1 retraces his steps back to σ_0 (in order to avoid clutter, we omitted the ABox part from the generating structure $\mathcal{G}_2$ in the picture).

The *states* of $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ are of the form $(\Theta_i, \Xi_i \mapsto w_i)$, $i \geq 0$, where $\Theta_i, \Xi_i \subseteq \Delta^{\mathcal{G}_2}$, $\Xi_i \neq \emptyset$ and $w_i \in \Delta^{\mathcal{G}_1}$. (Intuitively, Ξ_i is the set of elements of $\Delta^{\mathcal{G}_2}$ that are mapped to w_i, while Θ_i identifies illegitimate challenges for player 2, that is, the $\rightsquigarrow^\Sigma_2$-successors that have already been mapped to w_{i-1}.) The initial state is of the form $(\emptyset, \Xi_0 \mapsto w_0)$. 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

$$\text{if} \quad v \in \Theta_{i-1} \quad \text{then} \quad \boldsymbol{r}^{\mathcal{G}_2}_\Sigma(u, v) \not\subseteq \bar{\boldsymbol{r}}^{\mathcal{G}_1}_\Sigma(w_{i-2}, w_{i-1}). \qquad \textbf{(no-backward)}$$

(Player 2 loses if there is no challenge satisfying this condition.) Player 1 'guesses' some Ξ_i and w_i such that Ξ_i contains v, $\boldsymbol{r}^{\mathcal{G}_2}_\Sigma(u, v) \subseteq \boldsymbol{r}^{\mathcal{G}_1}_\Sigma(w_{i-1}, w_i)$ and responds with a state $(\Theta_i, \Xi_i \mapsto w_i)$, where Θ_i is determined by Ξ_{i-1} and w_i: $\Theta_i = \Xi_{i-1}$ if $w_i \notin \mathsf{ind}(\mathcal{K}_1)$ and $\Theta_i = \emptyset$, otherwise. We make challenges $u \rightsquigarrow^\Sigma_2 v$, for which

$$u \in \Xi_{i-1}, \qquad v \in \Theta_{i-1} \quad \text{and} \quad \boldsymbol{r}^{\mathcal{G}_2}_\Sigma(u, v) \subseteq \bar{\boldsymbol{r}}^{\mathcal{G}_1}_\Sigma(w_{i-2}, w_{i-1}),$$

'illegitimate' because, by the choice of Ξ_{i-2}, the element w_{i-2} was supposed to be used as a response; note that the last two conditions above are the complement of (**no-backward**). Because of this, player 1 always moves 'forward' in $\mathcal{G}_1$, but has to guess appropriate sets Ξ_i in advance. The states, initial states, challenges by player 2 and responses of player 1 are summarised in the table below:

start-bounded game $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$	
states, $i \geq 0$	$(\Theta_i, \Xi_i \mapsto w_i)$ with $\Theta_i, \Xi_i \subseteq \Delta^{\mathcal{G}_2}$, $\Xi_i \neq \emptyset$, $w_i \in \Delta^{\mathcal{G}_1}$ and $\boldsymbol{t}^{\mathcal{G}_2}_\Sigma(u) \subseteq \boldsymbol{t}^{\mathcal{G}_1}_\Sigma(w_i)$, for all $u \in \Xi_i$
initial state	$(\emptyset, \Xi_0 \mapsto w_0)$ such that $w_0 = u$ in case $u \in \Xi_0 \cap \mathsf{ind}(\mathcal{K}_2) \cap \mathsf{part}^{\mathcal{M}_2}_\Sigma$ and $\Xi_0 \cap \mathsf{ind}(\mathcal{K}_2)$ contains at most one element
challenges, $i > 0$	$u \rightsquigarrow^\Sigma_2 v$ such that $u \in \Xi_{i-1}$ and if $v \in \Theta_{i-1}$, for $i > 1$, then $\boldsymbol{r}^{\mathcal{G}_2}_\Sigma(u, v) \not\subseteq \bar{\boldsymbol{r}}^{\mathcal{G}_1}_\Sigma(w_{i-2}, w_{i-1})$
responses, $i > 0$	$(\Theta_i, \Xi_i \mapsto w_i)$ such that $v \in \Xi_i$ and $\Xi_i \cap \mathsf{ind}(\mathcal{K}_2) = \emptyset$, either $w_{i-1} \rightsquigarrow_1 w_i$ and $\Theta_i = \Xi_{i-1}$ or $w_{i-1}, w_i \in \mathsf{ind}(\mathcal{K}_1)$ and $\Theta_i = \emptyset$, and $\boldsymbol{r}^{\mathcal{G}_2}_\Sigma(u, v) \subseteq \boldsymbol{r}^{\mathcal{G}_1}_\Sigma(w_{i-1}, w_i)$

Note that of all Ξ_i only Ξ_0 may contain (at most one) individual from $\mathsf{ind}(\mathcal{K}_2)$; $\Theta_0 = \emptyset$ and of all Θ_i only Θ_1 may contain an individual.

Example 24. Consider $\mathcal{G}^\Sigma_2$ and $\mathcal{G}^\Sigma_1$ in Fig. 8b. 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 $\mathcal{M}_1$. We show that player 1 has an ω-winning strategy in $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ starting from $(\emptyset, \{u_2, u_9\} \mapsto w_0)$. Player 2 challenges with $u_2 \rightsquigarrow^\Sigma_2 u_6$, and player 1 responds with $(\{u_2, u_9\}, \{u_6, u_8\} \mapsto w_1)$. Then player 2 picks $u_6 \rightsquigarrow^\Sigma_2 u_7$ and player 1 responds with $(\{u_6, u_8\}, \{u_7\} \mapsto w_2)$, where the game ends because player 2 has no challenge available. Observe that this strategy involves only 3 rounds in contrast to the 5 rounds of the corresponding strategy in $G(\mathcal{G}_2, \mathcal{M}_1)$ shown in Fig. 8a. The strategy in $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ is indicated by the shaded states of the fragment of the game graph in Fig. 8c. Note the crucial guesses $\{u_2, u_9\} \mapsto w_0$ and $\{u_6, u_8\} \mapsto w_1$ made by player 1. For example, if player 1 responded with $(\{u_2, u_9\}, \{u_6\} \mapsto w_1)$ (and failed to guess that u_8 must also be mapped to w_1), then after the challenge $u_6 \rightsquigarrow^\Sigma_2 u_7$ and the only possible response $(\{u_6\}, \{u_7\} \mapsto w_2)$, player 2 would pick $u_7 \rightsquigarrow^\Sigma_2 u_8$ to which player 1 would not have a response; see the non-shaded states in Fig. 8c.

Lemma 25. *Conditions* (**win-*s***) *and* (**ω-win^s**) *are equivalent. More precisely, for any $u_0 \in \Delta^{\mathcal{G}_2}$ and $\sigma_0 \in \Delta^{\mathcal{M}_1}$, the following are equivalent*:

(*a*) *player 1 has an ω-winning start-bounded strategy in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ starting from $(u_0 \mapsto \sigma_0)$*;

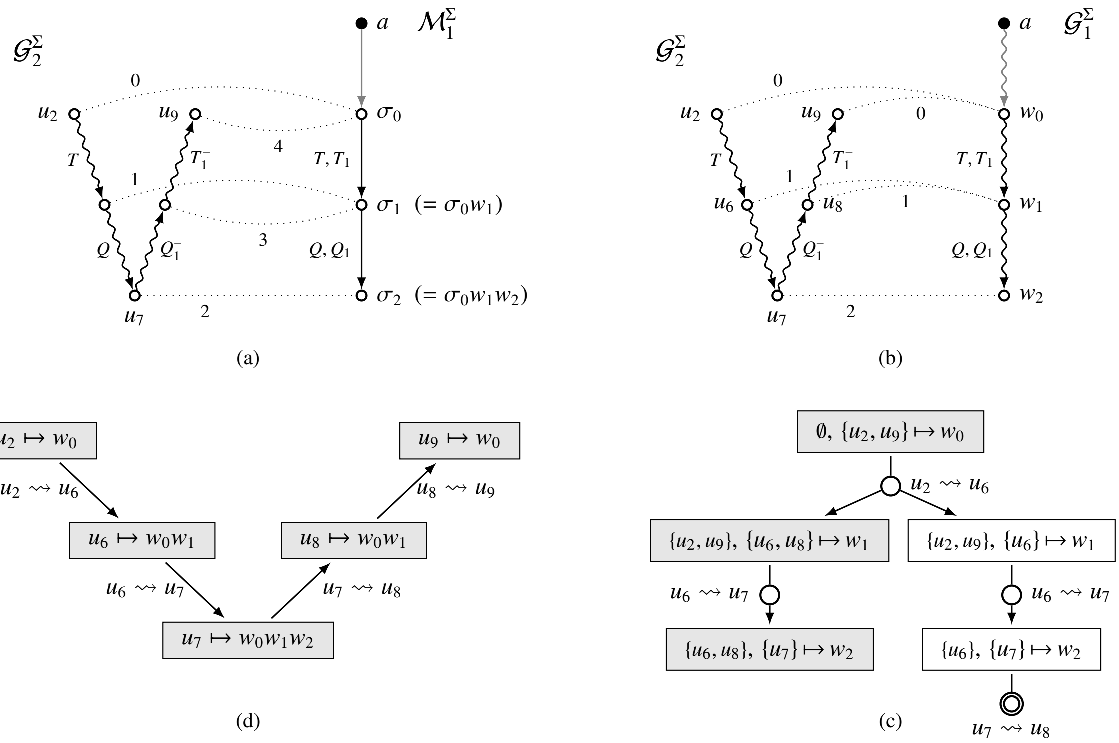


Figure 8: Example 23: (a) an ω-winning start-bounded strategy in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ from $(u_2 \mapsto \sigma_0)$; (b) an ω-winning strategy in $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ from $(\emptyset, \{u_2, u_9\} \mapsto w_0)$; (c) the respective fragment of the game graph of $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$; (d) the graph $\mathfrak{T}$ for extracting ω-winning strategies in $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$.

(*b*) *for every $n < \omega$, player 1 has an n-winning start-bounded strategy in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ starting from $(u_0 \mapsto \sigma_0)$;*

(*c*) *player 1 has an ω-winning strategy in $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ starting from $(\emptyset, \Xi_0 \mapsto \mathsf{tail}(\sigma_0))$, for some $\Xi_0 \ni u_0$.*

Proof. $(a) \Rightarrow (b)$ is trivial.

$(b) \Rightarrow (c)$ We define a (possibly infinite) directed graph $\mathfrak{T}$ whose nodes are of the form $(u \mapsto \delta)$, where $u \in \Delta^{\mathcal{G}_2}$ and δ is a suffix of some element in $\Delta^{\mathcal{M}_1}$, and whose arrows are labelled with $u \rightsquigarrow^\Sigma_2 u'$ so that the following conditions hold:

(1) $\mathfrak{T}$ contains an initial node $(u_0 \mapsto \mathsf{tail}(\sigma_0))$;

(2) $\boldsymbol{t}^{\mathcal{G}_2}_\Sigma(u) \subseteq \boldsymbol{t}^{\mathcal{G}_1}_\Sigma(\mathsf{tail}(\delta))$, for every node $(u \mapsto \delta)$ in $\mathfrak{T}$;

(3) for any $u \rightsquigarrow^\Sigma_2 u'$, every node $(u \mapsto \delta)$ in $\mathfrak{T}$ has exactly one $(u \rightsquigarrow^\Sigma_2 u')$-successor in $\mathfrak{T}$, which can be of the following forms:

(3.1) $(u' \mapsto \delta w')$, if $\mathsf{tail}(\delta) = w$, $w \rightsquigarrow_1 w'$ and $\boldsymbol{r}^{\mathcal{G}_2}_\Sigma(u, u') \subseteq \boldsymbol{r}^{\mathcal{G}_1}_\Sigma(w, w')$;

(3.2) $(u' \mapsto b)$, if $\delta = a \in \mathsf{ind}(\mathcal{K}_1)$, $b \in \mathsf{ind}(\mathcal{K}_1)$ and $\boldsymbol{r}^{\mathcal{G}_2}_\Sigma(u, u') \subseteq \boldsymbol{r}^{\mathcal{G}_1}_\Sigma(a, b)$;

(3.3) $(u' \mapsto \delta')$, if $\delta = \delta' w$, $\mathsf{tail}(\delta') = w'$, $w' \rightsquigarrow_1 w$ and $\boldsymbol{r}^{\mathcal{G}_2}_\Sigma(u, u') \subseteq \bar{\boldsymbol{r}}^{\mathcal{G}_1}_\Sigma(w', w)$.

Observe that these conditions coincide with the conditions given in the proof of Lemma 17 except that now (3.3) provides a possibility of going backward. The graph $\mathfrak{T}$ for the winning strategy in Example 23 is depicted in Fig. 8d.

We show that the graph $\mathfrak{T}$ (if it exists) gives rise to the required ω-winning strategy for player 1 in $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$. Consider the function s mapping the nodes in $\mathfrak{T}$ to states in the game $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ and defined by taking

$$\mathsf{s}(u \mapsto \delta) = \begin{cases} (\Xi_{\delta'}, \Xi_\delta \mapsto \mathsf{tail}(\delta)), & \text{if } \delta = \delta' w, \\ (\emptyset, \Xi_\delta \mapsto \delta), & \text{otherwise (that is, if } \delta = \mathsf{tail}(\sigma_0) \text{ or } \delta \in \mathsf{ind}(\mathcal{K}_1)), \end{cases}$$

where $\Xi_\delta = \{u \mid (u \mapsto \delta)$ a node in $\mathfrak{T}\}$. In particular, the initial node $\mathfrak{n}_0$ in $\mathfrak{T}$ is mapped to the initial state: $\mathsf{s}(\mathfrak{n}_0) = (\emptyset, \Xi_{\mathsf{tail}(\sigma_0)} \mapsto \mathsf{tail}(\sigma_0))$. (Note that only $\mathfrak{n}_0$ may refer to an individual from $\mathsf{ind}(\mathcal{K}_2)$, and so $\mathsf{s}(\mathfrak{n}_0)$ is a properly defined initial state.) In order to define the ω-winning strategy of player 1 in $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ from $\mathsf{s}(\mathfrak{n}_0)$, we show that, for all $\mathfrak{n}$ in $\mathfrak{T}$,

$$\text{if player 2 has a challenge } u \rightsquigarrow^\Sigma_2 u' \text{ in } \mathsf{s}(\mathfrak{n}), \text{ then there is } \mathfrak{r}_u \text{ and a } (u \rightsquigarrow^\Sigma_2 u')\text{-successor } \mathfrak{n}' \text{ of } \mathfrak{r}_u \text{ in } \mathfrak{T}$$
$$\text{such that } \mathsf{s}(\mathfrak{r}_u) = \mathsf{s}(\mathfrak{n}) \text{ and } \mathsf{s}(\mathfrak{n}') \text{ is a valid response by player 1 to } u \rightsquigarrow^\Sigma_2 u' \text{ in } \mathsf{s}(\mathfrak{n}).$$

Indeed, if $u \rightsquigarrow^\Sigma_2 u'$ is a challenge in $\mathsf{s}(\mathfrak{n})$ then $\mathsf{s}(\mathfrak{n})$ is of the form $(\Theta, \Xi_\delta \mapsto \mathsf{tail}(\delta))$, for some δ and $u \in \Xi_\delta$. By definition, $\mathfrak{T}$ contains a node $\mathfrak{r}_u = (u \mapsto \delta)$ and $\mathsf{s}(\mathfrak{r}_u) = \mathsf{s}(\mathfrak{n})$; moreover, $\mathfrak{r}_u$ has a $(u \rightsquigarrow^\Sigma_2 u')$-successor $\mathfrak{n}'$ in $\mathfrak{T}$. (Observe that, by the definition of s, for two distinct nodes $\mathfrak{n} = (v \mapsto \delta)$ and $\mathfrak{r}_u = (u \mapsto \delta)$, we may have $\mathsf{s}(\mathfrak{n}) = \mathsf{s}(\mathfrak{r}_u) = (\Theta, \Xi_\delta \mapsto \mathsf{tail}(\delta))$ and $\{u, v\} \subseteq \Xi_\delta$, and so $\mathfrak{T}$ may contain a node $\mathfrak{n}$ that has no $u \rightsquigarrow^\Sigma_2 u'$ successor for a valid challenge $u \rightsquigarrow^\Sigma_2 u'$ in $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ from $\mathsf{s}(\mathfrak{n})$. Similarly to the proof of Lemma 17, the choice of a particular $\mathfrak{r}_u$ is not essential.) It remains to show that $\mathsf{s}(\mathfrak{n}')$ is a valid response by player 1 to $u \rightsquigarrow^\Sigma_2 u'$ from $\mathsf{s}(\mathfrak{n})$. Consider all possible cases:

- If $\mathfrak{r}_u = (u \mapsto w)$ and $\mathfrak{n}' = (u' \mapsto ww')$ then $\mathsf{s}(\mathfrak{n}) = (\emptyset, \Xi_w \mapsto w)$ and $\mathsf{s}(\mathfrak{n}') = (\Xi_w, \Xi_{ww'} \mapsto w')$. By item (3.1) of the definition of $\mathfrak{T}$, $\mathsf{s}(\mathfrak{n}')$ is as required.

- If $\mathfrak{r}_u = (u \mapsto \delta w)$ and $\mathfrak{n}' = (u' \mapsto \delta ww')$ then $\mathsf{s}(\mathfrak{n}) = (\Xi_\delta, \Xi_{\delta w} \mapsto w)$ and $\mathsf{s}(\mathfrak{n}') = (\Xi_{\delta w}, \Xi_{\delta ww'} \mapsto w')$. By (3.1), $\mathsf{s}(\mathfrak{n}')$ is as required.

- If $\mathfrak{r}_u = (u \mapsto w)$ and $\mathfrak{n}' = (u' \mapsto w')$ then $w, w' \in \mathsf{ind}(\mathcal{K}_1)$, $\mathsf{s}(\mathfrak{n}) = (\emptyset, \Xi_w \mapsto w)$ and $\mathsf{s}(\mathfrak{n}') = (\emptyset, \Xi_{w'} \mapsto w')$. By (3.2), $\mathsf{s}(\mathfrak{n}')$ is as required.

- If $\mathfrak{r}_u = (u \mapsto \delta w' w)$ and $\mathfrak{n}' = (u' \mapsto \delta w')$ then $\mathsf{s}(\mathfrak{n}) = (\Xi_{\delta w'}, \Xi_{\delta w' w} \mapsto w)$ and $u' \in \Xi_{\delta w'}$, which is impossible because, in view of (3.3), we have $\boldsymbol{r}^{\mathcal{G}_2}_\Sigma(u, u') \subseteq \bar{\boldsymbol{r}}^{\mathcal{G}_1}_\Sigma(w', w)$ contrary to the fact that $u \rightsquigarrow^\Sigma_2 u'$ is a challenge in $\mathsf{s}(\mathfrak{n})$; see (**no-backward**).

The ω-winning strategy of player 1 in $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ from $\mathsf{s}(\mathfrak{n}_0)$ is then defined naturally.

Now we show that $\mathfrak{T}$ exists. The construction is similar to the proof of Lemma 17. Let $\mathbb{S}_0$ be the given set of n-winning start-bounded strategies in $G_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ starting from $(u_0 \mapsto \sigma_0)$ and let $w_0 = \mathsf{tail}(\sigma_0)$. Define $\mathfrak{T}_0$ to be the graph with the single initial node $(u_0 \mapsto w_0)$. Clearly, it satisfies (1) and (2) above. If it also satisfies (3), then we are done. Otherwise, as in the proof of Lemma 17, we take all the challenges $u_0 \rightsquigarrow^\Sigma_2 u^1_1, \dots, u_0 \rightsquigarrow^\Sigma_2 u^k_1$ by player 2 and using the pigeonhole principle find $w^1_1, \dots, w^k_1 \in \Delta^{\mathcal{G}_1}$ and a set $\mathbb{S}_1 \subseteq \mathbb{S}_0$ such that, for any challenge $u_0 \rightsquigarrow^\Sigma_2 u^i_1$, every strategy $\mathcal{S} \in \mathbb{S}_1$ gives a response $(u^i_1 \mapsto \sigma^i_1)$ with $\mathsf{tail}(\sigma^i_1) = w^i_1$. If $w^i_1 \in \mathsf{ind}(\mathcal{K}_1)$ then we add the node $(u^i_1 \mapsto w^i_1)$ to $\mathfrak{T}_0$, and if $w^i_1 \notin \mathsf{ind}(\mathcal{K}_1)$ then we add the node $(u^i_1 \mapsto w_0 w^i_1)$ to $\mathfrak{T}_0$; we also add an $u_0 \rightsquigarrow^\Sigma_2 u^i_1$ arrow connecting $(u_0 \mapsto w_0)$ with the newly introduced node. This gives us the graph $\mathfrak{T}_1$. To illustrate the construction of $\mathfrak{T}$ in the case of a backward step (which is impossible in round 1), consider now a challenge $u_1 \rightsquigarrow^\Sigma_2 u_2$ by player 2 for some $u_1 \in \{u^1_1, \dots, u^k_1\}$ such that the response according to $\mathcal{S}$ was $(u_1 \mapsto \sigma_0 w_1)$ and $(u_1 \mapsto w_0 w_1)$ is a node in $\mathfrak{T}_1$. Then, using the pigeonhole principle, we find either

- $w_2 \in \Delta^{\mathcal{G}_1}$ and a subset $\mathbb{S}_2 \subseteq \mathbb{S}_1$ such that every strategy $\mathcal{S} \in \mathbb{S}_2$ gives a response of the form $(u_2 \mapsto \sigma_0 w_1 w_2)$,

- or a subset $\mathbb{S}_2 \subseteq \mathbb{S}_1$ such that every strategy $\mathcal{S} \in \mathbb{S}_2$ gives a response of the form $(u_2 \mapsto \sigma_0)$.

In the former case we add the node $(u_2 \mapsto w_0 w_1 w_2)$ to $\mathfrak{T}_1$ and in the latter case we add $(u_2 \mapsto w_0)$ to $\mathfrak{T}_1$. We also add an $u_1 \rightsquigarrow^\Sigma_2 u_2$ arrow connecting $(u_1 \mapsto w_0 w_1)$ and the new node to $\mathfrak{T}_1$. This defines $\mathfrak{T}_2$. We proceed in the same way and construct a sequence of graphs $\mathfrak{T}_0 \subseteq \mathfrak{T}_1 \subseteq \dots$ until we either reach some $\mathfrak{T}_k$ satisfying (1)–(3) or obtain an infinite sequence and take $\mathfrak{T} = \bigcup_{k<\omega} \mathfrak{T}_k$, which obviously satisfies (1)–(3).

$(c) \Rightarrow (a)$ Suppose that player 1 has an ω-winning strategy $\mathcal{S}$ in $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ starting from $(\emptyset, \Xi_0 \mapsto \mathsf{tail}(\sigma_0))$ with $u_0 \in \Xi_0$. We transform the strategy $\mathcal{S}$ into an ω-winning start-bounded strategy $\mathcal{S}'$ in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ starting from $\mathfrak{s}_0 = (u_0 \mapsto \sigma_0)$. We associate with any (possibly infinite) sequence $u_0 \rightsquigarrow^\Sigma_2 u_1 \rightsquigarrow^\Sigma_2 \cdots \rightsquigarrow^\Sigma_2 u_i \rightsquigarrow^\Sigma_2 \cdots$ of challenges by player 2 in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ starting from the state $\mathfrak{s}_0$ a sequence $\mathfrak{s}_1 = (u_1 \mapsto \sigma_1), \dots, \mathfrak{s}_i = (u_i \mapsto \sigma_i), \dots$ of responses by player 1 which are start-bounded (that is, $\sigma_0 \neq \sigma_i w$, for any $w \in \Delta^{\mathcal{G}_1}$). To this end, we also define a sequence of states $\mathfrak{sb}_0 = (\Theta_0, \Xi_0 \mapsto w_0), \dots, \mathfrak{sb}_i = (\Theta_i, \Xi_i \mapsto w_i), \dots$ in $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ such that $u_i \in \Xi_i$ and $\mathsf{tail}(\sigma_i) = w_i$ for all i. To keep track of 'backward moves' we also define a sequence $\pi_0, \dots, \pi_i, \dots$ of sequences of states in $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ such that each π_i has length $|\sigma_i| + 1 - |\sigma_0|$ and its first state is of the form $(\emptyset, \Xi \mapsto w)$. Finally, we require that

$$\text{if} \quad \pi_i = \pi_j \cdot (\Theta^1, \Xi^1 \mapsto w^1) \cdots (\Theta^m, \Xi^m \mapsto w^m) \quad \text{then} \quad \sigma_i = \sigma_j w^1 \cdots w^m. \tag{4}$$

For $i = 0$, we set $\mathfrak{sb}_0 = (\emptyset, \Xi_0 \mapsto w_0)$ and $\pi_0 = \mathfrak{sb}_0$, which clearly has the required properties. Now assume that $\mathfrak{s}_0, \dots, \mathfrak{s}_{i-1}$, $\mathfrak{sb}_0, \dots, \mathfrak{sb}_{i-1}$ and $\pi_0, \dots, \pi_{i-1}$, for $i > 0$, are defined as above. Consider a challenge $u_{i-1} \rightsquigarrow^\Sigma_2 u_i$ in state $\mathfrak{s}_{i-1}$. We distinguish the following two cases.

- If $u_{i-1} \rightsquigarrow^\Sigma_2 u_i$ is a valid challenge in $\mathfrak{sb}_{i-1}$ then we define $\mathfrak{sb}_i = (\Theta_i, \Xi_i \mapsto w_i)$ as the response of player 1 in $\mathfrak{sb}_{i-1}$ according to $\mathcal{S}$. If $w_i \notin \mathsf{ind}(\mathcal{K}_1)$ then we set $\pi_i = \pi_{i-1} \cdot \mathfrak{sb}_i$ and $\mathfrak{s}_i = (u_i \mapsto \sigma_{i-1} w_i)$. Otherwise, $\Theta_i = \emptyset$ and we set $\pi_i = \mathfrak{sb}_i$ and $\mathfrak{s}_i = (u_i \mapsto w_i)$. Obviously, the conditions above hold for the resulting sequences.
- If $u_{i-1} \rightsquigarrow^\Sigma_2 u_i$ is not a valid challenge from $\mathfrak{sb}_{i-1}$ then $\Theta_{i-1} \neq \emptyset$, $u_i \in \Theta_{i-1}$ and $\boldsymbol{r}^{\mathcal{G}_2}_\Sigma(u_{i-1}, u_i) \subseteq \bar{\boldsymbol{r}}^{\mathcal{G}_1}_\Sigma(w, w_{i-1})$ for the predecessor w of w_{i-1} in σ_{i-1}. Let π_i be the result of removing the final state from π_{i-1}; let $\mathfrak{sb}_i$ be the final element of π_i; and let $\mathfrak{s}_i = (u_i \mapsto \sigma_i)$, where σ_i is obtained from σ_{i-1} by removing its final element. Clearly, (4) is satisfied. We show that $\mathfrak{s}_i$ is a valid response. First, observe that there exists $j \leq i - 2$ such that $\pi_j = \pi_i$ and $\pi_{j+1} = \pi_{i-1}$ for which $\mathfrak{sb}_{j+1}$ is the response to the challenge $u_j \rightsquigarrow^\Sigma_2 u_{j+1}$ from $\mathfrak{sb}_j$. By (4), $\sigma_{j+1} = \sigma_{i-1}$ and $\sigma_{j+1} = \sigma_j w_j$. By the construction of σ_i, $\sigma_i = \sigma_j$. Second, it remains to observe that $\Theta_{j+1} = \Theta_{i-1}$ and $\Theta_{j+1} = \Xi_j$, i.e., $u_i \in \Xi_j$ and $\boldsymbol{t}^{\mathcal{G}_2}_\Sigma(u_i) \subseteq \boldsymbol{t}^{\mathcal{G}_1}_\Sigma(w_j) = \boldsymbol{t}^{\mathcal{M}_1}_\Sigma(\sigma_i)$ (recall that, by (**no-backward**), $\boldsymbol{r}^{\mathcal{G}_2}_\Sigma(u_{i-1}, u_i) \subseteq \boldsymbol{r}^{\mathcal{M}_1}_\Sigma(\sigma_{i-1}, \sigma_i)$).

By repeating these steps, we obtain an ω-winning start-bounded strategy in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ starting from $(u_0 \mapsto \sigma_0)$. ❑

Similarly to $G^f_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$, player 1 has an ω-winning strategy in $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ starting from a state $\mathfrak{s}$ if and only if player 2 does not have a winning strategy in the reachability game on the full graph of $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ starting from $\mathfrak{s}$. However, now the size of the game graph is exponential in the size of $\mathcal{G}_2$. More precisely, each Θ_i and Ξ_i is a subset of $\Delta^{\mathcal{G}_2}$ with at most one individual name, which results in $O((|\mathsf{ind}(\mathcal{K}_2)| \times 2^{|\Delta^{\mathcal{G}_2} \setminus \mathsf{ind}(\mathcal{K}_2)|})^2 \times |\Delta^{\mathcal{G}_1}|)$ states in $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$. The number of vertices in the graph for the reachability game is then cubic in the number of states in $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ because (**no-backward**) involves three states. So the existence of the required ω-winning strategy for player 1 can be checked in time polynomial in $\mathcal{G}_1$ but exponential in $\mathcal{G}_2$. Moreover, as we shall see in Section 5, this problem is ExpTime-hard.

4.5. General Strategies and Game $G^g_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$

A general 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 26. Consider $\mathcal{G}^\Sigma_2$ and $\mathcal{M}^\Sigma_1$ shown in Fig. 9a. Starting from $(u_1 \mapsto \sigma_3)$, 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 23; 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 states, initial states, challenges and responses in the general game $G^g_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ are defined in the table below:

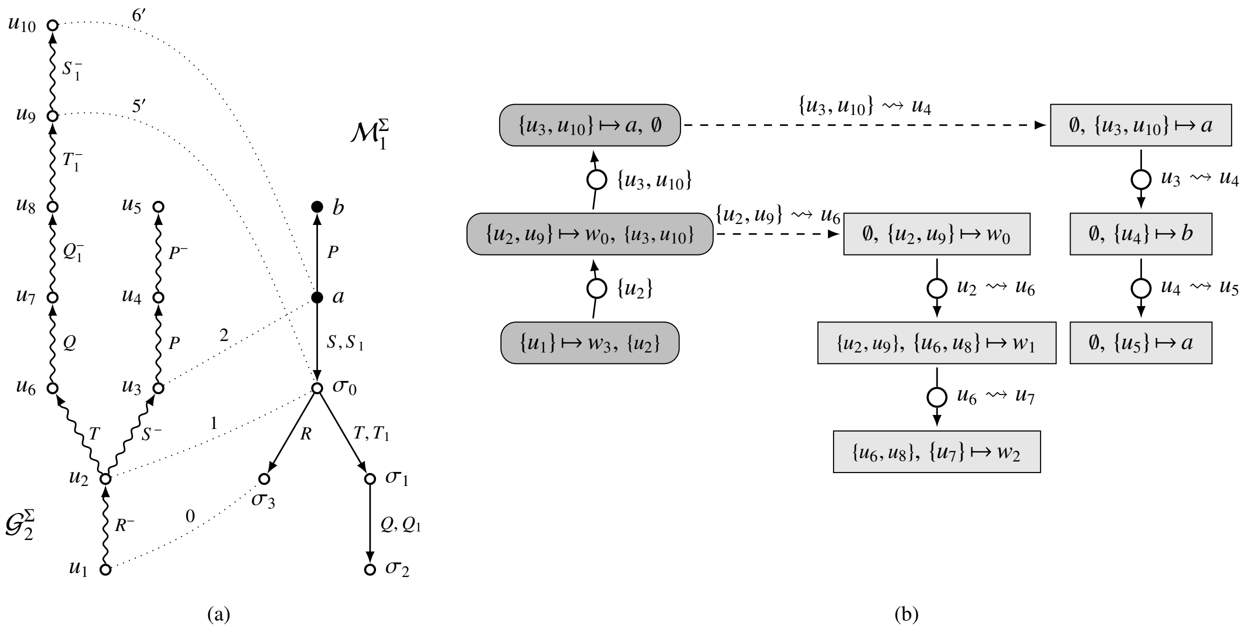


Figure 9: Example 26: (a) an ω-winning general strategy in the infinite game $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ from $(u_1 \mapsto \sigma_3)$; (b) the respective fragment of the game graph of $G^g_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ with graphs of the start-bounded games.

general game $G^g_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$	
states, $i \geq 0$	$(\Xi_i \mapsto w_i, \Psi_i)$ such that $\Xi_i \subseteq \Delta^{\mathcal{G}_2}$, $\Xi_i \neq \emptyset$, $w_i \in \Delta^{\mathcal{G}_1}$, $\Psi_i \subseteq \Xi_i^{\rightsquigarrow}$, $\boldsymbol{t}^{\mathcal{G}_2}_\Sigma(u) \subseteq \boldsymbol{t}^{\mathcal{G}_1}_\Sigma(w_i)$, for all $u \in \Xi_i$, $\Psi_i = \emptyset$ if $w_i \in \mathsf{ind}(\mathcal{K}_1)$, and player 1 *has an ω-winning strategy* in the start-bounded game $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ from $(\emptyset, \Xi_i \mapsto w_i)$ with the first challenge $u \rightsquigarrow^\Sigma_2 v$ by player 2 satisfying $v \in \Xi_i^{\rightsquigarrow} \setminus \Psi_i$
initial state	$(\Xi_0 \mapsto w_0, \Psi_0)$ such that $w_0 = u$ in case $u \in \Xi_0 \cap \mathsf{ind}(\mathcal{K}_2) \cap \mathsf{part}^{\mathcal{M}_2}_\Sigma$, $\Xi_0 \cap \mathsf{ind}(\mathcal{K}_2)$ contains at most one element
challenges, $i > 0$	Ψ_{i-1} provided that $\Psi_{i-1} \neq \emptyset$
responses, $i > 0$	$(\Xi_i \mapsto w_i, \Psi_i)$ such that $w_i \rightsquigarrow_1 w_{i-1}$, $\Xi_i \supseteq \Psi_{i-1}$ with $\Xi_i \cap \mathsf{ind}(\mathcal{K}_2) = \emptyset$, $\boldsymbol{r}^{\mathcal{G}_2}_\Sigma(u, v) \subseteq \bar{\boldsymbol{r}}^{\mathcal{G}_1}_\Sigma(w_i, w_{i-1})$, for all $u \in \Xi_{i-1}$ and $v \in \Xi_i$.

Thus, in every round $i > 0$ of the game, player 1 chooses a set $\Xi_i \supseteq \Psi_{i-1}$ and partitions the elements of $\Xi_i^{\rightsquigarrow}$ into those that will be mapped according to the backward strategy in round $i+1$ (the set Ψ_i) and those that will be mapped according to the start-bounded strategy (the set $\Xi_i^{\rightsquigarrow} \setminus \Psi_i$). Note the additional condition that player 1 *must* have an ω-winning strategy in the start-bounded game $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ from $(\emptyset, \Xi_i \mapsto w_i)$ where the first challenge by player 2 is restricted to $\Xi_i^{\rightsquigarrow} \setminus \Psi_i$.

Example 27. Figure 9b shows an ω-winning strategy for player 1 in $G^g_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ starting from $(\{u_1\} \mapsto w_3, \{u_2\})$, where $\mathcal{G}_1^\Sigma$ looks like $\mathcal{M}_1^\Sigma$ but with w_i in place of σ_i. The dashed transitions represent two launches of start-bounded games: one from the state $(\emptyset, \{u_2, u_9\} \mapsto w_0)$ with the initial challenge $u_2 \rightsquigarrow^\Sigma_2 u_6$, and the other from the state $(\emptyset, \{u_3, u_{10}\} \mapsto a)$ with the initial challenge $u_3 \rightsquigarrow^\Sigma_2 u_4$.

Lemma 28. *Conditions* **(win)***,* **(win-g)** *and* **(ω-win^g)** *are equivalent. More precisely, for any $u_0 \in \Delta^{\mathcal{G}_2}$ and $w_0 \in \Delta^{\mathcal{G}_1}$, the following are equivalent*:

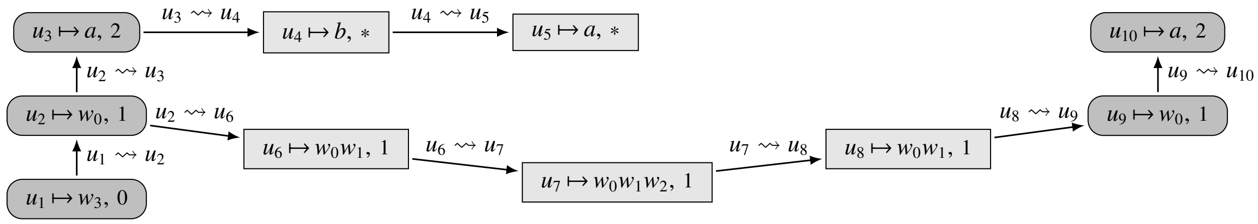


Figure 10: The graph $\mathfrak{T}$ for extracting ω-winning strategies in $G^g_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ from Example 26.

(a) *for every $n < \omega$, there is $\sigma^n_0 \in \Delta^{\mathcal{M}_1}$ such that $\mathsf{tail}(\sigma^n_0) = w_0$ and player 1 has an n-winning strategy in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ starting from $(u_0 \mapsto \sigma^n_0)$;*

(b) *player 1 has an ω-winning strategy in $G^g_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ starting from $(\Xi_0 \mapsto w_0, \Psi_0)$, for some $\Xi_0 \ni u_0$ and Ψ_0.*

Proof. (a) ⇒ (b) As before, we construct a (possibly infinite) directed graph $\mathfrak{T}$ whose nodes are of the form $(u \mapsto \delta, i)$, where $u \in \Delta^{\mathcal{G}_2}$, δ is a suffix of some element in $\Delta^{\mathcal{M}_1}$ and $0 \leq i < \omega$ or $i = *$, and whose arrows are labelled with $u \rightsquigarrow^\Sigma_2 u'$ and such that the following conditions hold:

(1) the initial node of $\mathfrak{T}$ is of the form $(u_0 \mapsto w_0, 0)$;

(2) $\boldsymbol{t}^{\mathcal{G}_2}_\Sigma(u) \subseteq \boldsymbol{t}^{\mathcal{G}_1}_\Sigma(\mathsf{tail}(\delta))$, for any node $(u \mapsto \delta, k)$ in $\mathfrak{T}$;

(3) for any $u \rightsquigarrow^\Sigma_2 u'$, every node $(u \mapsto \delta, i)$ in $\mathfrak{T}$ has exactly one $(u \rightsquigarrow^\Sigma_2 u')$-successor in $\mathfrak{T}$, which can be of the following forms:

(3.1) $(u' \mapsto \delta w', i)$, if $\mathsf{tail}(\delta) = w$, $w \rightsquigarrow_1 w'$ and $\boldsymbol{r}^{\mathcal{G}_2}_\Sigma(u, u') \subseteq \boldsymbol{r}^{\mathcal{G}_1}_\Sigma(w, w')$;

(3.2) $(u' \mapsto b, *)$, if $\delta = a \in \mathsf{ind}(\mathcal{K}_1)$, $b \in \mathsf{ind}(\mathcal{K}_1)$ and $\boldsymbol{r}^{\mathcal{G}_2}_\Sigma(u, u') \subseteq \boldsymbol{r}^{\mathcal{G}_1}_\Sigma(a, b)$;

(3.3) $(u' \mapsto \delta', i)$, if $\delta = \delta' w$, $\mathsf{tail}(\delta') = w'$, $w' \rightsquigarrow_1 w$ and $\boldsymbol{r}^{\mathcal{G}_2}_\Sigma(u, u') \subseteq \bar{\boldsymbol{r}}^{\mathcal{G}_1}_\Sigma(w', w)$;

(3.4) $(u' \mapsto w', i + 1)$, if $\delta = w \in \Delta^{\mathcal{G}_1}$, $w' \rightsquigarrow_1 w$ and $\boldsymbol{r}^{\mathcal{G}_2}_\Sigma(u, u') \subseteq \bar{\boldsymbol{r}}^{\mathcal{G}_1}_\Sigma(w', w)$.

(4) for any nodes $(u \mapsto w, i)$ and $(u' \mapsto w', i)$ in $\mathfrak{T}$ with $w, w' \in \Delta^{\mathcal{G}_1}$ and $i \neq *$, we have $w = w'$.

Note that the conditions on $\mathfrak{T}$ combine the conditions given in the proofs of Lemma 21 (backward strategies, cf. (3.4) and (4)) and Lemma 25 (start-bounded strategies, cf. (3.1)–(3.3)). The graph $\mathfrak{T}$ for the ω-winning strategy in Example 26 is depicted in Fig. 10.

We show first that such a graph $\mathfrak{T}$ exists. Let $\mathbb{S}_0$ be the given set of n-winning strategies of player 1 in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ starting from $(u_0 \mapsto \sigma^n_0)$. Define $\mathfrak{T}_0$ to be the graph with the single initial node $(u_0 \mapsto w_0, 0)$. In the sequel, we slightly abuse notation and use ε for the empty word so that εa is regarded to be the same as a, an element of $\mathsf{ind}(\mathcal{K}_1)$. We say that a strategy $\mathcal{S} \in \mathbb{S}_0$ *respects* $\mathfrak{T}$ if there exists a sequence $\sigma^{\mathcal{S}}_0, \sigma^{\mathcal{S}}_1, \dots$ of elements of $\Delta^{\mathcal{M}_1} \cup \{\varepsilon\}$ such that

– each $\sigma^{\mathcal{S}}_i$ satisfies $\sigma^{\mathcal{S}}_{i-1} = \sigma^{\mathcal{S}}_i w$, for some $w \in \Delta^{\mathcal{G}_1}$, with $\sigma^{\mathcal{S}}_{-1} = \sigma^n_0$, and

– if $(u' \mapsto \delta', i')$ is a $(u \rightsquigarrow^\Sigma_2 u')$-successor of $(u \mapsto \delta, i)$ in $\mathfrak{T}$ then, according to $\mathcal{S}$, player 1 responds to the challenge $u \rightsquigarrow^\Sigma_2 u'$ of player 2 in the state $(u \mapsto \sigma^{\mathcal{S}}_i \delta)$ with $(u' \mapsto \sigma^{\mathcal{S}}_{i'} \delta')$,

where $\sigma^{\mathcal{S}}_* = \varepsilon$. (Intuitively, $\sigma^{\mathcal{S}}_i$ is the $\sigma^{\mathcal{S}}_{i-1}$ without the last element, and so the sequence $\sigma^{\mathcal{S}}_0 w_0, \sigma^{\mathcal{S}}_1 w_1, \sigma^{\mathcal{S}}_2 w_2, \dots$, with $w_i = \mathsf{tail}(\sigma^{\mathcal{S}}_{i-1})$, are the responses to the challenges of the strategy.) Clearly, all strategies in $\mathbb{S}_0$ respect $\mathfrak{T}_0$. Suppose we have already constructed $\mathfrak{T}_k$ and $\mathbb{S}_k$ such that every $\mathcal{S} \in \mathbb{S}_k$ respects $\mathfrak{T}_k$. If $\mathfrak{T}_k$ satisfies (3), then we are done. Otherwise, $\mathfrak{T}_k$ contains a node $(u \mapsto \delta, i)$ without a $(u \rightsquigarrow^\Sigma_2 u')$-successor, for some $u \rightsquigarrow^\Sigma_2 u'$. (We take such a node to be closest to the initial node.) Using the pigeonhole principle, we can find δ', i' and a subset $\mathbb{S}_{k+1} \subseteq \mathbb{S}_k$ such that one of the following four options holds for *all* strategies $\mathcal{S} \in \mathbb{S}_{k+1}$ simultaneously: the response of player 1 according to $\mathcal{S}$ to the challenge $(u \rightsquigarrow_2^\Sigma u')$ in state $(u \mapsto \sigma_i^{\mathcal{S}}\delta)$ is of the form

$$(u' \mapsto \sigma_i^{\mathcal{S}}\delta') \quad \text{with} \quad \delta' = \delta w' \quad \text{and} \quad i' = i, \tag{r.1}$$

$$(u' \mapsto \delta') \;\text{ with }\; \sigma_i^{\mathcal{S}} = \varepsilon,\; \delta, \delta' \in \mathsf{ind}(\mathcal{K}_1) \;\text{ and }\; i' = *, \tag{r.2}$$

$$(u' \mapsto \sigma_i^{\mathcal{S}}\delta') \;\text{ with }\; \delta = \delta' w \;\text{ and }\; i' = i, \tag{r.3}$$

$$(u' \mapsto \sigma_i^{\mathcal{S}}) \;\text{ with }\; \delta \in \Delta^{\mathcal{G}_1},\; \delta' = \mathsf{tail}(\sigma_i^{\mathcal{S}}) \;\text{ and }\; i' = i+1; \tag{r.4}$$

see also items (3.1)–(3.4) above. In each of the four cases, we define $\mathfrak{T}_{k+1}$ by extending $\mathfrak{T}_k$ with $(u' \mapsto \delta', i')$ as a $(u \rightsquigarrow_2^\Sigma u')$-successor of $(u \mapsto \delta, i)$. Observe also that all $\mathcal{S} \in \mathbb{S}_{k+1}$ clearly respect $\mathfrak{T}_{k+1}$. We proceed in the same way and construct sequences $\mathfrak{T}_0 \subseteq \mathfrak{T}_1 \subseteq \dots$ and $\mathbb{S}_0 \supseteq \mathbb{S}_1 \supseteq \mathbb{S}_2, \dots$ until we either reach some $\mathfrak{T}_n$ satisfying (1)–(4) or obtain infinite sequences and take $\mathfrak{T} = \bigcup_{n<\omega} \mathfrak{T}_n$, which obviously satisfies (1)–(4).

Now we show that $\mathfrak{T}$ defines an ω-winning strategy for player 1 in $G_\Sigma^g(\mathcal{G}_2, \mathcal{G}_1)$ starting from some $(\Xi_0 \mapsto w_0, \Psi_0)$. Let $w_0, w_1, \dots$ be the longest (and possibly infinite) sequence of elements of $\Delta^{\mathcal{G}_1}$ such that, for each w_i, there exists u with $(u \mapsto w_i, i)$ a node in $\mathfrak{T}$. Note that, by (4), every w_i (if it exists) is uniquely determined. For each $i \geq 0$ with w_i defined, set

$$\Xi_i = \{\, u \mid (u \mapsto w_i, i) \text{ in } \mathfrak{T} \,\} \qquad \text{and} \qquad \Psi_i = \{\, u' \mid u \rightsquigarrow_2^\Sigma u',\ (u \mapsto w_i, i) \text{ and } (u' \mapsto w_{i+1}, i+1) \text{ are in } \mathfrak{T} \,\}$$

and observe that $u_0 \in \Xi_0$, $\Xi_i \neq \emptyset$ and $\Psi_i \subseteq \Xi_i^{\rightsquigarrow}$, $\Psi_i \subseteq \Xi_{i+1}$, for all $i \geq 0$ such that the sets are defined. Note also that if the sequence $w_0, w_1, \dots$ is finite then the last Ψ_k is empty. Similarly to the proof of Lemma 21, take the maximal $m < \omega$ such that w_m exists and $w_i \neq w_m$ for all $i < m$.

To show that each $(\Xi_i \mapsto w_i, \Psi_i)$, for $0 \leq i \leq m$, is a valid state in the game $G_\Sigma^g(\mathcal{G}_2, \mathcal{G}_1)$, we have to define an ω-winning strategy for the start-bounded game $G_\Sigma^s(\mathcal{G}_2, \mathcal{G}_1)$ from $(\emptyset, \Xi_i \mapsto w_i)$ with the first-round challenges $u \rightsquigarrow_2^\Sigma v$ such that $v \notin \Psi_i$. Fix i and define a graph $\mathfrak{T}_i$ containing the nodes $(u \mapsto \delta)$, for $(u \mapsto \delta, i)$ in $\mathfrak{T}$, and all the nodes $(u \mapsto \delta)$ such that $(u \mapsto \delta, *)$ is reachable from some $(u' \mapsto \delta', i)$ in $\mathfrak{T}$ by a path *not* containing any $(u'' \mapsto \delta'', i+1)$. The arrows and their labels in $\mathfrak{T}_i$ are induced in the obvious way by the arrows of $\mathfrak{T}$. Observe that $\mathfrak{T}_i$ satisfies (1) and (2) of Lemma 25 and satisfies (3) except, perhaps, in nodes $(u \mapsto w_i)$ with $u \rightsquigarrow_2^\Sigma v$ and $v \in \Psi_i$. It can now be shown in the same way as in Lemma 25 that player 1 has an ω-winning strategy in the start-bounded game $G_\Sigma^s(\mathcal{G}_2, \mathcal{G}_1)$ from $(\emptyset, \Xi_i \mapsto w_i)$ provided that the challenge $u \rightsquigarrow_2^\Sigma v$ in the first round satisfies $v \notin \Psi_i$.

Now, by (3.4), the states $(\Xi_i \mapsto w_i, \Psi_i)$, $i \leq m$, clearly define an ω-winning strategy for player 1 in the game $G_\Sigma^g(\mathcal{G}_2, \mathcal{G}_1)$ starting from $(\Xi_0 \mapsto w_0, \Psi_0)$: if player 2 challenges (with Ψ_i) in some state $(\Xi_i \mapsto w_i, \Psi_i)$, then player 1 responds with $(\Xi_{i+1} \mapsto w_{i+1}, \Psi_{i+1})$ if $i < m$, and by the uniquely determined $(\Xi_k \mapsto w_k, \Psi_k)$ such that $w_k = w_{m+1}$ if $i = m$.

$(b) \Rightarrow (a)$ Suppose player 1 has an ω-winning strategy $\mathcal{S}$ starting from $\mathfrak{a}_0 = (\Xi_0 \mapsto w_0, \Psi_0)$ in $G_\Sigma^g(\mathcal{G}_2, \mathcal{G}_1)$ with $u_0 \in \Xi_0$ and let $n < \omega$. Consider any play in $G_\Sigma^g(\mathcal{G}_2, \mathcal{G}_1)$ starting from $\mathfrak{a}_0$ and conforming with $\mathcal{S}$. One can represent the play as a sequence

$$(\mathfrak{a}_0,\ u_0^0 \rightsquigarrow_2^\Sigma v_0^0, \dots,\ u_{k_0}^0 \rightsquigarrow_2^\Sigma v_{k_0}^0), \quad (\mathfrak{a}_1,\ u_0^1 \rightsquigarrow_2^\Sigma v_0^1, \dots,\ u_{k_1}^1 \rightsquigarrow_2^\Sigma v_{k_1}^1), \quad \dots \quad ,$$

where each $\mathfrak{a}_i$ is a response of player 1 (a state of the game $G_\Sigma^g(\mathcal{G}_2, \mathcal{G}_1)$) to the (uniquely determined) challenge in $\mathfrak{a}_{i-1}$, and $u_0^i \rightsquigarrow_2^\Sigma v_0^i, \dots, u_{k_i}^i \rightsquigarrow_2^\Sigma v_{k_i}^i$ are the challenges of player 2 in the start-bounded game $G_\Sigma^s(\mathcal{G}_2, \mathcal{G}_1)$ from $\mathfrak{a}_i$ (in which case player 1 has an ω-winning strategy). Similarly to the backward game, the sequence $\mathfrak{a}_0, \mathfrak{a}_1, \dots$ does not depend on the challenges of player 2 but only on $\mathfrak{a}_0$ and $\mathcal{S}$. So we fix the sequence $\mathfrak{a}_0, \mathfrak{a}_1, \dots, \mathfrak{a}_k$, where either $k = n$ or $k < n$ is the maximal number of states reached in any play starting from $\mathfrak{a}_0$ according to $\mathcal{S}$. This sequence induces a sequence $w_0, w_1, \dots, w_k$ of elements of $\Delta^{\mathcal{G}_1}$ given by the states $\mathfrak{a}_i = (\Xi_i \mapsto w_i, \Psi_i)$. We take any element $\sigma \in \Delta^{\mathcal{M}_1}$ with $\mathsf{tail}(\sigma) = w_k$ and let $\sigma_0^n = \sigma w_{k-1} \dots w_0$. In addition to the ω-winning strategy $\mathcal{S}$, we also fix the ω-winning strategies for player 1 in the start-bounded games for $G_\Sigma^g(\mathcal{G}_2, \mathcal{G}_1)$ from $\mathfrak{a}_i$ with the appropriate challenge in the first round.

Now, for any sequence $u_0 \rightsquigarrow_2^\Sigma u_1 \rightsquigarrow_2^\Sigma \cdots \rightsquigarrow_2^\Sigma u_{m-1} \rightsquigarrow_2^\Sigma u_m$, $m \leq n$, of challenges by player 2 in the game $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ starting from $\mathfrak{s}_0 = (u_0 \mapsto \sigma_0)$ with $\sigma_0 = \sigma_0^n$, we construct a sequence of responses $\mathfrak{s}_1 = (u_1 \mapsto \sigma_1)$, $\dots$, $\mathfrak{s}_m = (u_m \mapsto \sigma_m)$ of player 1. In order to do this, we define inductively a sequence $\pi_0, \dots, \pi_m$ (of non-empty sequences) such that the following hold for each $i \leq m$:

– π_i begins with one of the states $\mathfrak{a}_0, \dots, \mathfrak{a}_k$, and all other elements in π_i are states $(\Theta, \Xi \mapsto w)$ of the respective start-bounded game;

– if $\pi_i = \pi_j \cdot (\Theta^1, \Xi^1 \mapsto w^1) \cdots (\Theta^\ell, \Xi^\ell \mapsto w^\ell)$ then $\sigma_i = \sigma_j w^1 \cdots w^\ell$.

For $i = 0$, we set $\pi_0 = \mathfrak{a}_0 = (\Xi_0 \mapsto w_0, \Psi_0)$, which clearly has the required properties. Now suppose that $\mathfrak{s}_0, \dots, \mathfrak{s}_{i-1}$ and $\pi_0, \dots, \pi_{i-1}$ have already been defined, for $1 \leq i \leq m$. Consider a challenge $u_{i-1} \rightsquigarrow_2^\Sigma u_i$ in the state $\mathfrak{s}_{i-1}$. Two cases are possible.

– If π_{i-1} consists of a single state $(\Xi \mapsto w, \Psi)$ then it coincides with some $\mathfrak{a}_{j-1}$, for $j \leq k$. Recall that $u_{i-1} \in \Xi$ and $\mathsf{tail}(\sigma_{i-1}) = w$. We have the following two options.

  – If $u_i \in \Psi$ then we set $\pi_i = \mathfrak{a}_j$ and obtain σ_i from σ_{i-1} by removing its final element, w.

  – Otherwise, $u_i \in \Xi^{\rightsquigarrow} \setminus \Psi$ and we launch the start-bounded game $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ from $(\emptyset, \Xi \mapsto w)$ and set $\pi_i = \pi_{i-1} \cdot (\Theta', \Xi' \mapsto w')$ and $\sigma_i = \sigma_{i-1} w'$, where $(\Theta', \Xi' \mapsto w')$ is the response of player 1 to $u_{i-1} \rightsquigarrow_2^\Sigma u_i$ according to the ω-winning strategy in the start-bounded game.

– Otherwise, the final element of π_{i-1} is a state of the start-bounded game, and we follow the construction from the proof of $(c) \Rightarrow (a)$ in Lemma 25.

This completes the proof of the lemma. ❑

Similarly to the start-bounded game, the size of the game graph for $G^g_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ is exponential in the size of $\mathcal{G}_2$ as it contains $O((|\mathsf{ind}(\mathcal{K}_2)| \times 2^{|\Delta^{\mathcal{G}_2} \setminus \mathsf{ind}(\mathcal{K}_2)|})^2 \times |\Delta^{\mathcal{G}_1}|)$ states. Note, however, that when constructing the graph, we have to check that for each of its states player 1 has an ω-winning strategy in the corresponding start-bounded game. As observed in Section 4.4, this can also be done in time exponential in $\Delta^{\mathcal{G}_2} \setminus \mathsf{ind}(\mathcal{K}_2)$ and polynomial in both $\mathsf{ind}(\mathcal{K}_2)$ and $\Delta^{\mathcal{G}_1}$. In view of Theorem 11 (i) and (iv), we then obtain:

Theorem 29. *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.

For *DL-Lite$_{core}$* and *DL-Lite$_{horn}$* KBs, the general game $G^g_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ can be significantly simplified. Note first that the start-bounded game $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ in this case can be reduced to the forward game $G^f_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$. Indeed, by **(lite$_2$)** and the fact that $(u, v)^{\mathcal{G}}$ is always a singleton set in the generating structures for *DL-Lite$_{horn}$*, player 2 cannot challenge player 1 in any round $i > 0$ of $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ with $u \rightsquigarrow_2^\Sigma v$ such that $\boldsymbol{r}^{\mathcal{G}_2}_\Sigma(u, v) \subseteq \bar{\boldsymbol{r}}^{\mathcal{G}_1}_\Sigma(w_{i-2}, w_{i-1})$. Thus, **(no-backward)** holds for any set Θ_i, and so we obtain: for any $u_0 \in \Delta^{\mathcal{G}_2}$ and $w_0 \in \Delta^{\mathcal{G}_1}$, player 1 has an ω-winning strategy in $G^s_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ with an initial state $(\emptyset, \Xi_0 \mapsto w_0)$ and $u_0 \in \Xi_0$ if and only if player 1 has an ω-winning strategy in $G^f_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ with the initial state $(u_0 \mapsto w_0)$.

Second, since having a start-bounded ω-winning strategy with an initial state $(\emptyset, \Xi \mapsto w)$ is equivalent to having forward ω-winning strategies for all initial states $(u \mapsto w)$ with $u \in \Xi$, for any general ω-winning strategy player 1 can choose Ξ_i as small as possible: $\Xi_i = \{u_0\}$ in the initial state and $\Xi_i = \Psi_{i-1}$, for $i > 0$. Also observe that in the general game, if Ξ_{i-1} contains at most one element, then player 1 has to choose for Ψ_i a set containing at most one element (if player 1 chooses a set with at least two elements, then he will not have a response to the challenge Ψ_i since the generating structures for *DL-Lite$_{horn}$* KBs are functional). It follows by induction that if player 1 has an ω-winning strategy in the general game then player 1 has an ω-winning strategy in which all states are of the form $(\Xi_i \mapsto w_i, \Psi_i)$, where Ξ_i is a singleton set, Ψ_i has at most one element, and $\Xi_i = \Psi_{i-1}$. The number of states in this game is polynomial, and so the existence of an ω-winning strategy can be checked in P. Note also that this strategy corresponds to the winning strategy in the naïve game $G^n_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$ sketched in Section 4.1.

Theorem 30. *Σ-query entailment for DL-Lite$_{core}$ and DL-Lite$_{horn}$ KBs is in* P *for both combined and data complexity.*

5. Lower Bounds

In this section, we show that the upper complexity bounds obtained in Section 4 are optimal. Throughout the section we assume that the materialisations of the KBs we deal with are the unravellings of the generating structures for those KBs constructed as described in Section 3.

As we have seen in the previous section, the problems of Σ-query entailment and inseparability for all of our DLs are in P for data complexity. The next theorem establishes a matching lower bound:

Theorem 31. *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 p_i and $p_i \wedge p_{i'} \to p_j$, with $i, i' < j$, decide whether p_n is true in every model of φ. Consider a *DL-Lite$_{core}$* TBox $\mathcal{T}$ containing the following concept inclusions:

$$V \sqsubseteq \exists S, \quad \exists S^- \sqsubseteq \exists R_k \text{ and } \exists R_k^- \sqsubseteq V, \text{ for } k = 1, 2,$$

and let an ABox $\mathcal{A}$ consist of $F(p_n)$ and

$$S(p_i, p_i), \; R_1(p_i, p_i), \; R_2(p_i, p_i), \qquad \text{for each clause } p_i \text{ in } \varphi,$$

$$S(p_j, c), \; R_1(c, p_i), \; R_2(c, p_{i'}), \qquad \text{for each clause } c = p_i \wedge p_{i'} \to p_j \text{ in } \varphi.$$

Set $\Sigma = \{F, S, R_1, R_2\}$, $\mathcal{K}_1 = (\emptyset, \mathcal{A})$ and $\mathcal{K}_2 = (\mathcal{T}, \mathcal{A} \cup \{V(p_n)\})$. 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 p_n. Indeed, the materialisation $\mathcal{M}_2$ of $\mathcal{K}_2$ is infinite, while the materialisation $\mathcal{M}_1$ of $\mathcal{K}_1$ is finite. So, the only way to embed finite prefixes of $\mathcal{M}_2$ of arbitrary depth into $\mathcal{M}_1$ is by mapping subtrees of unbounded depth into the loops in $\mathcal{M}_1$ for unary clauses p_i in φ, which is only possible if there is a tree of clauses of the form $p_i \wedge p_{i'} \to p_j$ with root p_n and leaves among the clauses p_i of φ (that is, if there is a derivation of p_n from φ).

For $\mathcal{EL}$, we take $\mathcal{T} = \{ V \sqsubseteq \exists S.(\exists R_1.V \sqcap \exists R_2.V) \}$. The remainder of the proof is the same as above. ❑

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

Theorem 32. *For combined complexity, the problems of Σ-query entailment and inseparability are* ExpTime*-hard for DL-Lite$^{\mathcal{H}}_{core}$ KBs.*

Proof. The proof is by encoding alternating Turing machines (ATMs) with polynomial tape and using the fact that APSpace = ExpTime; see, e.g. [34].

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

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

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 leads to a universal state, and vice versa. We extend δ with the instructions $\delta(q_1, a, j) = (q_1, a, 0)$, for $a \in \Lambda$ and $j = 1, 2$, which go into an infinite loop if M reaches the accepting state q_1. Thus, assuming that M terminates on every input, it accepts an input $\boldsymbol{w}$ if and only if the modified ATM M' has a run on $\boldsymbol{w}$ all branches of which are infinite.

Given M' and an input $\boldsymbol{w}$, our aim is to construct TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ and a signature Σ such that M' has a run with only infinite branches if and only if the materialisation $\mathcal{M}_2$ of $(\mathcal{T}_2, \mathcal{A})$ is finitely Σ-homomorphically embeddable into the materialisation $\mathcal{M}_1$ of $(\mathcal{T}_1, \mathcal{A})$, where $\mathcal{A}$ is an ABox with a single assertion $A(c)$. Let f be a polynomial such that, on any input of length m, M' uses at most $n = f(m)$ cells, which are numbered from 1 to n, and throughout any computation the head remains to the right of cell 0, which contains a special marker $\flat \in \Lambda$.

The construction proceeds in four steps. In the definition of the TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$, we use concept inclusions of the form $B \sqsubseteq \exists R.(C_1 \sqcap \cdots \sqcap C_k)$ as an abbreviation for

$$B \sqsubseteq \exists R_0, \quad R_0 \sqsubseteq R \text{ and } \exists R_0^- \sqsubseteq C_i, \text{ for } 1 \leq i \leq k,$$

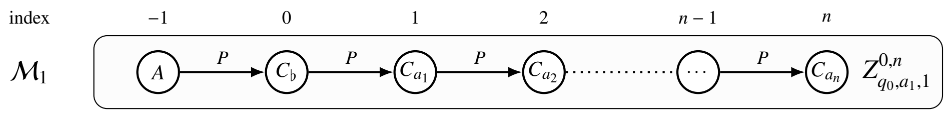


Figure 11: Encoding the initial configuration by a block.

where R_0 is a fresh role name. If C_i is a complex concept then $\exists R_0^- \sqsubseteq C_i$ is also treated as an abbreviation for the respective concept and role inclusions.

Step 1. First we encode configurations and transitions of M' using $\mathcal{T}_1$. We represent a configuration (that is, the content of every cell on the tape, the state and the position of the head) by a sequence of $n+2$ domain elements in $\mathcal{M}_1$, which will be called a *block*. The first element in each block is used to distinguish the type of the block, whereas the remaining elements are assigned indexes from 0 to n: if the element with index i belongs to C_a, for some $a \in \Lambda$, then the ith cell of the tape is assumed to contain a in the configuration defined by the block as shown in Fig. 11 (the first element of the block has index -1). The first block represents the initial configuration, that is, symbols $a_1, \dots, a_n$ written in the n cells of the tape (the input $\boldsymbol{w}$ padded with $\llcorner\!\lrcorner$) and the initial state q_0, which is achieved by the following inclusion in $\mathcal{T}_1$:

$$A \sqsubseteq \exists P.(C_\flat \sqcap \exists P.(C_{a_1} \sqcap \exists P.(C_{a_2} \sqcap \exists P.(\dots \exists P.(C_{a_n} \sqcap Z^{0,n}_{q_0,a_1,1})\dots)))). \qquad (\mathcal{T}_1\text{-}1)$$

Step 2. The current state $q \in Q$, the position k of the head and the content $a \in \Lambda$ of the active cell scanned by the head are recorded in the concept $Z^{0,n}_{q,a,k}$ that contains the last element of the block. At the end of the block we branch out one block for each of the two instructions and propagate via the $Z^{1,i}_{q,a,k}$ and the $Z^{2,i}_{q,a,k}$ the current state, head position and symbol in the active cell: for $q \in Q$, $a \in \Lambda$ and $1 \le k \le n$, we add to $\mathcal{T}_1$ the inclusions

$$Z^{0,n}_{q,a,k} \sqsubseteq \bigsqcap_{j=1,2} \exists P.(X_j \sqcap Z^{j,-1}_{q,a,k}), \qquad (\mathcal{T}_1\text{-}2)$$

where X_1 and X_2 are two fresh concept names (which specify the type of the block).

The acceptance condition for M' is enforced by means of $\mathcal{T}_2$. For the initial block representing the initial configuration we take

$$A \sqsubseteq \exists P.\underbrace{\exists P.\cdots\exists P.}_{n \text{ times}} \bigsqcap_{j=1,2} \exists P.X_j. \qquad (\mathcal{T}_2\text{-}1)$$

The two concept names, X_1 and X_2, are used to distinguish between the two blocks for universal successor states and one more concept name, X_3, marks both blocks for existential state successors. These blocks are arranged into an infinite tree-like structure: the initial block is the root from which an X_1- and an X_2-blocks branch out (recall that successors of the initial state q_0 are universal). Each of them is followed by an X_3-block, which branches out an X_1- and an X_2-block, and so on. This is achieved by adding to $\mathcal{T}_2$ the following inclusions:

$$X_3 \sqsubseteq \exists P.\underbrace{\exists P.(G \sqcap \exists P.(\cdots \exists P.(G \sqcap}_{n \text{ times}} \bigsqcap_{j=1,2} \exists P.X_j))), \qquad (\mathcal{T}_2\text{-}2)$$

$$X_j \sqsubseteq \exists P.\underbrace{\exists P.(G \sqcap \exists P.(\cdots \exists P.(G \sqcap}_{n \text{ times}} \exists P.X_3))), \qquad \text{for } j = 1,2, \qquad (\mathcal{T}_2\text{-}3)$$

where G is a fresh concept name (which marks every cell of the tape). If $\Sigma = \{A, X_1, X_2, P\}$ then there is a unique Σ-homomorphism from the initial block in $\mathcal{M}_2$ to the block of the initial configuration in $\mathcal{M}_1$. Next, signature concepts X_1 and X_2 ensure that the X_1- and X_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 X_3-block, however, contains no signature marker (X_1 or X_2) and can be mapped to either of the blocks in $\mathcal{M}_1$, which reflects the choice in existential states; see Fig. 12, where possible Σ-homomorphisms are shown by thick dashed arrows.

Step 3. Recall that the $Z^{j,i}_{q,a,k}$, for $-1 \le i \le n$, specify the position k of the head on the tape. Let the active cell in the previous configuration be k. Then, until the cell $k-2$ is reached in the current configuration, the following inclusions

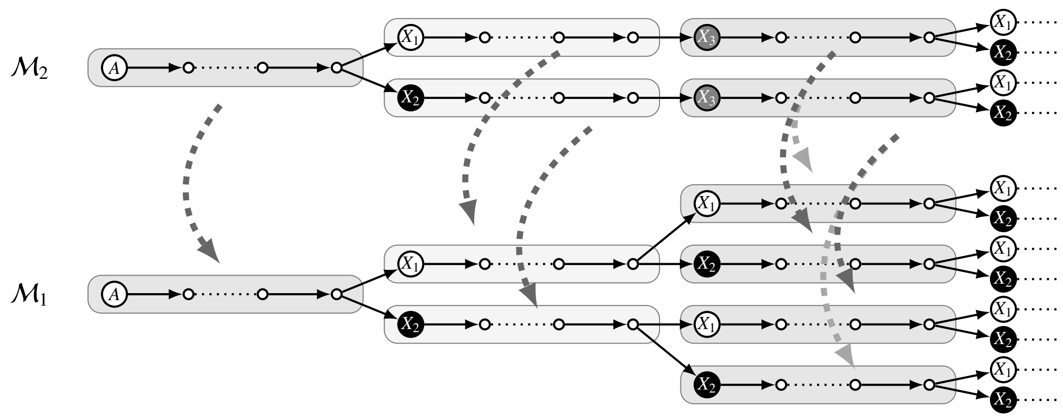


Figure 12: The structure of Σ-homomorphisms from $\mathcal{M}_2$ to $\mathcal{M}_1$: note that $A, X_1, X_2 \in \Sigma$ but $X_3 \notin \Sigma$.

in $\mathcal{T}_1$ propagate its current state ($q \in Q$), the symbol in the active cell ($a \in \Lambda$), the head position ($1 \leq k \leq n$) and the block type ($j = 1, 2$) along the domain elements constituting the block: for $-1 < i \leq n$ with $i \neq k-1$,

$$Z^{j,i-1}_{q,a,k} \sqsubseteq \bigsqcap_{b \in \Lambda} \exists P.(C_b \sqcap Z^{j,i}_{q,a,k}) \qquad (\mathcal{T}_1\text{-}3)$$

(for each $b \in \Lambda$, these concept inclusions also 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). 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 $Z^{j,i}_{q,a,k}$ to specify the head position, k, and the cell number, i. When the cell $k-2$ is reached, the contents of the active cell, the information from the subscripts of the $Z^{j,i}_{q,a,k}$ is used to perform the instruction according to δ:

$$Z^{j,k-2}_{q,a,k} \sqsubseteq \begin{cases} \bigsqcap_{b \in \Lambda} \exists P.(C_b \sqcap \exists P.(F_{a'} \sqcap Z^{0,k}_{q',b,k-1})), & \text{if } \delta(q,a,j) = (q',a',-1), \\ \bigsqcap_{b \in \Lambda} \exists P.(C_b \sqcap \exists P.(F_{a'} \sqcap Z^{0,k}_{q',a',k})), & \text{if } \delta(q,a,j) = (q',a',0), \\ \bigsqcap_{b \in \Lambda} \exists P.(C_b \sqcap \exists P.(F_{a'} \sqcap \bigsqcap_{b' \in \Lambda} \exists P.(C_{b'} \sqcap Z^{0,k+1}_{q',b',k+1}))), & \text{if } \delta(q,a,j) = (q',a',+1). \end{cases} \qquad (\mathcal{T}_1\text{-}4)$$

Specifically, the symbol in the active cell, k, is changed according to the instruction and the cell is marked by concept $F_{a'}$. Then the current state, symbol in the active cell of the successive configuration and the new head position are recorded in the subscripts of the concepts $Z^{0,i}_{q,a,k}$; note that the block type marker, $j = 1, 2$, is replaced by 0. These three situations are depicted in Fig. 13, where the hatched nodes denote domain elements *two* cells before the active cell of the configuration (where inclusion ($\mathcal{T}_1$-4) becomes 'active) and the filled black and grey nodes denote domain elements for the active cell. (Note that the element corresponding to the cell $k-1$ has only one P-successor, which encodes the new symbol, a', in that cell; see explanations below.) Then the new state and the symbol in the active cell of the successive configurations are propagated further along the tape using ($\mathcal{T}_1$-3) with $j = 0$ and $i > k-1$.

Step 4. The inclusions ($\mathcal{T}_1$-3)–($\mathcal{T}_1$-4) generate a separate P-successor for each $b \in \Lambda$, thus not preserving the contents of the tape between transitions. We now add a number of inclusions to both TBoxes so that wrong branches would be ignored by any finite Σ-homomorphism, h, from $\mathcal{M}_2$ to $\mathcal{M}_1$, where

$$\Sigma = \{A, P, X_1, X_2\} \cup \{D_a \mid a \in \Lambda\}. \qquad (5)$$

Suppose $h(d_2) = d_1$ and d_2 belongs to G in $\mathcal{M}_2$ (and therefore, it represents a cell in a non-initial configuration). We

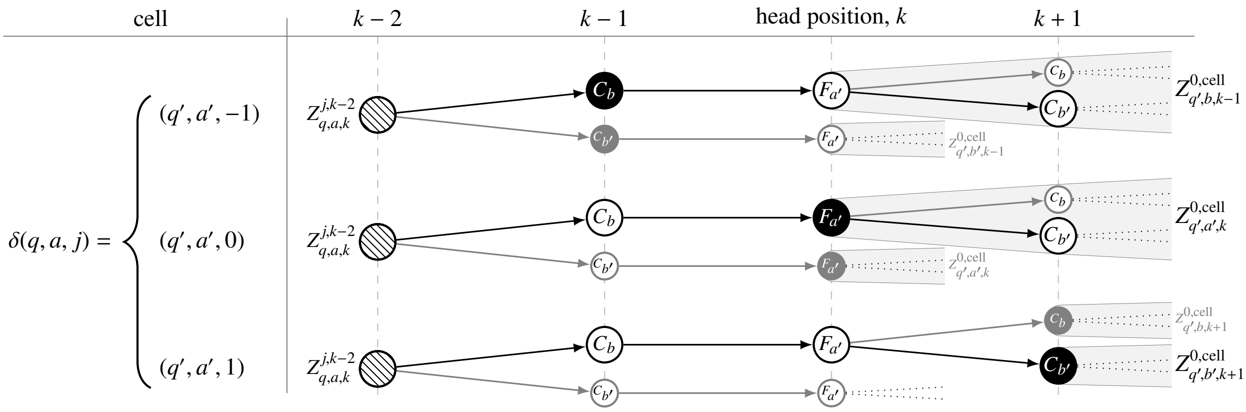


Figure 13: Executing the instructions of M'.

add the following two inclusions to $\mathcal{T}_2$:

$$G \sqsubseteq \bigsqcap_{b\in\Lambda} G_b, \tag{$\mathcal{T}_2$-4}$$

$$G_b \sqsubseteq \exists P^-.\underbrace{\exists P^-.\cdots\exists P^-}_{n \text{ times}}.\exists P^-.D_b, \qquad \text{for } b \in \Lambda. \tag{$\mathcal{T}$-1}$$

Then, for each symbol $b \in \Lambda$, the element d_2 generates a block of $n+2$-many P^--connected elements that ends in the concept D_b; we call it a *D_b-block of d_2*. Recall from Step 3 that, for $a \in \Lambda$, if $d_1 \in F_a^{\mathcal{M}_1}$ then it represents a cell whose content is changed to a (in which case d_1 has no 'siblings', that is, the P-predecessor of d_1 has a single P-successor, d_1). However, if $d_1 \in C_a^{\mathcal{M}_1}$ then the content of the cell represented by d_1 must be copied from the previous configuration). This is achieved by adding ($\mathcal{T}$-1) and the following inclusions to $\mathcal{T}_1$:

$$F_a \sqsubseteq D_a \sqcap \bigsqcap_{b\in\Lambda} G_b, \tag{$\mathcal{T}_1$-5}$$

$$C_a \sqsubseteq D_a \sqcap \bigsqcap_{b\in\Lambda\setminus\{a\}} G_b. \tag{$\mathcal{T}_1$-6}$$

So, if $d_1 \in F_a^{\mathcal{M}_1}$ then d_1 has a D_b-block for any $b \in \Lambda$ and, by the choice of Σ, each of the D_b-blocks of d_2 in $\mathcal{M}_2$ can be mapped by h to the respective D_b-block of d_1 in $\mathcal{M}_1$. On the other hand, if $d_1 \in C_a^{\mathcal{M}_1}$ then d_1 has a D_b-block only for $b \in \Lambda$ with $b \neq a$. So, all D_b-blocks of d_2 with $b \neq a$ can still be mapped by h to the respective D_b-blocks of d_1 in $\mathcal{M}_1$. The remaining D_a-block of d_2 could be mapped in the *reverse order* along the 'main' branch in $\mathcal{M}_1$ *but only* if the cell contains a in the preceding configuration (that is, the element that is $n+2$ steps closer to the root of $\mathcal{M}_1$ belongs to D_a); see Fig. 14.

One can show now that $\mathcal{T}_1$ and $\mathcal{T}_2$ are as required: M' has a run with only infinite branches if and only if the materialisation $\mathcal{M}_2$ of $(\mathcal{T}_2, \mathcal{A})$ is finitely Σ-homomorphically embeddable into the materialisation $\mathcal{M}_1$ of $(\mathcal{T}_1, \mathcal{A})$. It remains to use Theorem 5 and the fact that APSpace = ExpTime. It follows, by Theorem 12, that deciding Σ-query inseparability is also ExpTime-hard. ❑

Theorem 33. *For combined complexity, the problems of Σ-query entailment and inseparability are* 2ExpTime*-hard for* Horn-$\mathcal{ALCI}$ *KBs.*

Proof. The proof is by encoding alternating Turing machines (ATMs) with exponential tape and using the fact that AExpSpace = 2ExpTime.

As in the proof of Theorem 32, let $M = (\Lambda, Q, q_0, q_1, \delta)$ 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. Given M' and an input

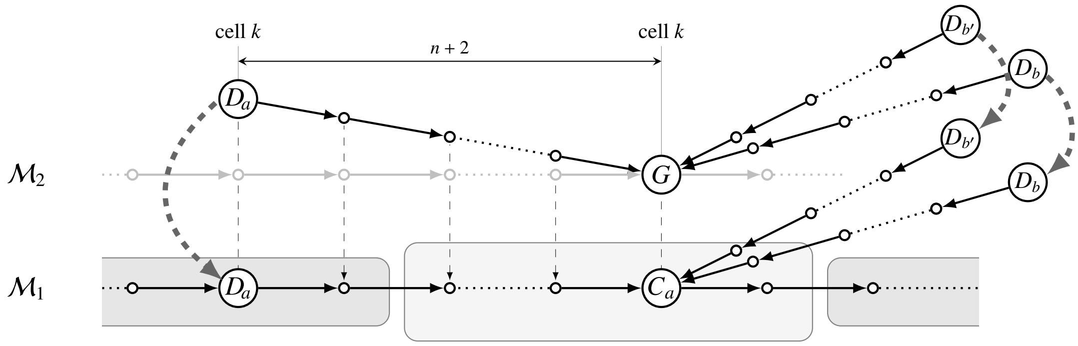


Figure 14: Ensuring succession of M' configurations.

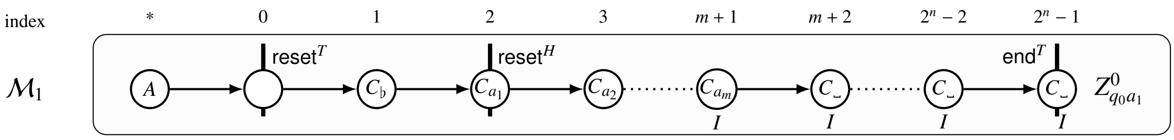


Figure 15: Encoding the initial configuration by a block.

$\boldsymbol{w}$, our aim is to construct two TBoxes, $\mathcal{T}_1'$ and $\mathcal{T}_2'$, and a signature Σ such that M' has a run with only infinite branches if and only if the materialisation $\mathcal{M}_2$ of $(\mathcal{T}_2', \mathcal{A})$ is finitely Σ-homomorphically embeddable into the materialisation $\mathcal{M}_1$ of $(\mathcal{T}_1', \mathcal{A})$, where $\mathcal{A} = \{A(c)\}$. 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 throughout any computation the head remains to the right of cell 0, which contains a special marker $\flat \in \Lambda$. The construction proceeds in five steps (steps 1–4 are similar to steps 1–4 in the proof of Theorem 32).

Step 0. We use tuples of $2n$ concept names 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 concept inclusions to encode an n-bit *R-counter on Y*:

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

(Note that we will need P-counters as well as P^--counters.) We use the expression end^Y on the left-hand side of concept inclusions 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 $\mathsf{not\text{-}end}^Y$ on the left-hand side of concept inclusions for the complementary statement (which is a shortcut for n concept inclusions with $\mathsf{not\text{-}end}^Y$ replaced by each of $\overline{Y}_{n-1}, \dots, \overline{Y}_0$). Finally, we use reset^Y on the right-hand side of concept inclusions 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 end^Y 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. As in Theorem 32, the first element distinguishes the blocks for the two alternative instructions; 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 $a \in \Lambda$: 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 in Fig. 15.

The first block represents the initial configuration: the input $\boldsymbol{w} = a_1 \dots a_m$ is followed by $2^n - m - 2$ blank symbols $\llcorner\!\lrcorner$ 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 concept inclusions in the TBox $\mathcal{T}_1'$:

$$A \sqsubseteq \exists P.(\mathsf{reset}^T \sqcap \exists P.(C_\flat \sqcap \exists P.(C_{a_1} \sqcap \mathsf{reset}^H \sqcap \exists P.(C_{a_2} \sqcap \exists P.(\dots \exists P.(C_{a_m} \sqcap I)\dots))))), \qquad (\mathcal{T}_1'\text{-}1)$$

$$\mathsf{not\text{-}end}^T \sqcap I \sqsubseteq \exists P.(I \sqcap C_{\llcorner}), \qquad (\mathcal{T}_1'\text{-}2)$$

$$\mathsf{end}^T \sqcap I \sqsubseteq Z^0_{q_0 a_1}, \qquad (\mathcal{T}_1'\text{-}3)$$

where I is a fresh concept name that is used only for padding of the input with $\llcorner$; cf. ($\mathcal{T}_1$-1).

Step 2. Similarly to the proof of Theorem 32, the current state $q \in Q$ and the content $a \in \Lambda$ of the active cell scanned by the head is recorded in the subscripts of concepts Z^0_{qa} that contain the last element of the block; note, however, that the position of the head must now be specified using the P-counter on H. 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 $a \in \Lambda$, we add to $\mathcal{T}_1'$ the concept inclusion

$$\mathsf{end}^T \sqcap Z^0_{qa} \ \sqsubseteq \ \bigsqcap_{j=1,2} \exists P.(X_j \sqcap \exists P.(\mathsf{reset}^T \sqcap Z^j_{qa})), \qquad (\mathcal{T}_1'\text{-}4)$$

where X_1 and X_2 are two fresh concept names that distinguish the type of the block; cf. ($\mathcal{T}_1$-2).

As in the proof of Theorem 32, the acceptance condition for M' is enforced by means of $\mathcal{T}_2'$, which uses four types of blocks. In this proof, however, we need to use P-counters to reach the end of the block. The P-counter on a tuple T creates the initial block for the initial configuration:

$$A \sqsubseteq \exists P.(\mathsf{reset}^T \sqcap B_0), \qquad (\mathcal{T}_2'\text{-}1)$$

$$\mathsf{not\text{-}end}^T \sqcap B_0 \sqsubseteq \exists P.B_0, \qquad (\mathcal{T}_2'\text{-}2)$$

where B_0 is a fresh concept, an indicator of the initial block. We use X_1- and X_2-blocks for universal states (these blocks are indicated by concepts B_1 and B_2, respectively) and X_3-blocks for existential states (indicated by concept B_3). The tree-like structure of the blocks is achieved by adding to $\mathcal{T}_2'$ the following inclusions:

$$\mathsf{end}^T \sqcap B_k \sqsubseteq \bigsqcap_{j=1,2} \exists P.(X_j \sqcap \exists P.(\mathsf{reset}^T \sqcap B_j)), \quad \text{for } k = 0, 3, \qquad (\mathcal{T}_2'\text{-}3)$$

$$\mathsf{end}^T \sqcap B_j \sqsubseteq \exists P.(X_3 \sqcap \exists P.(\mathsf{reset}^T \sqcap B_3)), \quad \text{for } j = 1, 2, \qquad (\mathcal{T}_2'\text{-}4)$$

$$\mathsf{not\text{-}end}^T \sqcap B_j \sqsubseteq \exists P.(G \sqcap B_j), \quad \text{for } j = 1, 2 \text{ and } 3, \qquad (\mathcal{T}_2'\text{-}5)$$

where G is a fresh concept name; cf. ($\mathcal{T}_2$-2) and ($\mathcal{T}_2$-3); see also Fig. 12. (Note that ($\mathcal{T}_2'$-3) with $k = 0$ is required as a replacement of part of ($\mathcal{T}_2$-1).)

Step 3. Recall that the P-counter on H measures the distance from the head: if the active cell in the current configuration has index k, then its H-value is 0 and the H-value of the cell with index $k - 2$ in a *successor* configuration is $2^n - 1$ (note that since the head never visits cells with indexes 0 and 1, the P-counter on T is ahead of the P-counter on H at least by 2, whence $k - 2 \geq 0$). So, until the H-counter reaches $2^n - 1$, the following concept inclusions in $\mathcal{T}_1'$ propagate the state and symbol in the active cell along the elements constituting the blocks: for $q \in Q$, $a \in \Lambda$ and $j = 0, 1, 2$,

$$\mathsf{not\text{-}end}^T \sqcap \mathsf{not\text{-}end}^H \sqcap Z^j_{qa} \ \sqsubseteq \ \bigsqcap_{b \in \Lambda} \exists P.(C_b \sqcap Z^j_{qa}); \qquad (\mathcal{T}_1'\text{-}5)$$

cf. ($\mathcal{T}_1$-3); note that $\mathsf{not\text{-}end}^T$ means that this concept inclusion is not 'applicable' to the last and the first elements of each block (with indexes $2^n - 1$ and -1, respectively). When the distance from the last head position is $2^n - 2$, the contents of the cell and the current state are changed according to δ: for $q \in Q$, $a \in \Lambda$ and $j = 1, 2$,

$$\mathsf{end}^H \sqcap Z^j_{qa} \ \sqsubseteq \ \begin{cases} \bigsqcap_{b \in \Lambda} \exists P.(C_b \sqcap \mathsf{reset}^H \sqcap Z^0_{q'b} \sqcap \exists P.F_{a'}), & \text{if } \delta(q, a, j) = (q', a', -1), \\ \bigsqcap_{b \in \Lambda} \exists P.(C_b \sqcap \exists P.(F_{a'} \sqcap \mathsf{reset}^H \sqcap Z^0_{q'a'})), & \text{if } \delta(q, a, j) = (q', a', 0), \\ \bigsqcap_{b \in \Lambda} \exists P.\big(C_b \sqcap \exists P.(F_{a'} \sqcap \bigsqcap_{b' \in \Lambda} \exists P.(C_{b'} \sqcap \mathsf{reset}^H \sqcap Z^0_{q'b'}))\big), & \text{if } \delta(q, a, j) = (q', a', +1) \end{cases} \qquad (\mathcal{T}_1'\text{-}6)$$

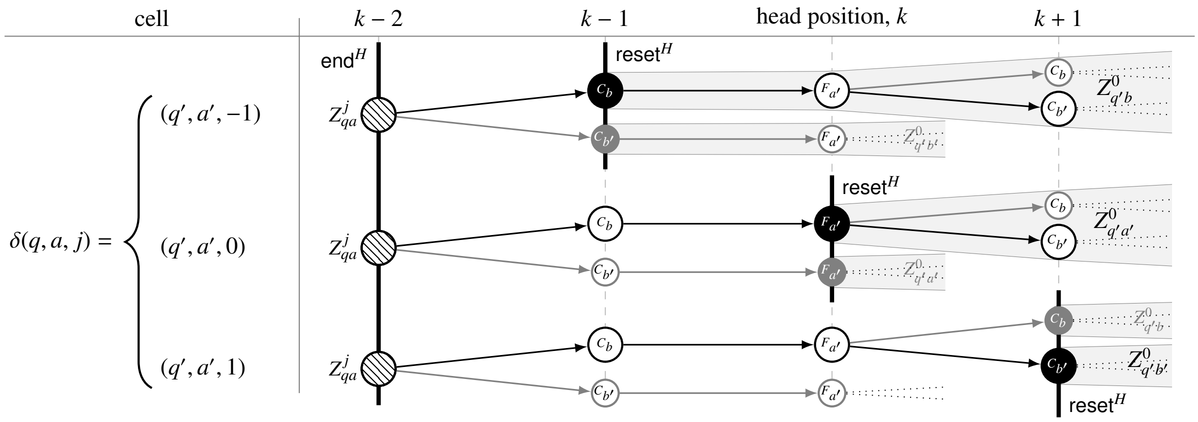


Figure 16: Encoding the transitions of M' in $\mathcal{M}_1$.

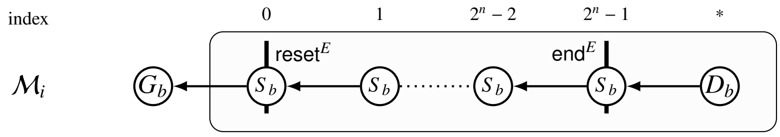


Figure 17: A D_b-block is generated using a P^--counter on a tuple E.

(the symbol in the active cell is changed according to the instruction, and the current state and symbol in the active cell of a successive configuration are then recorded in the subscripts of the Z^0_{qa}). These three situations are depicted in Fig. 16, where hatched nodes denote domain elements with H-values of 2^n-1 and grey and black nodes with H-values of 0. (Again, the element corresponding to the cell $k-1$ has only one P-successor, which encodes the updated symbol, a', in that cell.) Then, the current state and the symbol in the active cell are propagated along the tape using ($\mathcal{T}_1'$-5) with $j=0$.

Step 4. The concept inclusions ($\mathcal{T}_1'$-5)–($\mathcal{T}_1'$-6) generate a separate P-successor for each $b \in \Lambda$. As in the proof of Theorem 32, the correct one is chosen by a finite Σ-homomorphism, h, from $\mathcal{M}_2$ to $\mathcal{M}_1$ for Σ defined by (5). We add ($\mathcal{T}_2$-4) from the proof of Theorem 32 along with the following replacement of ($\mathcal{T}$-1) to $\mathcal{T}_2'$:

$$G_b \sqsubseteq \exists P^-.(S_b \sqcap \mathsf{reset}^E), \qquad (\mathcal{T}'\text{-1})$$

$$\mathsf{not\text{-}end}^E \sqcap S_b \sqsubseteq \exists P^-.S_b, \qquad (\mathcal{T}'\text{-2})$$

$$\mathsf{end}^E \sqcap S_b \sqsubseteq \exists P^-.D_b, \qquad (\mathcal{T}'\text{-3})$$

where we use a P^--counter on a tuple E (unlike P-counters in all other cases) and a concept S_b to propagate b along the whole block, which will be called a D_b-block; see Fig. 17. Like in the proof of Theorem 32, the length of any D_b-block, 2^n+1, matches the length of blocks representing configurations and the last element of a D_b-block belongs to concept D_b. We also add ($\mathcal{T}_1$-5)–($\mathcal{T}_1$-6) from the proof of Theorem 32 and ($\mathcal{T}'$-1)–($\mathcal{T}'$-3) to $\mathcal{T}_1'$, which generate D_b-blocks for all $b \neq a$ from every domain element in C_a and D_b-blocks for all $b \in \Lambda$ from domain elements in F_a. The rest of the argument is as in the proof of Theorem 32; see Fig. 14.

One can show that M' has a run with only infinite branches if and only if $(\mathcal{T}_1', \mathcal{A})$ Σ-query entails $(\mathcal{T}_2', \mathcal{A})$. By Theorem 12, Σ-query inseparability is also 2ExpTime-hard. ❑

6. Query Inseparability for Restricted Sets of Individuals

In the definition of Σ-query entailment and inseparability discussed so far we considered *all* tuples of individuals in the KBs that are certain answers to CQs. In this section, we refine this notion by allowing the user to define the set of individuals he is interested in. This leads to the following generalisation of Definition 1.

Definition 34. Let $\mathcal{K}_1$ and $\mathcal{K}_2$ be KBs, Σ a relational signature and Γ an individual signature. We say that $\mathcal{K}_1$ (Σ, Γ)-*query entails* $\mathcal{K}_2$ if

$$\mathcal{K}_2 \models \boldsymbol{q}(\boldsymbol{a}) \text{ implies } \boldsymbol{a} \subseteq \mathsf{ind}(\mathcal{K}_1) \text{ and } \mathcal{K}_1 \models \boldsymbol{q}(\boldsymbol{a}), \quad \text{for all } \Sigma\text{-CQs } \boldsymbol{q}(\boldsymbol{x}) \text{ and all tuples } \boldsymbol{a} \text{ in } \mathsf{ind}(\mathcal{K}_2) \cap \Gamma.$$

KBs $\mathcal{K}_1$ and $\mathcal{K}_2$ are (Σ, Γ)-*query inseparable* if they (Σ, Γ)-query entail each other, in which case we write $\mathcal{K}_1 \equiv_{\Sigma,\Gamma} \mathcal{K}_2$.

By definition, $\mathcal{K}_1$ Σ-query entails $\mathcal{K}_2$ if and only if $\mathcal{K}_1$ (Σ, Γ)-query entails $\mathcal{K}_2$ for all individual signatures Γ. Also, if $\Gamma \supseteq \mathsf{ind}(\mathcal{K}_2)$ then $\mathcal{K}_1$ Σ-query entails $\mathcal{K}_2$ in case $\mathcal{K}_1$ (Σ, Γ)-query entails $\mathcal{K}_2$. As only the intersection $\mathsf{ind}(\mathcal{K}_2) \cap \Gamma$ is relevant for (Σ, Γ)-query entailment, in what follows without loss of generality we assume that $\Gamma \subseteq \mathsf{ind}(\mathcal{K}_2)$.

One can analyse (Σ, Γ)-query entailment between KBs, one of which is inconsistent, in a way similar to Σ-query entailment. So, in the sequel we only focus on consistent KBs without mentioning this explicitly. The main difference between Σ-query entailment and (Σ, Γ)-query entailment can already be seen on KBs with empty TBoxes and empty individual signature Γ. Note that for KBs with empty TBoxes, Σ-query entailment is trivial as $\mathcal{K}_1 = (\emptyset, \mathcal{A}_1)$ Σ-query entails $\mathcal{K}_2 = (\emptyset, \mathcal{A}_2)$ if and only if, for all $a, b \in \mathsf{ind}(\mathcal{K}_2)$ with $A(a) \in \mathcal{A}_2$, $A \in \Sigma$, or $P(a, b) \in \mathcal{A}_2$, $P \in \Sigma$, it follows that $A(a) \in \mathcal{A}_1$ or $P(a, b) \in \mathcal{A}_1$, respectively. Note also that $(\Sigma, \emptyset)$-query entailment between any KBs $\mathcal{K}_1$ and $\mathcal{K}_2$ means that all *Boolean* Σ-CQs entailed by $\mathcal{K}_2$ are entailed by $\mathcal{K}_1$ as well.

Theorem 35. *Checking* $(\Sigma, \emptyset)$-*query entailment and* $(\Sigma, \emptyset)$-*inseparability of KBs with empty TBoxes are both* NP-*hard for data complexity.*

Proof. Let $\mathcal{K}_i = (\emptyset, \mathcal{A}_i)$, for $i = 1, 2$. Clearly, $\mathcal{K}_1$ $(\Sigma, \emptyset)$-query entails $\mathcal{K}_2$ if and only if there exists a $(\Sigma, \emptyset)$-homomorphism from (the interpretation corresponding to) $\mathcal{A}_2$ to $\mathcal{A}_1$. The latter problem is the standard homomorphism problem for relational structures which is known to be NP-hard [35]. To show NP-hardness of $(\Sigma, \emptyset)$-query inseparability, observe that there is a $(\Sigma, \emptyset)$-homomorphism from $\mathcal{A}_2$ to $\mathcal{A}_1$ if and only if $(\emptyset, \mathcal{A}_1 \uplus \mathcal{A}_2)$ and $(\emptyset, \mathcal{A}_1)$ are $(\Sigma, \emptyset)$-query inseparable, where $\mathcal{A}_1 \uplus \mathcal{A}_2$ is the *disjoint union* of $\mathcal{A}_1$ and $\mathcal{A}_2$. ❑

We now show that checking the existence of a homomorphism between ABoxes is the only additional source of complexity for (Σ, Γ)-query entailment compared to Σ-query entailment. In particular, for data complexity, checking (Σ, Γ)-query entailment is in NP for all of our DLs; for combined complexity, it is either NP-complete or harder than NP, in which case it is of the same complexity as Σ-query entailment. We begin by generalising the semantic characterisation of Σ-query entailment via finite Σ-homomorphic embeddability of materialisations:

Theorem 36. *Suppose* $\mathcal{K}_i$ *is a KB with a materialisation* $\mathcal{I}_i$, *for* $i = 1, 2$, Σ *is a relational signature, and* $\Gamma \subseteq \mathsf{ind}(\mathcal{K}_2)$. *Then* $\mathcal{K}_1$ (Σ, Γ)-*query entails* $\mathcal{K}_2$ *if and only if* $\mathcal{I}_2$ *is finitely* (Σ, Γ)-*homomorphically embeddable into* $\mathcal{I}_1$.

Proof. A straightforward extension of the proof of Theorem 5. ❑

Now we generalise the game-theoretic characterisation provided by Theorem 13. Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be materialisations obtained by unravelling finite generating structures $\mathcal{G}_1$ and $\mathcal{G}_2$ for KBs $\mathcal{K}_1$ and $\mathcal{K}_2$, respectively, and let $\mathcal{M}_2^{\mathsf{ind}}$ be the subinterpretation of $\mathcal{M}_2$ with domain $\mathsf{ind}(\mathcal{K}_2)$.

Theorem 37. *Let* $\Gamma \subseteq \mathsf{ind}(\mathcal{K}_2)$. *Then* $\mathcal{M}_2$ *is finitely* (Σ, Γ)-*homomorphically embeddable into* $\mathcal{M}_1$ *if and only if the following conditions are satisfied*:

(win$_{\mathsf{wit}}$) *for any* $u \in \Delta^{\mathcal{G}_2} \setminus \mathsf{ind}(\mathcal{K}_2)$ *and* $n < \omega$, *there exists* $\sigma \in \Delta^{\mathcal{M}_1}$ *such that player 1 has an* n-*winning strategy in the game* $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ *starting from* $(u \mapsto \sigma)$;

(h+win$_{\mathsf{ind}}$) *for any* $n < \omega$, *there is a* (Σ, Γ)-*homomorphism* $h_n \colon \mathcal{M}_2^{\mathsf{ind}} \to \mathcal{M}_1$ *such that, for every* $a \in \mathsf{ind}(\mathcal{K}_2)$, *player 1 has an* n-*winning strategy in the game* $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ *starting from* $(a \mapsto h_n(a))$.

Proof. A straightforward modification of the proof of Theorem 13. ❑

Condition **(win$_{\mathsf{wit}}$)** is the restriction of **(win)** in Theorem 13 to $u \in \Delta^{\mathcal{G}_2} \setminus \mathsf{ind}(\mathcal{K}_2)$, and so can be reduced, by Lemma 28, to conditions for games on the finite generating structures $\mathcal{G}_1$ and $\mathcal{G}_2$. We now show that **(h+win$_{\mathsf{ind}}$)** can also be reduced to certain conditions on $\mathcal{G}_1$ and $\mathcal{G}_2$. In contrast to the case where one could not restrict the set of

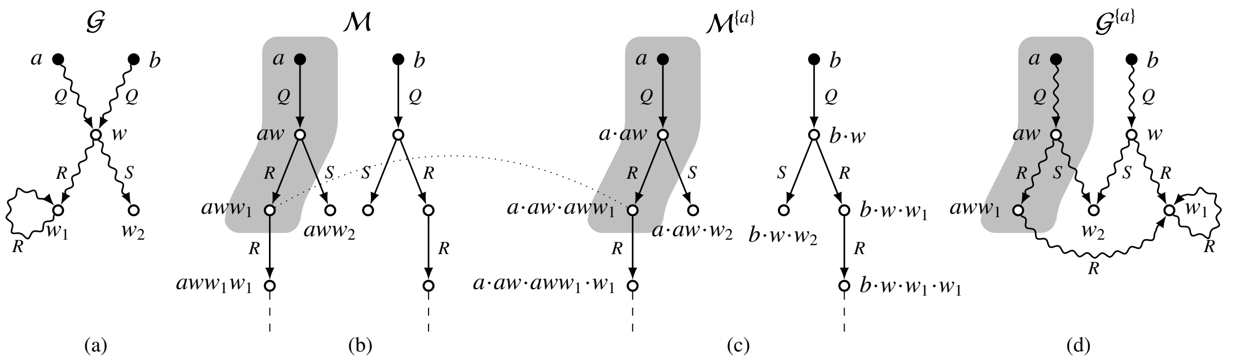


Figure 18: (a) Generating structure $\mathcal{G}$, (b) its unravelling $\mathcal{M}$, (c) the unravelling $\mathcal{M}^{\{a\}}$ of the extended generating structure $\mathcal{G}^{\{a\}}$, and (d) the extended generating structure $\mathcal{G}^{\{a\}}$ ($\Pi_{\{a\}}$ is shaded).

individuals and individuals were mapped to themselves (cf. **(abox)**), we now require a (Σ, Γ)-homomorphism h from $\mathcal{M}_2^{\mathsf{ind}}$ to an extension of $\mathcal{G}_1$, which is obtained by a partial unravelling of $\mathcal{G}_1$ defined as follows.

Consider $\mathcal{G} = (\Delta^{\mathcal{G}}, \cdot^{\mathcal{G}}, \rightsquigarrow)$ and let $X \subseteq \Delta^{\mathcal{G}}$, where either $X \subseteq \mathsf{ind}(\mathcal{K})$ or $X = \{w\}$ for some $w \in \Delta^{\mathcal{G}} \setminus \mathsf{ind}(\mathcal{K})$. We associate with X a finite prefix-closed set Π_X of paths π of the form $w_0 \cdots w_n$ such that $w_0 \in X$ and $w_i \rightsquigarrow w_{i+1}$, for $i < n$ (cf. Definition 6). The structure $\mathcal{G}^X = (\Delta^{\mathcal{G}^X}, \cdot^{\mathcal{G}^X}, \rightsquigarrow^X)$ is defined by first taking $\Delta^{\mathcal{G}^X} = \Delta^{\mathcal{G}} \cup \Pi_X$,

$$\pi \rightsquigarrow^X w \qquad \text{if} \quad \pi \in \Delta^{\mathcal{G}} \cup \Pi_X, \ \ \mathsf{tail}(\pi) \rightsquigarrow w \ \ \text{but} \ \ \pi w \notin \Pi_X,$$

$$\pi \rightsquigarrow^X \pi w \qquad \text{if} \quad \pi, \pi w \in \Pi_X,$$

$A^{\mathcal{G}^X} = A^{\mathcal{G}} \cup \{\pi \in \Pi_X \mid \mathsf{tail}(\pi) \in A^{\mathcal{G}}\}$, for each concept name A, $P^{\mathcal{G}^X} = P^{\mathcal{G}}$, for each role name P, and $(\pi, \pi')^{\mathcal{G}^X} = (\mathsf{tail}(\pi), \mathsf{tail}(\pi'))^{\mathcal{G}}$, for each arrow $\pi \rightsquigarrow^X \pi'$. Then we remove all 'disconnected' elements from $\mathcal{G}^X$ to make sure that each $\Delta^{\mathcal{G}^X} \setminus \mathsf{ind}(\mathcal{K})$ is reachable from $\mathsf{ind}(\mathcal{K})$ via a path of $\rightsquigarrow$ arrows. (Note that $\mathcal{G}^X$ depends on Π_X, which will always be clear from the context.)

Observe that the unravelling $\mathcal{M}^X$ of $\mathcal{G}^X$ is isomorphic to the unravelling $\mathcal{M}$ of $\mathcal{G}$. We denote the natural isomorphism from $\mathcal{M}^X$ onto $\mathcal{M}$ by g. Note that if $X \subseteq \mathsf{ind}(\mathcal{K})$ then, on $g^{-1}(\Pi_X)$, the function g coincides with tail; otherwise, if $X = \{w_0\}$ then $g(\delta \cdot w_0) = g(\delta)w_0$, for $\delta \cdot w_0 \in \Delta^{\mathcal{M}^X}$, and $g(\delta \cdot \pi \cdot \pi w) = g(\delta \cdot \pi)w$, for $\delta \cdot \pi \in \Delta^{\mathcal{M}^X}$ and $\pi, \pi w \in \Pi_X$.

Example 38. Consider the generating structure $\mathcal{G}$ depicted in Fig. 18a. The extended generating structure $\mathcal{G}^{\{a\}}$, with $\Pi_{\{a\}} = \{a, aw, aww_1\}$, is shown in Fig. 18d. Observe that the shaded part, $\Pi_{\{a\}}$, of $\mathcal{G}^{\{a\}}$ coincides with the shaded part of the unravelling $\mathcal{M}$ of $\mathcal{G}$ and that the unravelling $\mathcal{M}^{\{a\}}$ of $\mathcal{G}^{\{a\}}$ is isomorphic to $\mathcal{M}$ so that, on the shaded area, the natural isomorphism g coincides with tail: for example, $g(a \cdot aw \cdot aww_1) = aww_1 = \mathsf{tail}(a \cdot aw \cdot aww_1)$, as shown by the dotted line in Fig. 18.

Next, consider the generating structure $\mathcal{G}_1$ depicted in Fig. 19a. The extended generating structure $\mathcal{G}_1^{\{w\}}$, with $\Pi_{\{w\}} = \{w, ww'\}$, is shown in Fig. 19d. Note that w' does not belong to $\mathcal{G}_1^{\{w\}}$ because it would not be connected to any other domain element. Observe again that the unravelling $\mathcal{M}_1^{\{w\}}$ of $\mathcal{G}_1^{\{w\}}$ is isomorphic to the unravelling $\mathcal{M}_1$ of $\mathcal{G}_1$: the natural isomorphism g is such that $g(c \cdot w_i \cdot w) = cw_iw$ and $g(c \cdot w_i \cdot w \cdot ww') = g(c \cdot w_i \cdot w)w'$, for $i = 1, 2$. Note also that both unravellings contain two isomorphic copies of $\Pi_{\{w\}}$ from $\mathcal{G}_1^{\{w\}}$ (shaded in Fig. 19d): for example, the elements $cw_1\pi$ and $cw_2\pi$ in $\mathcal{M}_1$ are copies of $\pi \in \Pi_{\{w\}}$.

It will be convenient to consider h-images of maximal Σ-connected components of $\mathcal{M}_2^{\mathsf{ind}}$ separately. A subset Δ_0 of the domain $\Delta^{\mathcal{M}}$ of an interpretation $\mathcal{M}$ is called Σ-*connected* if, for any $u, u' \in \Delta_0$, there are $u_0, \dots, u_n$ such that $u_0 = u$, $u_n = u'$ and, for each $i < n$, there exists a Σ-role R with $(u_i, u_{i+1}) \in R^{\mathcal{M}}$.

Theorem 39. *Condition* **(h+win$_{\mathsf{ind}}$)** *holds if and only if, for every maximal Σ-connected component Δ_0 of $\mathcal{M}_2^{\mathsf{ind}}$, there are $X \subseteq \Delta^{\mathcal{G}_1}$, a structure $\mathcal{G}_1^X$ and a map $h\colon \Delta_0 \to \Delta^{\mathcal{G}_1^X}$ such that either $X \subseteq \mathsf{ind}(\mathcal{K}_1)$ or $X = \{w_0\}$ for $w_0 \in \Delta^{\mathcal{G}_1} \setminus \mathsf{ind}(\mathcal{K}_1)$, and $h(\Delta_0) = \Pi_X$,*

(h$^\Gamma$) *$h(a) = a$, for any $a \in \Gamma \cap \Delta_0 \cap \mathsf{part}_\Sigma^{\mathcal{M}_2}$, and $\boldsymbol{t}_\Sigma^{\mathcal{M}_2}(a) \subseteq \boldsymbol{t}_\Sigma^{\mathcal{G}_1^X}(h(a))$ and $\boldsymbol{r}_\Sigma^{\mathcal{M}_2}(a, b) \subseteq \boldsymbol{r}_\Sigma^{\mathcal{G}_1^X}(h(a), h(b))$, for any $a, b \in \Delta_0$,*

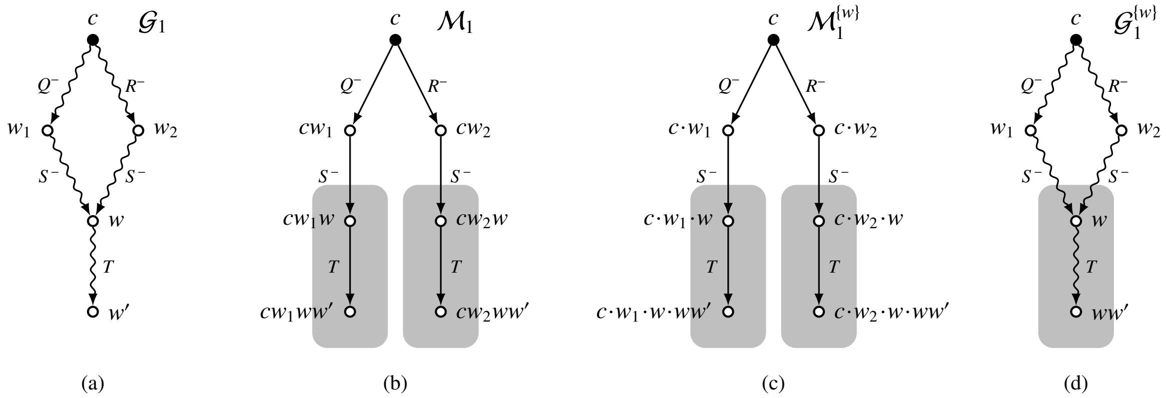


Figure 19: (a) Generating structure $\mathcal{G}_1$, (b) its unravelling $\mathcal{M}_1$, (c) the unravelling $\mathcal{M}_1^{\{w\}}$ of the extended generating structure $\mathcal{G}_1^{\{w\}}$, and (d) the extended generating structure $\mathcal{G}_1^{\{w\}}$ ($\Pi_{\{w\}}$ is shaded).

(h+win$_X$) *for each $\pi \in \Pi_X$, there exists a state $\mathfrak{a}_\pi = (\Xi_\pi \mapsto \pi, \Psi_\pi)$ such that $\Xi_\pi \supseteq h^{-1}(\pi)$, player 1 has an ω-winning strategy in $G^g_\Sigma(\mathcal{G}_2, \mathcal{G}_1^X)$ from $\mathfrak{a}_\pi$, and if $X = \{w_0\}$ then the $\mathfrak{a}_\pi$ are co-ordinated in the following sense:*

$$\mathfrak{a}_\pi \textit{ is a valid response to the challenge } \Psi_{\pi w} \textit{ in the state } \mathfrak{a}_{\pi w} \textit{ in } G^g_\Sigma(\mathcal{G}_2, \mathcal{G}_1^X), \textit{ for any } \pi, \pi w \in \Pi_X. \tag{6}$$

Proof. ($\Rightarrow$) Let Δ_0 be a maximal Σ-connected component of $\mathcal{M}_2^{\mathsf{ind}}$. For any $n < \omega$, take a (Σ, Γ)-homomorphism h_n from $\mathcal{M}_2^{\mathsf{ind}}$ to $\mathcal{M}_1$ such that, for every $a \in \mathsf{ind}(\mathcal{K}_2)$, player 1 has an n-winning strategy in the game $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ starting from $(a \mapsto h_n(a))$. Two cases are possible now.

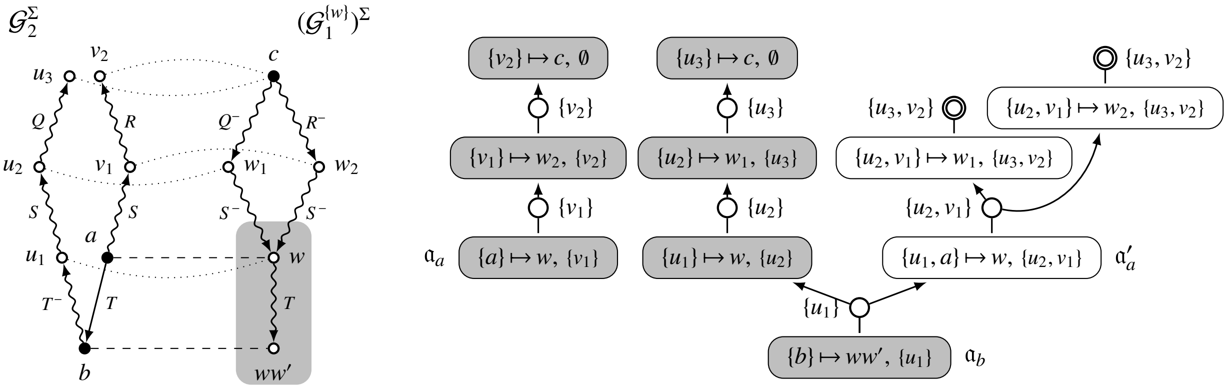
– If $h_n(\Delta_0) \cap \mathsf{ind}(\mathcal{K}_1) \neq \emptyset$ for infinitely many $n < \omega$ then, since $h_n(\Delta_0)$ is Σ-connected, the number of distinct sets $h_n(\Delta_0)$ with $h_n(\Delta_0) \cap \mathsf{ind}(\mathcal{K}_1) \neq \emptyset$ is finite. Thus, by the pigeonhole principle, there is an infinite set $\mathbb{H}$ of natural numbers n with $h_n(\Delta_0) \cap \mathsf{ind}(\mathcal{K}_1) \neq \emptyset$ such that the restrictions of all h_n to Δ_0 coincide. Let h be the restriction of some h_n, for $n \in \mathbb{H}$, to Δ_0. We set $X = h(\Delta_0) \cap \mathsf{ind}(\mathcal{K}_1)$ and $\Pi_X = h(\Delta_0)$. Using the map h, one can now construct the required starting states and ω-winning strategies in $G^g_\Sigma(\mathcal{G}_2, \mathcal{G}_1^X)$ in exactly the same way as in the proof of $(a) \Rightarrow (b)$ in Lemma 28.

– Otherwise, $h_n(\Delta_0) \cap \mathsf{ind}(\mathcal{K}_1) = \emptyset$ for infinitely many $n < \omega$ and, as Δ_0 is Σ-connected, by the pigeonhole principle there exists $w_0 \in \Delta^{\mathcal{G}_1} \setminus \mathsf{ind}(\mathcal{K}_1)$ such that, for infinitely many $n < \omega$, $h_n(\Delta_0)$ is a tree with root $\sigma^n w_0 \in \Delta^{\mathcal{M}_1}$. We set $X = \{w_0\}$ and can define, again by the pigeonhole principle, Π_X in such a way that there is an infinite set $\mathbb{H}$ of natural numbers n such that $h_n(\Delta_0) = \{\sigma^n \pi \mid \pi \in \Pi_X\}$. Then, for every $a \in \Delta_0$, there is $h(a) \in \Pi_X$ such that $h_n(a) = \sigma^n h(a)$, for all $n \in \mathbb{H}$. Using the map h, one can now construct the required starting states satisfying (6), and ω-winning strategies in $G^g_\Sigma(\mathcal{G}_2, \mathcal{G}_1^X)$ in exactly the same way as in the proof of $(a) \Rightarrow (b)$ in Lemma 28.

($\Leftarrow$) Let Δ_0 be a maximal Σ-connected component of $\mathcal{M}_2^{\mathsf{ind}}$. Set $\Gamma' = \Gamma \cap \Delta_0$. It is sufficient to show that **(h+win$_{\mathsf{ind}}$)** holds for Δ_0 in place of $\mathsf{ind}(\mathcal{K}_2)$, i.e., for any $n < \omega$, there exists a (Σ, Γ')-homomorphism h_n from $\mathcal{M}_{\Delta_0}$ to $\mathcal{M}_1$ such that player 1 has an n-winning strategy in the game $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ starting from $(a \mapsto h_n(a))$ for all $a \in \Delta_0$, where $\mathcal{M}_{\Delta_0}$ is the interpretation $\mathcal{M}_2$ relativised to the domain Δ_0. Let $X \subseteq \Delta^{\mathcal{G}_1}$, $h\colon \Delta_0 \to \Delta^{\mathcal{G}_1^X}$, and $n < \omega$ be given, where X and h satisfy the conditions of the theorem.

– If $X \subseteq \mathsf{ind}(\mathcal{K}_1)$ then we set $h_n(a) = h(a)$ for all $a \in \Delta_0$. It is readily checked that h_n is a (Σ, Γ')-homomorphism from $\mathcal{M}_{\Delta_0}$ to $\mathcal{M}_1$. For each $a \in \Delta_0$, by Lemma 28, player 1 has an n-winning strategy in the game $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1^X)$ from some $(a \mapsto \delta)$ with $\mathsf{tail}(\delta) = h(a)$. Then the natural isomorphism g from $\mathcal{M}_1^X$ onto $\mathcal{M}_1$ translates this strategy into an n-winning strategy in the game $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ from $(a \mapsto h(a))$.

– Otherwise, $X = \{w_0\}$ for $w_0 \in \Delta^{\mathcal{G}_1} \setminus \mathsf{ind}(\mathcal{K}_1)$. Since $\Xi_{w_0} \supseteq h^{-1}(w_0)$, by Lemma 28, for each $a \in h^{-1}(w_0)$, player 1 has an n-winning strategy in $G_\Sigma(\mathcal{G}_1, \mathcal{M}_1^X)$ from some $(a \mapsto \delta)$ with $\mathsf{tail}(\delta) = \sigma w_0 \in \Delta^{\mathcal{M}_1}$. Then the natural



(a) (b)

Figure 20: Co-ordination of starting states: (a) generating structure $\mathcal{G}_2^\Sigma$ and extended generating structure $(\mathcal{G}_1^X)^\Sigma$ with $\Pi_X = \{w, ww'\}$; (b) the relevant fragment of the game graph.

isomorphism g from $\mathcal{M}_1^X$ onto $\mathcal{M}_1$ translates each such strategy into an n-winning strategy in $G_\Sigma(\mathcal{G}_2, \mathcal{M}_1)$ from $(a \mapsto \sigma w_0)$.

We set $h_n(a) = \sigma\pi$, for each $a \in h^{-1}(\pi)$ and $\pi \in \Pi_X$. Then h_n is a (Σ, Γ')-homomorphism from $\mathcal{M}_{\Delta_0}$ to $\mathcal{M}_1$. We show by induction that, for all $\pi \in \Pi_X$,

$$\text{player 1 has an } n\text{-winning strategy in } G_\Sigma(\mathcal{G}_2, \mathcal{M}_1) \text{ from } (a \mapsto h_n(a)), \text{ for each } a \in h^{-1}(\pi). \quad (7)$$

For $\pi = w$, this holds by the definition of σ. Now assume that (7) has been proved for π and let $\pi w \in \Pi_X$. By the induction hypothesis and the proof of Lemma 28, it suffices to show that $\mathfrak{a}_\pi$ is a response of player 1 to the challenge $\Psi_{\pi w}$ in the state $\mathfrak{a}_{\pi w}$ of $G_\Sigma^g(\mathcal{G}_2, \mathcal{G}_1^X)$, which is guaranteed by (6).

This completes the proof of the theorem. ❑

Condition (6) is necessary for co-ordinating the starting states of the games when $X = \{w_0\}$, for $w_0 \in \Delta^{\mathcal{G}_1} \setminus \mathsf{ind}(\mathcal{K}_1)$. On the other hand, if $\Gamma \supseteq \mathsf{ind}(\mathcal{K}_2)$ then all Σ-participating individuals in $\mathsf{ind}(\mathcal{K}_2)$ must be mapped to themselves, and so condition (6) is not applicable in this case. The following example shows that without (6) we cannot guarantee that **(h+win$_{\mathsf{ind}}$)** holds, and so $\mathcal{M}_2$ may not be finitely (Σ, Γ)-homomorphically embeddable into $\mathcal{M}_1$.

Example 40. Consider KBs $\mathcal{K}_2$ and $\mathcal{K}_1$ and a relational signature Σ such that $\mathsf{ind}(\mathcal{K}_2) = \{a, b\}$, $\mathsf{ind}(\mathcal{K}_1) = \{c\}$ and their generating structures $\mathcal{G}_2^\Sigma$ and $(\mathcal{G}_1^{\{w\}})^\Sigma$ are as in Fig. 20a, with $\Pi_{\{w\}} = \{w, ww'\}$ (see also Figs. 19a and d for $\mathcal{G}_1$ and $\mathcal{G}_1^{\{w\}}$, respectively). Let $\Gamma = \emptyset$ and suppose that $h(a) = w$ and $h(b) = ww'$ (see the dashed lines in Fig. 20a). Player 1 has ω-winning strategies in $G_\Sigma^g(\mathcal{G}_2, \mathcal{G}_1^X)$ from the states $\mathfrak{a}_a = (\{a\} \mapsto w, \{v_1\})$ and $\mathfrak{a}_b = (\{b\} \mapsto ww', \{u_1\})$: see the dotted lines in Fig. 20a and the game graph in Fig. 20b. However, the two starting states, $\mathfrak{a}_a$ and $\mathfrak{a}_b$, do not satisfy the co-ordination condition (6). In fact, the map they induce is not a (Σ, Γ)-homomorphism from $\mathcal{M}_2$ to $\mathcal{M}_1$ because it sends a to cw_2w and b to cw_1ww', which are not connected by the role T in $\mathcal{M}_1$. Moreover, it is not hard to see that there is no (Σ, Γ)-homomorphism from $\mathcal{M}_2$ to $\mathcal{M}_1$. Indeed, our co-ordination condition means that we have to choose appropriate starting states for each of the elements in Π_X. So, we can pick $\mathfrak{a}_b$ for ww', from which, as we noted above, player 1 has an ω-winning strategy. We cannot, however, choose $\mathfrak{a}_a$ for w because $\Psi_{ww'} = \{u_1\}$, and so, by (6), Ξ_w must contain u_1 (along with a) but the 'uncoordinated' starting state $\mathfrak{a}_a$ does not include u_1. Thus, we have to take $\mathfrak{a}'_a = (\{u_1, a\} \mapsto w, \{u_2, v_1\})$ for w, from which player 1 has no ω-winning strategy: see the graph in Fig. 20b, where all the paths from $\mathfrak{a}'_a$ lead to dead-ends.

Finally, we obtain the following tight complexity results for KB (Σ, Γ)-query entailment and inseparability.

Theorem 41. *For combined complexity, both KB (Σ, Γ)-query entailment and inseparability are* 2ExpTime*-complete for* Horn-$\mathcal{ALCHI}$ *and* Horn-$\mathcal{ALCI}$; ExpTime*-complete for Horn-$\mathcal{ALCH}$, Horn-$\mathcal{ALC}$, DL-Lite$_{horn}^{\mathcal{H}}$ and DL-Lite$_{core}^{\mathcal{H}}$; and* NP*-complete for $\mathcal{ELH}_\bot^{dr}$, $\mathcal{EL}$, DL-Lite$_{horn}$ and DL-Lite$_{core}$. For data complexity, these problems are* NP*-complete.*

Proof. Note first that the size of X and Π_X is bounded by the size of $\mathsf{ind}(\mathcal{K}_2)$, so the size of $\mathcal{G}_1^X$ is polynomial in the size of $\mathcal{G}_1$ and $\mathsf{ind}(\mathcal{K}_2)$. Note also that if $\mathcal{G}_1$ is a forward generating structure then so is $\mathcal{G}_1^X$; if $\mathcal{G}_1$ is a functional generating structure then so is $\mathcal{G}_1^X$; and if $\mathcal{G}_1$ satisfies **(lite$_1$)** and **(lite$_2$)** then so does $\mathcal{G}_1^X$.

We start with an NP algorithm for data complexity. Let $\mathcal{G}_i$ be a generating structure for a KB $\mathcal{K}_i$, $i = 1, 2$. For each maximal Σ-connected component Δ_0 of $\mathcal{M}_2^{\mathsf{ind}}$, the algorithm performs two NP steps: (*i*) it guesses sets X, Π_X and a map h from Δ_0 onto Π_X, computes $\mathcal{G}_1^X$, and checks whether $(\mathbf{h}^\Gamma)$ is satisfied; then (*ii*) it guesses sets Ξ_π and Ψ_π satisfying (6) if $X \not\subseteq \mathsf{ind}(\mathcal{K}_1)$, for each $\pi \in \Pi_X$, and finally checks whether $(\mathbf{h{+}win}_X)$ holds. It is not hard to see both (*i*) and (*ii*) can be done in polynomial time in the size of $\mathsf{ind}(\mathcal{K}_1)$ and $\mathsf{ind}(\mathcal{K}_2)$.

It is easy to see that for $\mathcal{ELH}^{dr}_\bot$ and *DL-Lite$_{horn}$* KBs, the algorithm above provides an NP upper bound for the combined complexity as well. For the more expressive DLs, the upper bounds for combined complexity stay the same as before because there is at most an exponential number of distinct sets Π_X, maps h and states $\mathfrak{a}_\pi$. The ExpTime- and 2ExpTime-hardness results also carry over from Σ-query inseparability and Σ-query entailment, and NP-hardness follows from Theorem 35. ❑

7. Related Work and Applications

In this section, we discuss the relationship between (Σ, Γ)-query inseparability and knowledge exchange, TBox inseparability, and query-based comparison of OBDA specifications. Σ-query inseparability of KBs has not been investigated systematically before. Note, however, that the polynomial upper bound for $\mathcal{EL}$ was established as a preliminary step to study Σ-query inseparability of TBoxes [36], and that this notion was also used to study forgetting in *DL-Lite*$^{\mathcal{N}}_{bool}$ [37].

7.1. Knowledge Exchange

For the motivation of studying knowledge exchange between KBs and illustrating examples, we refer the reader to Section 1. Here we establish a tight link between deciding Σ-query inseparability and deciding the membership problem for universal CQ-solutions. We also consider the connection between (Σ, Γ)-query inseparability and the membership problem for universal CQ-solutions with nulls.

Assume (without loss of generality) that $\mathcal{K}_1$ and $\mathcal{K}_2$ are KBs given in disjoint relational signatures Σ_1 and Σ_2. Suppose also that $\mathcal{T}_{12}$ consists of inclusions of the form $S_1 \sqsubseteq S_2$ such that the S_i are concept or role names in Σ_i. Then the problem of deciding whether $\mathcal{K}_1 \cup \mathcal{T}_{12} \equiv_{\Sigma_2} \mathcal{K}_2$ is called the *membership problem for universal CQ-solutions*. For any of our 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 $\mathcal{L}$, and so the upper complexity bounds for Σ-query inseparability can be applied directly to obtain upper bounds for the membership problem for universal CQ-solutions. The following result establishes the converse:

Theorem 42. *Σ-query entailment for any of our DLs $\mathcal{L}$ is* LogSpace*-reducible to the membership problem for universal CQ-solutions in $\mathcal{L}$.*

The proof uses the construction from the proof of Theorem 12 and is given in Appendix A. As a consequence of Theorems 42, 29 and 33 we obtain the following:

Theorem 43. *For combined complexity, the membership problem for universal CQ-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 $\mathcal{ELH}^{dr}_\bot$ and $\mathcal{EL}$. For data complexity, all these problems are* P*-complete.*

Note that the combined complexity of the membership problem for universal CQ-solutions remains open for *DL-Lite$_{core}$* and *DL-Lite$_{horn}$*.

In the case of *DL-Lite*$^{\mathcal{H}}_{core}$, we also obtain an ExpTime algorithm for checking the existence and computing universal CQ-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 CQ-solution (there are at most $O(2^{|\Sigma_2|})$ such KBs). This gives an ExpTime upper bound for the non-emptiness problem for universal CQ-solutions in *DL-Lite*$^{\mathcal{H}}_{core}$ [23].

A more flexible knowledge exchange model allows the target KB to use additional individuals (i.e., not only the individuals in $\mathcal{K}_1$), which however cannot be returned as certain answers [23]. These ‘anonymous’ individuals are similar to nulls in the standard approaches to incomplete databases, and intuitively represent objects the existence of which is implied by $\mathcal{K}_1 \cup \mathcal{T}_{12}$. The reader can find an illustrating example in Section 1. Formally, we say that a KB $\mathcal{K}_2$ with a relational signature Σ_2 is a *universal CQ-solution with nulls* for a KB $\mathcal{K}_1$ and a mapping specification $\mathcal{T}_{12}$ if $\mathcal{K}_1 \cup \mathcal{T}_{12} \equiv_{\Sigma_2,\mathsf{ind}(\mathcal{K}_1)} \mathcal{K}_2$ (which is equivalent to the definition given in [23]). Thus we obtain the following result:

Theorem 44. *For combined complexity, the membership problem for universal CQ-solutions with nulls 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* NP*-complete for* $\mathcal{ELH}^{dr}_{\bot}$ *and* $\mathcal{EL}$*. For data complexity, all these problems are* NP*-complete.*

Proof. The upper bounds follow from Theorem 41. The ExpTime and 2ExpTime lower bounds follow from Theorem 43, and the NP lower bound can be obtained from the proof of Theorem 35 by a straightforward modification. ❑

Again, the combined complexity of the membership problem for universal CQ-solutions with nulls remains open for *DL-Lite*$_{core}$ and *DL-Lite*$_{horn}$.

7.2. TBox Inseparability and OBDA Specifications

We remind the reader that, for a relational signature Σ, TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ are called Σ-*query inseparable* if, for *all* Σ-ABoxes $\mathcal{A}$, the KBs $(\mathcal{T}_1, \mathcal{A})$ and $(\mathcal{T}_2, \mathcal{A})$ are Σ-query inseparable. TBox Σ-query inseparability has been extensively studied; see, e.g., [17, 36, 24, 10]. 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.

For many DLs, TBox Σ-query inseparability is harder than KB query inseparability. For *DL-Lite*$_{horn}$, the space of relevant Σ-ABox counterexamples is of exponential size and, in fact, Σ-query inseparability of TBoxes is NP-hard [17], while Σ-query inseparability of KBs is in P. Similarly, we have seen that Σ-query inseparability of $\mathcal{EL}$ KBs is in P, while Σ-query inseparability of $\mathcal{EL}$ TBoxes is ExpTime-complete [36]. The complexity of TBox Σ-query inseparability for Horn-DLs extending *Horn*-$\mathcal{ALC}$ is not known.

The complexity of Σ-query inseparability of *DL-Lite*$^{\mathcal{H}}_{core}$ TBoxes was known to sit between PSpace and ExpTime [24]. Using the fact that witness Σ-ABoxes for Σ-query inseparability of *DL-Lite*$^{\mathcal{H}}_{core}$ TBoxes can always be chosen among the singleton Σ-ABoxes [24, Theorem 8], one can easily modify the proof of Theorem 32 to improve the PSpace lower bound:

Theorem 45. *TBox Σ-query inseparability of DL-Lite*$^{\mathcal{H}}_{core}$ *TBoxes is* ExpTime*-complete.*

For work on other notions of TBox inseparability and the corresponding notions of modules and forgetting, we refer the reader to [38, 12, 39, 40, 41, 42, 43, 44].

In ontology-based data access (OBDA), a TBox $\mathcal{T}$ provides a vocabulary for user queries, which is connected by a declarative mapping $\mathcal{M}$ to a data source schema S (see, e.g., [2, 45]). The pair $\mathcal{S} = (\mathcal{T}, \mathcal{M})$ is called an OBDA specification (sometimes, it also includes integrity constraints of the data source). For example, $\mathcal{M}$ can consist of implications $\forall \boldsymbol{x}\boldsymbol{y}\, (\varphi(\boldsymbol{x}, \boldsymbol{y}) \to \psi(\boldsymbol{x}))$, where $\varphi(\boldsymbol{x}, \boldsymbol{y})$ is a conjunction of atoms over S and $\psi(\boldsymbol{x})$ is a conjunction of atoms over the signature of $\mathcal{T}$ (in which case $\mathcal{M}$ is called a GAV mapping). For a data instance D over S and a CQ $\boldsymbol{q}(\boldsymbol{x})$, the certain answers to $\boldsymbol{q}(\boldsymbol{x})$ over D under the OBDA specification $\mathcal{S}$ are defined in the obvious way. In [46], the following generalisation of TBox Σ-query entailment is introduced to support the static analysis of OBDA specifications. Say that an OBDA specification $\mathcal{S}_1$ query entails an OBDA specification $\mathcal{S}_2$ if, for every CQ $\boldsymbol{q}(\boldsymbol{x})$ and every data instance D over S, the certain answers to $\boldsymbol{q}(\boldsymbol{x})$ over D under $\mathcal{S}_2$ are contained in the certain answers to $\boldsymbol{q}(\boldsymbol{x})$ over D under $\mathcal{S}_1$. It was shown [46] that the complexity of query entailment between OBDA specifications is closely linked to the complexity of Σ-query entailment. In fact, for GLAV, GAV, and linear mappings $\mathcal{M}$, and *DL-Lite*$^{\mathcal{H}}_{core}$ TBoxes $\mathcal{T}$, the tight complexity results obtained in this article for Σ-query entailment are used to obtain the same complexity for deciding query entailment between OBDA specifications.

8. 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 article 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 $G^f_\Sigma(\mathcal{G}_2, \mathcal{G}_1)$, 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 [24] 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 CQ-solutions in data exchange as well as algorithms for computing universal CQ-solutions in various DLs.

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

References

[1] A. Polleres, A. Hogan, R. Delbru, J. Umbrich, RDFS and OWL reasoning for linked data, in: The 9th International Summer School (RW 2013), volume 8067 of *Lecture Notes in Computer Science*, Springer, 2013, pp. 91–149.

[2] A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, R. Rosati, Linking data to ontologies, J. on Data Semantics 10 (2008) 133–173.

[3] M. Giese, D. Calvanese, P. Haase, I. Horrocks, Y. Ioannidis, H. Kllapi, M. Koubarakis, M. Lenzerini, R. Möller, M. Rodriguez-Muro, O. Özcep, R. Rosati, R. Schlatte, M. Schmidt, A. Soylu, A. Waaler, Scalable end-user access to big data, in: Big Data Computing, CRC Press, 2013.

[4] P. Hitzler, M. Krötzsch, S. Rudolph, Foundations of Semantic Web Technologies, Chapman & Hall/CRC, 2009.

[5] U. Hustadt, B. Motik, U. Sattler, Data complexity of reasoning in very expressive description logics, in: Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI), 2005, pp. 466–471.

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

[7] C. Lutz, D. Toman, F. Wolter, 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), 2009, pp. 2070–2075.

[8] R. Kontchakov, M. Zakharyaschev, An introduction to description logics and query rewriting, in: The 10th Int. Summer School on Reasoning Web (RW 2014), volume 8714 of *Lecture Notes in Computer Science*, Springer, 2014, pp. 195–244.

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

[10] B. Konev, M. Ludwig, D. Walther, F. Wolter, The logical difference for the lightweight description logic EL, J. of Artificial Intelligence Research (JAIR) 44 (2012) 633–708.

[11] H. Stuckenschmidt, C. Parent, S. Spaccapietra (Eds.), Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization, volume 5445 of *Lecture Notes in Computer Science*, Springer, 2009.

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

[13] P. Koopmann, R. A. Schmidt, Forgetting and uniform interpolation for ALC-ontologies with ABoxes, in: Informal Proc. of the 27th Int. Workshop on Description Logics, 2014, pp. 245–257.

[14] M. Arenas, P. Barceló, L. Libkin, F. Murlak, Foundations of Data Exchange, Cambridge University Press, 2014.

[15] M. Arenas, E. Botoeva, D. Calvanese, V. Ryzhikov, E. Sherkhonov, Exchanging description logic knowledge bases, in: Proc. of the 13th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2012), AAAI Press, 2012.

[16] M. Arenas, J. Pérez, J. L. Reutter, Data exchange beyond complete data, J. ACM 60 (2013) 28.

[17] R. Kontchakov, F. Wolter, M. Zakharyaschev, Logic-based ontology comparison and module extraction, with an application to DL-Lite, Artificial Intelligence 174 (2010) 1093–1141.

[18] P. Shvaiko, J. Euzenat, Ontology matching: State of the art and future challenges, IEEE Trans. Knowl. Data Eng. 25 (2013) 158–176.

[19] B. Cuenca Grau, B. Motik, Reasoning over ontologies with hidden content: The import-by-query approach, J. of Artificial Intelligence Research (JAIR) 45 (2012) 197–255.

[20] M. Krötzsch, S. Rudolph, P. Hitzler, Complexities of Horn description logics, ACM Trans. Comput. Log. 14 (2013) 2.

[21] A. Artale, D. Calvanese, R. Kontchakov, M. Zakharyaschev, The DL-Lite family and relations, Journal of Artificial Intelligence Research 36 (2009) 1–69.

[22] F. Baader, S. Brandt, C. Lutz, Pushing the EL envelope, in: Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI-05), 2005, pp. 364–369.

[23] M. Arenas, E. Botoeva, D. Calvanese, V. Ryzhikov, Exchanging OWL 2 QL knowledge bases, in: Proc. of the 23rd Int. Joint Conf. on Artificial Intelligence (IJCAI), IJCAI/AAAI, 2013.

[24] B. Konev, R. Kontchakov, M. Ludwig, T. Schneider, F. Wolter, M. Zakharyaschev, Conjunctive query inseparability of OWL 2 QL TBoxes, in: Proc. of the 25th AAAI Conf. on Artificial Intelligence (AAAI 2011), AAAI Press, 2011.

[25] T. Imieliński, W. Lipski, Incomplete information in relational databases, J. of the ACM 31 (1984) 761–791.

[26] Y. Kazakov, Consequence-driven reasoning for Horn SHIQ ontologies, in: Proc. of the 21st Int. Joint Conf. on Artificial Intelligence (IJCAI), 2009, pp. 2040–2045.

[27] R. Rosati, On conjunctive query answering in EL, in: Proc. of the 2007 Int. Workshop on Description Logics (DL), 2007.

[28] T. Eiter, G. Gottlob, M. Ortiz, M. Simkus, 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*, Springer, 2008, pp. 166–179.

[29] M. Krötzsch, S. Rudolph, P. Hitzler, Complexity boundaries for horn description logics, in: Proc. of the 22nd Nat. Conf. on Artificial Intelligence (AAAI 2007), 2007, pp. 452–457.

[30] S. Tobies, Complexity results and practical algorithms for logics in Knowledge Representation, Ph.D. thesis, LuFG Theoretical Computer Science, RWTH-Aachen, Germany, 2001.

[31] S. Brandt, Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and—what else?, in: Proc. of the 16th European Conf. on Artificial Intelligence (ECAI-2004), IOS Press, 2004, pp. 298–302.

[32] C. Lutz, F. Wolter, Conservative extensions in the lightweight description logic EL, in: Proc. of the 21st Int. Conf. on Automated Deduction (CADE), volume 4603 of *Lecture Notes in Computer Science*, Springer, 2007, pp. 84–99.

[33] R. Mazala, Infinite games, in: Automata, Logics, and Infinite Games, 2001, pp. 23–42.

[34] D. Kozen, Theory of Computation, Springer, 2006.

[35] C. Papadimitriou, Computational Complexity, Addison-Wesley, 1994.

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

[37] Z. Wang, K. Wang, R. W. Topor, J. Z. Pan, Forgetting for knowledge bases in DL-Lite, Ann. Math. Artif. Intell. 58 (2010) 117–151.

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

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

[40] N. Nikitina, S. Rudolph, (Non-)succinctness of uniform interpolants of general terminologies in the description logic EL, Artif. Intell. 215 (2014) 120–140.

[41] N. Nikitina, B. Glimm, 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*, Springer, 2012, pp. 394–409.

[42] C. Lutz, I. Seylan, F. Wolter, 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, 2012.

[43] P. Koopmann, R. A. Schmidt, Count and forget: Uniform interpolation of $\mathcal{SHQ}$-ontologies, in: Proc. of the 7th Int. Joint Conf. on Automated Reasoning (IJCAR), 2014, pp. 434–448.

[44] R. Nortje, K. Britz, T. Meyer, Reachability modules for the description logic $\mathcal{SRIQ}$, in: Proc. of 19th Int. Conf. on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-19), volume 8312 of *Lecture Notes in Computer Science*, Springer, 2013.

[45] M. Rodriguez-Muro, R. Kontchakov, M. Zakharyaschev, Ontology-based data access: Ontop of databases, in: Proc. of the 12th Int. Semantic Web Conf. (ISWC 2013), Part I, volume 8218 of *Lecture Notes in Computer Science*, Springer, 2013, pp. 558–573.

[46] M. Bienvenu, R. Rosati, Query-based comparison of OBDA specifications, in: Proc. of DL, 2015.

Appendix A. Proofs

Lemma 9. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a consistent KB with a Horn-$\mathcal{ALCHI}$ TBox in normal form and $\mathcal{M}$ the unravelling of $\mathcal{G}$. Then $\mathcal{M}$ is a model of $\mathcal{K}$. Moreover,*

- $\boldsymbol{\tau}_{\mathcal{T}}^{\mathcal{M}}(a) = \{ C \in \mathsf{con}(\mathcal{T}) \mid \mathcal{K} \models C(a) \}$, *for all* $a \in \mathsf{ind}(\mathcal{K})$,
- $\boldsymbol{\tau}_{\mathcal{T}}^{\mathcal{M}}(\sigma) = \boldsymbol{\tau}$, *for all* $\sigma \in \Delta^{\mathcal{M}}$ *with* $\mathsf{tail}(\sigma) = ([S], \boldsymbol{\tau})$.

Proof. First, we show that $a \in C^{\mathcal{M}}$ iff $\mathcal{K} \models C(a)$, for all $a \in \mathsf{ind}(\mathcal{K})$, and $\sigma \in C^{\mathcal{M}}$ iff $C \in \boldsymbol{\tau}$, for all $\sigma \in \Delta^{\mathcal{M}}$ with $\mathsf{tail}(\sigma) = ([S], \boldsymbol{\tau})$. We consider the following two cases for C:

1: $C = A$**.** For $a \in \mathsf{ind}(\mathcal{K})$, we clearly have $a \in A^{\mathcal{M}}$ iff $a \in A^{\mathcal{G}}$ iff $\mathcal{K} \models A(a)$. Similarly, for any $\sigma \in \Delta^{\mathcal{M}}$ with $\mathsf{tail}(\sigma) = ([S], \boldsymbol{\tau})$, we have $\sigma \in A^{\mathcal{M}}$ iff $([S], \boldsymbol{\tau}) \in A^{\mathcal{G}}$ iff $A \in \boldsymbol{\tau}$.

2: $C = \exists R.B$**.** Let $a \in (\exists R.B)^{\mathcal{M}}$. If there is $b \in \mathsf{ind}(\mathcal{K})$ with $(a, b) \in R^{\mathcal{M}}$ and $b \in B^{\mathcal{M}}$ then, by the construction of $\mathcal{M}$ and $\mathcal{G}$, there is some P with $P(a, b) \in \mathcal{A}$ and $\mathcal{T} \models P \sqsubseteq R$, whence $\mathcal{K} \models R(a, b)$. On the other hand, by item **1**, $\mathcal{K} \models B(b)$, whence $\mathcal{K} \models (\exists R.B)(a)$. If there is no $b \in \mathsf{ind}(\mathcal{K})$ with $(a, b) \in R^{\mathcal{M}}$ and $\mathcal{K} \models B(b)$, then $a \rightsquigarrow ([R], \boldsymbol{\tau})$, for some $\mathcal{T}$-type $\boldsymbol{\tau}$ such that $\mathcal{K} \models (\exists R.\boldsymbol{\tau})(a)$ and $B \in \boldsymbol{\tau}$, whence $\mathcal{K} \models (\exists R.B)(a)$.

Conversely, let $\mathcal{K} \models (\exists R.B)(a)$. If there is $b \in \mathsf{ind}(\mathcal{K})$ with $P(a, b) \in \mathcal{A}$, $\mathcal{T} \models P \sqsubseteq R$ and $\mathcal{K} \models B(b)$ then, by construction, $(a, b) \in R^{\mathcal{M}}$ and, by item **1**, $b \in B^{\mathcal{M}}$, whence $a \in (\exists R.B)^{\mathcal{M}}$. Otherwise, let $\boldsymbol{\tau}$ be a maximal $\mathcal{T}$-type such that $\mathcal{K} \models (\exists R.\boldsymbol{\tau})(a)$ and $B \in \boldsymbol{\tau}$. Then $a \rightsquigarrow ([R], \boldsymbol{\tau})$ and, by the construction of $\mathcal{G}$ and $\mathcal{M}$, $(a, a \cdot ([R], \boldsymbol{\tau})) \in R^{\mathcal{M}}$ and, by item **1**, $a \cdot ([R], \boldsymbol{\tau}) \in B^{\mathcal{M}}$, whence $a \in (\exists R.B)^{\mathcal{M}}$.

Now, suppose $\sigma \in (\exists R.B)^{\mathcal{M}}$. Then there is σ' such that $(\sigma, \sigma') \in R^{\mathcal{M}}$ and $\sigma' \in B^{\mathcal{M}}$. By construction, the following three options are possible.

- If $\sigma' = \sigma \cdot ([S'], \boldsymbol{\tau}')$ then $\mathcal{T} \models \boldsymbol{\tau} \sqsubseteq \exists S'.\boldsymbol{\tau}'$, $\mathcal{T} \models S' \sqsubseteq R$ and $B \in \boldsymbol{\tau}'$, whence $\mathcal{T} \models \boldsymbol{\tau} \sqsubseteq \exists R.B$, and so, as $\boldsymbol{\tau}$ is a $\mathcal{T}$-type, $\exists R.B \in \boldsymbol{\tau}$.
- If $\sigma = \sigma' \cdot ([S], \boldsymbol{\tau})$ with $\mathsf{tail}(\sigma') = ([S'], \boldsymbol{\tau}')$ then $\mathcal{T} \models \boldsymbol{\tau}' \sqsubseteq \exists S.\boldsymbol{\tau}$, $\mathcal{T} \models S \sqsubseteq R^-$ and $B \in \boldsymbol{\tau}'$. It follows that we have $\mathcal{T} \models \boldsymbol{\tau}' \sqsubseteq \exists R^-.\boldsymbol{\tau}$ and $B \in \boldsymbol{\tau}'$. Since $\boldsymbol{\tau}$ is *maximal*, it must contain $\exists R.B$ (for otherwise $\boldsymbol{\tau}' \sqcap \exists R^-.(\boldsymbol{\tau} \sqcap \forall R.\neg B)$ would be consistent).
- If $\sigma = \sigma' \cdot ([S], \boldsymbol{\tau})$ with $\sigma' = a \in \mathsf{ind}(\mathcal{K})$ then $\mathcal{K} \models (\exists S.\boldsymbol{\tau})(a)$, $\mathcal{T} \models S \sqsubseteq R^-$ and, by item **1**, $\mathcal{K} \models B(a)$. Thus, we have $\mathcal{K} \models (\exists R^-.t)(a)$ and $\mathcal{K} \models B(a)$. Again, since $\boldsymbol{\tau}$ is maximal it must contain $\exists R.B$.

Conversely, let $\exists R.B \in \boldsymbol{\tau}$. Then, by construction, $([S], \boldsymbol{\tau}) \rightsquigarrow ([R], \boldsymbol{\tau}')$, for some $\mathcal{T}$-type $\boldsymbol{\tau}'$ with $B \in \boldsymbol{\tau}'$. It follows then that $(\sigma, \sigma \cdot ([R], \boldsymbol{\tau}')) \in R^{\mathcal{M}}$ and, by item **1**, $(\sigma \cdot ([R], \boldsymbol{\tau}')) \in B^{\mathcal{M}}$, whence $\sigma \in (\exists R.B)^{\mathcal{M}}$.

Next, we show that $\mathcal{M}$ is a model of $(\mathcal{T}, \mathcal{A})$. Clearly, $\mathcal{M}$ is a model of $\mathcal{A}$. That $\mathcal{M} \models (C_1 \sqsubseteq C_2)$, for each $C_1 \sqsubseteq C_2 \in \mathcal{T}$, follows immediately from the two properties of $\boldsymbol{\tau}_{\mathcal{T}}^{\mathcal{M}}$, the fact that $\mathcal{T}$-types are closed under the concept inclusions in $\mathcal{T}$, and that $C \sqsubseteq \forall R.A$ is equivalent to $\exists R^-.C \sqsubseteq A$.

Consider now $R_1 \sqsubseteq R_2 \in \mathcal{T}$. Let $(\sigma, \sigma') \in R_1^{\mathcal{M}}$. If $\sigma = a \in \mathsf{ind}(\mathcal{K})$ and $\sigma = b \in \mathsf{ind}(\mathcal{K})$ then $\mathcal{K} \models R_1(a, b)$. Since $R_1 \sqsubseteq R_2 \in \mathcal{T}$, we obtain $\mathcal{K} \models R_2(a, b)$, whence $(\sigma, \sigma') \in R_2^{\mathcal{M}}$. If $\sigma' = \sigma \cdot ([R], \boldsymbol{\tau})$, for some R and $\boldsymbol{\tau}$, then, by the construction of $R_1^{\mathcal{M}}$, $\mathcal{T} \models R \sqsubseteq R_1$. Thus $\mathcal{T} \models R \sqsubseteq R_2$, and so $(\sigma, \sigma') \in R_2^{\mathcal{M}}$. The case of $\sigma = \sigma' \cdot ([R], \boldsymbol{\tau})$ is the mirror image. ❑

Theorem 12. *Let $\mathcal{L}$ be any of our DLs containing $\mathcal{EL}$ or having role inclusions. Then Σ-query entailment for consistent $\mathcal{L}$-KBs is* LogSpace*-reducible to Σ-query inseparability for $\mathcal{L}$-KBs.*

Proof. Let $\mathcal{K}_i = (\mathcal{T}_i, \mathcal{A}_i)$, $i = 1, 2$, be consistent $\mathcal{L}$-KBs and Σ a relational signature. We want to decide whether $\mathcal{K}_1$ Σ-query entails $\mathcal{K}_2$ assuming that we know how to decide Σ-query inseparability. Without loss of generality, we may assume that $\Sigma = \mathsf{sig}(\mathcal{K}_1) = \mathsf{sig}(\mathcal{K}_1) \cap \mathsf{sig}(\mathcal{K}_2)$. To show this, we note first that we can add trivial concept inclusions $A \sqsubseteq A$ and $\exists P.\top \sqsubseteq \exists P.\top$ to KBs to ensure that $\Sigma \subseteq \mathsf{sig}(\mathcal{K}_1) = \mathsf{sig}(\mathcal{K}_1) \cap \mathsf{sig}(\mathcal{K}_2)$. For symbols $S \in \mathsf{sig}(\mathcal{K}_1) \cap \mathsf{sig}(\mathcal{K}_2)$

that are not in Σ we introduce a fresh S^* and replace S by S^* in $\mathcal{K}_2$. Denote the resulting KB by $\mathcal{K}_2^*$. Then $\mathcal{K}_1$ Σ-query entails $\mathcal{K}_2$ iff $\mathcal{K}_1$ Σ^*-query entails $\mathcal{K}_2^*$ for $\Sigma^* = \mathsf{sig}(\mathcal{K}_1)$, as required.

Case 1: $\mathcal{L}$ has role inclusions.

Case 1.1: Assume that the *trivial interpretation* $\mathcal{I}_\emptyset$ with $|\Delta^{\mathcal{I}_\emptyset}| = 1$ and $S^{\mathcal{I}_\emptyset} = \emptyset$, for any symbol S, is a model of the $\mathcal{T}_i$ for $i = 1, 2$ (we show how the KBs $\mathcal{K}_1$ and $\mathcal{K}_2$ can be modified to ensure that this assumption holds in Case 1.2). Let $\mathcal{K}_i^i$ be a copy of $\mathcal{K}_i$ in which all symbols S are replaced by fresh symbols S_i, and let $\mathcal{K}_i'$ be the extension of $\mathcal{K}_i^i$ with $S_i \sqsubseteq S$, for all $S \in \Sigma$. The purpose of this construction is to avoid the interaction between the symbols used in $\mathcal{K}_1$ and the symbols used in $\mathcal{K}_2$ (as shown in Section 3 after the formulation of the theorem). We show that

$$\mathcal{K}_1 \ \Sigma\text{-query entails } \mathcal{K}_2 \quad \text{iff} \quad \mathcal{K}_1 \text{ and } \mathcal{K}_1' \cup \mathcal{K}_2' \text{ are } \Sigma\text{-query inseparable.}$$

The interesting direction is to show that if $\mathcal{K}_1$ Σ-query entails $\mathcal{K}_2$ then $\mathcal{K}_1$ Σ-query entails $\mathcal{K}_1' \cup \mathcal{K}_2'$. Suppose that $\mathcal{K}_1$ Σ-query entails $\mathcal{K}_2$. Then $\mathcal{K}_1$ Σ-query entails both $\mathcal{K}_1'$ and $\mathcal{K}_2'$. We use the following construction to 'merge' materialisations of the $\mathcal{K}_i'$. Let $\mathcal{M}_1$ be a materialisation of $\mathcal{K}_1$ and, for $i = 1, 2$, let $\mathcal{U}_i$ be a materialisation of $\mathcal{K}_i'$ obtained by unravelling a generating structure for $\mathcal{K}_i'$. By Lemma 9, $\mathcal{U}_i$ is a model of $\mathcal{K}_i'$. It should be clear that we can also assume that

$$\Delta^{\mathcal{U}_1} \cap \Delta^{\mathcal{U}_2} = \mathsf{ind}(\mathcal{K}_1) \cap \mathsf{ind}(\mathcal{K}_2). \tag{A.1}$$

Denote by $\mathcal{U}$ the union of $\mathcal{U}_1$ and $\mathcal{U}_2$ defined by setting $\Delta^{\mathcal{U}} = \Delta^{\mathcal{U}_1} \cup \Delta^{\mathcal{U}_2}$ and $S^{\mathcal{U}} = S^{\mathcal{U}_1} \cup S^{\mathcal{U}_2}$ for all concept and role names S. We show that

(i) $\mathcal{U}$ is a model of $\mathcal{K}_1' \cup \mathcal{K}_2'$, and

(ii) $\mathcal{U}$ is finitely $(\Sigma, \mathsf{ind}(\mathcal{K}_1) \cup \mathsf{ind}(\mathcal{K}_2))$-homomorphically embeddable into $\mathcal{M}_1$.

It will then follow, by Theorem 5, that $\mathcal{K}_1$ Σ-query entails $\mathcal{K}_1' \cup \mathcal{K}_2'$. Indeed, let $\mathcal{M}'$ be a materialisation of $\mathcal{K}_1' \cup \mathcal{K}_2'$. Since, by (i), $\mathcal{U}$ is a model of $\mathcal{K}_1' \cup \mathcal{K}_2'$, by Lemma 10, there is a homomorphism from (any finite subinterpretation of) $\mathcal{M}'$ to $\mathcal{U}$, and so, by (ii), from any finite subinterpretation of $\mathcal{M}'$ to $\mathcal{M}_1$.

Now, for item (i), recall that, for both $i = 1, 2$, the trivial interpretation is a model of the TBox of $\mathcal{K}_{3-i}'$, which does not contain any negative occurrences of the symbols of $\mathcal{K}_i'$, and $\mathcal{U}_i$ is a model of $\mathcal{K}_i'$; therefore, $\mathcal{U}$ is a model of $\mathcal{K}_i'$. For (ii), consider a finite subinterpretation $\mathcal{U}_0$ of $\mathcal{U}$ and, for $i = 1, 2$, let $\mathcal{U}_{0i}$ be the respective finite subinterpretation of $\mathcal{U}_i$. Since $\mathcal{K}_1$ Σ-query entails both $\mathcal{K}_1'$ and $\mathcal{K}_2'$, by Theorem 5, we have $(\Sigma, \mathsf{ind}(\mathcal{K}_i'))$-homomorphisms h_i from $\mathcal{U}_{0i}$ to $\mathcal{M}_1$, for $i = 1, 2$. Define h by taking $h(u) = h_1(u)$, for all $u \in \Delta^{\mathcal{U}_{01}}$, and $h(u) = h_2(u)$, for all $u \in \Delta^{\mathcal{U}_{02}} \setminus \Delta^{\mathcal{U}_{01}}$. Since (A.1) and $h_1(a) = h_2(a)$, for all $a \in \mathsf{part}_\Sigma^{\mathcal{U}_1} \cap \mathsf{part}_\Sigma^{\mathcal{U}_2}$, the function h is a $(\Sigma, \mathsf{ind}(\mathcal{K}_1) \cup \mathsf{ind}(\mathcal{K}_2))$-homomorphism from $\mathcal{U}_0$ to $\mathcal{M}_1$, as required.

Case 1.2: Suppose that the trivial interpretation is not a model of $\mathcal{T}_i$, for some $i \in \{1, 2\}$. We construct $\mathcal{K}_i'' = (\mathcal{T}_i'', \mathcal{A}_i'')$, $i = 1, 2$, such that the trivial interpretation is a model of $\mathcal{T}_i''$, for $i = 1, 2$, and $\mathcal{K}_1$ Σ-query entails $\mathcal{K}_2$ iff $\mathcal{K}_1''$ Σ-query entails $\mathcal{K}_2''$ (this will reduce Case 1.2 to Case 1.1). The construction is by careful relativisation. We assume that the TBoxes $\mathcal{T}_i$ are in normal form (see Theorem 7). If the $\mathcal{T}_i$ do not contain inclusions of the form $\top \sqsubseteq A$ then the trivial interpretation is a model of the TBoxes and we are done. Otherwise, for $i = 1, 2$, let D^i be fresh concept names: D^i will replace $\top$ in the inclusion $\top \sqsubseteq A$ in $\mathcal{T}_i$, which will ensure that the trivial interpretation is a model of the resulting TBox. In addition, we have to ensure that D^i contains all domain elements of the materialisation. To deal with the individual names in the ABox $\mathcal{A}_i$, we take $\mathcal{A}_i'' = \mathcal{A}_i \cup \mathcal{A}_i^{D_i}$, where

$$\mathcal{A}_i^{D_i} = \{ D^i(a) \mid a \in \mathsf{ind}(\mathcal{K}_i) \}. \tag{A.2}$$

The TBoxes $\mathcal{T}_i''$ are obtained from $\mathcal{T}_i$ by replacing

- any inclusion $\top \sqsubseteq A$ with $D^i \sqsubseteq A$;
- any inclusion $A \sqsubseteq \exists R.C$ with
  - $A \sqsubseteq \exists R$ and $\exists R^- \sqsubseteq D^i$, if the $\mathcal{T}_i$ are members of the *DL-Lite* family ($C = \top$ in this case), and

– $A \sqsubseteq \exists R.(D^i \sqcap C)$, otherwise.

The remaining inclusions are not modified and the modification of inclusions of the form $A \sqsubseteq \exists R.C$ ensures that D^i holds in all generated domain elements of the materialisations constructed to prove Theorem 11. Note that if $\mathcal{T}_i$ is an $\mathcal{L}$-TBox, then $\mathcal{T}_i''$ is an $\mathcal{L}$-TBox as well, for any of our DLs. We show that the $\mathcal{K}_i'' = (\mathcal{T}_i'', \mathcal{A}_i'')$, for $i = 1, 2$, are as required. First, by construction, the trivial interpretation $\mathcal{I}_\emptyset$ is a model of $\mathcal{T}_i''$. Second, let $\mathcal{M}_i$ be the unravelling of a generating structure for $\mathcal{K}_i$. By Theorem 8, $\mathcal{M}_i$ is a materialisation of $\mathcal{K}_i$. Observe that the interpretation $\mathcal{U}_i$ obtained from $\mathcal{M}_i$ by interpreting D^i as the domain of $\mathcal{M}_i$ is a materialisation of $\mathcal{K}_i''$. Thus, by Theorem 5, $\mathcal{K}_1$ Σ-query entails $\mathcal{K}_2$ iff $\mathcal{K}_1''$ Σ-query entails $\mathcal{K}_2''$, as required.

Case 2: $\mathcal{L}$ contains $\mathcal{EL}$ and has no role inclusions (that is, $\mathcal{L} \in \{\mathcal{EL}, \mathcal{ALC}, \mathcal{ALCI}\}$). We construct $\mathcal{K}_1' = (\mathcal{T}_1', \mathcal{A}_1')$ and $\mathcal{K}_2' = (\mathcal{T}_2', \mathcal{A}_2')$ such that

$$\mathcal{K}_1 \text{ } \Sigma\text{-query entails } \mathcal{K}_2 \quad \text{iff} \quad \mathcal{K}_1 \text{ and } \mathcal{K}_1' \cup \mathcal{K}_2' \text{ are } \Sigma\text{-query inseparable.} \tag{A.3}$$

We employ relativisation again. Let D^i be fresh concept names, for $i = 1, 2$. In this case, apart from ensuring that D^i contains all domain elements of the materialisation of $\mathcal{K}_i$ we have to ensure that merging the materialisations of $\mathcal{K}_1$ and $\mathcal{K}_2$ does not lead to additional domain elements. Let $\mathcal{A}_i' = \mathcal{A}_i \cup \mathcal{A}_i^{D_i}$, for $i = 1, 2$, where $\mathcal{A}_i^{D_i}$ is defined by (A.2). Assume the TBoxes $\mathcal{T}_i$ are in normal form and define $\mathcal{T}_i'$ by replacing

– any inclusion $\top \sqsubseteq A$ with $D^i \sqsubseteq A$;

– any inclusion $A_1 \sqsubseteq A_2$ with $A_1 \sqcap D^i \sqsubseteq A_2$;

– any inclusion $A_1 \sqcap A_2 \sqsubseteq A$ with $A_1 \sqcap A_2 \sqcap D^i \sqsubseteq A$;

– any inclusion $\exists R.C \sqsubseteq A$ with $\exists R.(C \sqcap D^i) \sqcap D^i \sqsubseteq A$;

– any inclusion $A \sqsubseteq \exists R.C$ with $A \sqcap D^i \sqsubseteq \exists R.(D^i \sqcap C)$;

– any inclusion $A_1 \sqsubseteq \forall R.A_2$ with $A_1 \sqcap D^i \sqsubseteq \forall R.(\neg D^i \sqcup A_2)$.

Note that $\mathcal{T}_i'$ is not necessarily in normal form, but it is an $\mathcal{L}$-TBox, which can then be transformed to normal form by Theorem 7.

We show (A.3). The interesting direction is 'if $\mathcal{K}_1$ Σ-query entails $\mathcal{K}_2$ then $\mathcal{K}_1$ Σ-query entails $\mathcal{K}_1' \cup \mathcal{K}_2'$'. Suppose $\mathcal{K}_1$ Σ-query entails $\mathcal{K}_2$. Then $\mathcal{K}_1$ Σ-query entails both $\mathcal{K}_1'$ and $\mathcal{K}_2'$ (as $\mathcal{K}_1$ Σ-query entails both $\mathcal{K}_1$ and $\mathcal{K}_2$). Let $\mathcal{M}_1$ be a materialisation of $\mathcal{K}_1$ and, for $i = 1, 2$, let $\mathcal{U}_i$ be a materialisation of $\mathcal{K}_i'$ obtained by unravelling a generating structure for $\mathcal{K}_i'$. We proceed as in Case 1.1: we construct $\mathcal{U}$ by merging $\mathcal{U}_1$ and $\mathcal{U}_2$ and show that conditions (i) and (ii) hold. It will then follow that $\mathcal{K}_1$ Σ-query entails $\mathcal{K}_1' \cup \mathcal{K}_2'$.

For item (i), observe that, for $i = 1, 2$, the trivial interpretation $\mathcal{I}_\emptyset$ is a model of $\mathcal{T}_i'$ and every inclusion of $\mathcal{T}_i'$ is relativised to D^i: it is 'applicable' only to elements in D_i and 'generates' only elements in D_i again. Thus, $\mathcal{U}$ is a model of $\mathcal{K}_1' \cup \mathcal{K}_2'$. The argument for item (ii) is analogous to Case 1.1, which completes the proof. ❑

Theorem 42. *Σ-query entailment for any of our DLs $\mathcal{L}$ is* LogSpace*-reducible to the membership problem for universal CQ-solutions in $\mathcal{L}$.*

Proof. The use the proof of Theorem 12. Suppose $\mathcal{L}$ KBs $\mathcal{K}_1, \mathcal{K}_2$, and a signature Σ are given. We want to reduce the problem to decide whether $\mathcal{K}_1$ Σ-query entails $\mathcal{K}_2$ to the membership problem for universal CQ-solutions in $\mathcal{L}$. As argued in the proof of Theorem 12, we may assume that $\Sigma = \mathsf{sig}(\mathcal{K}_1) = \mathsf{sig}(\mathcal{K}_1) \cap \mathsf{sig}(\mathcal{K}_2)$.

For the reduction to the membership problem for universal CQ-solutions in $\mathcal{L}$, we do not have to consider the case that $\mathcal{L}$ does not have role inclusions since they can always be used in the mapping $\mathcal{T}_{12}$. Thus, we follow the proof of Case 1 in the in the proof of Theorem 12 and first assume that the trivial interpretation $\mathcal{I}_\emptyset$ is a model of $\mathcal{T}_i$, for $i = 1, 2$. Recall the definition of $\mathcal{K}_i^i$: $\mathcal{K}_i^i$ is obtained from $\mathcal{K}_i$ by replacing every symbol S in $\mathcal{K}_i$ with a fresh symbol S_i. Then it is shown in the proof of Theorem 12 (Case 1.1) that $\mathcal{K}_1$ Σ-query entails $\mathcal{K}_2$ iff $\mathcal{K}_1' \cup \mathcal{K}_2' \cup \mathcal{T}_{12}$ and $\mathcal{K}_1$ are Σ-query inseparable, where $\mathcal{T}_{12} = \{S_i \sqsubseteq S \mid S \in \Sigma\}$. But the latter problem is a membership problem for universal CQ-solutions since we assume that $\Sigma = \mathsf{sig}(\mathcal{K}_1)$.

We complete the proof by considering the case when $\mathcal{I}_\emptyset$ is not a model of $\mathcal{T}_i$ for some $i \in \{1, 2\}$. We reduce this case to the previous one by constructing KBs $\mathcal{K}_i'' = (\mathcal{T}_i'', \mathcal{A}_i'')$ such that $\mathcal{I}_\emptyset$ is a model of $\mathcal{T}_i''$ and $\mathcal{K}_1$ Σ-query entails $\mathcal{K}_2$ iff $\mathcal{K}_1''$ Σ-query entails $\mathcal{K}_2''$. But KBs $\mathcal{K}_i''$ with these properties have been constructed in the proof of Theorem 12 (Case 1.2) already (observe that no role inclusions are introduced in the construction of $\mathcal{K}_i''$, and so $\mathcal{K}_i''$ is an $\mathcal{L}$-KB if $\mathcal{K}_i$ is an $\mathcal{L}$-KB for any of our DLs $\mathcal{L}$). ❑