Reverse Engineering of Temporal Queries with and without LTL Ontologies: First Steps

Marie Fortin[1], Boris Konev[1], Vladislav Ryzhikov[2], Yury Savateev[2], Frank Wolter[1] and Michael Zakharyaschev[2]

[1]*Department of Computer Science, University of Liverpool, UK*

[2]*Department of Computer Science and Information Systems, Birkbeck, University of London, UK*

Abstract

In reverse engineering of database queries, one aims to construct a query from a set of positively and negatively labelled answers and non-answers. The query can then be used to explore the data further or as an explanation of the answers and non-answers. We consider this reverse engineering problem for queries formulated in various fragments of positive linear temporal logic *LTL* over data instances given by timestamped atomic concepts. We focus on the design of suitable query languages and the complexity of the separability problem: 'does there exist a query in the given query language that separates the given answers from the non-answers?'. We deal with both plain *LTL* queries and those that are mediated by ontologies providing background knowledge and formulated in fragments of clausal *LTL*.

Keywords

Reverse engineering of queries, query-by-example, explanation, linear temporal logic, ontology-mediated query, computational complexity.

1. Introduction

Supporting users of databases by constructing a query from examples of answers and non-answers to the query has been a major research area for many years [1]. In the database community, research has focussed on standard query languages such as (fragments of) SQL, graph query languages, and SPARQL [2, 3, 4, 5, 6, 7, 8, 9]. The KR community has been concerned with constructing queries from examples under the open world semantics and with background knowledge given by an ontology [10, 11, 12, 13, 14]. In both cases, the focus has been on general multi-purpose query languages. A fundamental problem that has been investigated by both communities is known as *separability* or *query-by-example*: given sets E^+ and E^- of pairs $(\mathcal{D}, \boldsymbol{d})$ with a database $\mathcal{D}$ and a tuple $\boldsymbol{d}$ in $\mathcal{D}$, and a query language $\mathcal{Q}$, does there exist a query $\boldsymbol{q} \in \mathcal{Q}$ that separates (E^+, E^-) is the sense that $\mathcal{D} \models \boldsymbol{q}(\boldsymbol{d})$ for all $(\mathcal{D}, \boldsymbol{d}) \in E^+$ and $\mathcal{D} \not\models \boldsymbol{q}(\boldsymbol{d})$ for all $(\mathcal{D}, \boldsymbol{d}) \in E^-$ (or $\mathcal{O}, \mathcal{D} \models \boldsymbol{q}(\boldsymbol{d})$ for all $(\mathcal{D}, \boldsymbol{d}) \in E^+$ and $\mathcal{O}, \mathcal{D} \not\models \boldsymbol{q}(\boldsymbol{d})$ for all $(\mathcal{D}, \boldsymbol{d}) \in E^-$ if an ontology $\mathcal{O}$ is present)?[1] There are various strategies to ensure that the query $\boldsymbol{q}$ is a generalisation of the positive examples and does not overfit. For instance, one can ask for

DL 2022: 35th International Workshop on Description Logics, August 7–10, 2022, Haifa, Israel

marie.fortin@liverpool.ac.uk (M. Fortin); boris.konev@liverpool.ac.uk (B. Konev); vlad@dcs.bbk.ac.uk (V. Ryzhikov); yury@dcs.bbk.ac.uk (Y. Savateev); wolter@liverpool.ac.uk (F. Wolter); michael@dcs.bbk.ac.uk (M. Zakharyaschev)

CEUR Workshop Proceedings (CEUR-WS.org)

[1] If such a $\boldsymbol{q}$ exists, then (E^+, E^-) is often called *satisfiable* w.r.t. $\mathcal{Q}$ and the construction of $\boldsymbol{q}$ is called *learning*.

the existence of a small separating query in $\mathcal{Q}$ or one can choose a query language that enforces generalisation by not admitting disjunction. In the latter case, query-by-example is often very hard computationally: it is coNExpTime-complete for conjunctive queries (CQs) over standard relational databases [15, 16] and even undecidable for CQs under $\mathcal{ELI}$ or $\mathcal{ALC}$ ontologies [17].

In many applications, the input data is timestamped and queries are naturally formulated in languages with temporal operators. Taking into account the prohibitive complexity of many query-by-example problems already in the static case, it does not seem wise to start an investigation of the temporal case by considering temporal extensions of standard query languages (which can only lead to computationally even harder problems). Instead, we investigate the simpler but still very useful case in which data, $\mathcal{D}$, is a set of timestamped atomic concepts. Our query languages are positive fragments of linear temporal logic *LTL* with the temporal operators $\Diamond$ (eventually), $\bigcirc$ (next), and U (until) interpreted under the strict semantics [18]. To avoid overfitting, we only consider such fragments without $\vee$. The most expressive query language we deal with, $\mathcal{Q}[\mathsf{U}]$, is thus defined as the set of formulas constructed from atoms using $\wedge$ and U (via which $\bigcirc$ and $\Diamond$ can be defined). The fragments $\mathcal{Q}[\Diamond]$, $\mathcal{Q}[\bigcirc]$, and $\mathcal{Q}[\bigcirc, \Diamond]$ are defined analogously.

Within this temporal setting, we take a broad view of the potential applications of the reverse engineering of queries and the query-by-example problem. On the one hand, there are non-expert end-users who would like to explore data via queries but are not familiar with temporal logic. They usually are, however, capable of providing data examples illustrating the queries they are after. Query-by-example supports such users in the construction of those queries. On the other hand, the positive and negative data examples might come from an application, and the user is interested in possible explanations of the examples. Such an explanation is then provided by a temporal query separating the positive examples from the negative ones. In this case, our goal is similar to recent work on learning linear temporal logic formulas and, more generally, explainable AI [19, 20, 21, 22, 23]. The following example illustrates this point.

Example 1. Imagine an engineer whose task is to explain the behaviour of the monitored equipment (say, why an engine stops) in terms of qualitative sensor data such as 'low temperature', which can be represented by the atomic concept T, 'strong vibration', V, etc. Suppose the engine stopped after the runs $\mathcal{D}_1^+$ and $\mathcal{D}_2^+$ shown below but did not stop after the runs $\mathcal{D}_1^-$, $\mathcal{D}_2^-$, $\mathcal{D}_3^-$, where we assume the runs to start at 0 and measurements to be recorded at $0, 1, 2, \dots$:

$$\mathcal{D}_1^+ = \{T(2), V(4)\}, \mathcal{D}_2^+ = \{T(1), V(4)\}, \mathcal{D}_1^- = \{T(1)\}, \mathcal{D}_2^- = \{V(4)\}, \mathcal{D}_3^- = \{V(1), T(2)\}.$$

The $\Diamond$-query $\boldsymbol{q} = \Diamond(T \wedge \Diamond\Diamond V)$ is true at 0 in the positive data instances $\mathcal{D}_i^+$, false in the negative ones $\mathcal{D}_i^-$, and so provides a possible natural explanation of what could cause the engine failure. The example set $(\{\mathcal{D}_3^+, \mathcal{D}_4^+\}, \{\mathcal{D}_4^-\})$ with

$$\mathcal{D}_3^+ = \{T(1), V(2)\}, \;\; \mathcal{D}_4^+ = \{T(1), T(2), V(3)\}, \;\; \mathcal{D}_4^- = \{T(1), V(3)\}$$

can be explained by the U-query $T \,\mathsf{U}\, V$. Using background knowledge of the domain, we can compensate for sensor failures, which result in incomplete data. To illustrate, suppose that $\bar{\mathcal{D}}_1^+ = \{H(3), V(4)\}$, where H stands for 'heater is on'. If a background ontology $\mathcal{O}$ contains the axiom $\bigcirc H \to T$ saying that a heater can only be triggered by the low temperature at the previous moment, then the same $\boldsymbol{q}$ will separate $\{\bar{\mathcal{D}}_1^+, \mathcal{D}_2^+\}$ from $\{\mathcal{D}_1^-, \mathcal{D}_2^-, \mathcal{D}_3^-\}$ under $\mathcal{O}$. ⊣

The queries used in Example 1 are of a particular 'linear' form and suggest a restriction to *path queries* in which the order of the atoms is fixed and not left open as in $\Diamond A \land \Diamond B$. More precisely, *path* $\bigcirc\Diamond$*-queries* in the class $\mathcal{Q}_p[\bigcirc, \Diamond]$ take the form

$$\boldsymbol{q} = \rho_0 \land \boldsymbol{o}_1(\rho_1 \land \boldsymbol{o}_2(\rho_2 \land \dots \land \boldsymbol{o}_n\rho_n)), \tag{1}$$

where $\boldsymbol{o}_i \in \{\bigcirc, \Diamond\}$ and ρ_i is a conjunction of atoms; $\mathcal{Q}_p[\Diamond]$ and $\mathcal{Q}_p[\bigcirc]$ restrict $\boldsymbol{o}_i$ to $\{\Diamond\}$ and $\{\bigcirc\}$, respectively; and *path* U*-queries* in the class $\mathcal{Q}_p[\mathsf{U}]$ take the form

$$\boldsymbol{q} = \rho_0 \land (\lambda_1 \mathbin{\mathsf{U}} (\rho_1 \land (\lambda_2 \mathbin{\mathsf{U}} (\dots (\lambda_n \mathbin{\mathsf{U}} \rho_n) \dots)))), \tag{2}$$

where λ_i is a conjunction of atoms or $\bot$. Path queries are motivated by two observations. First, if a query language $\mathcal{Q}$ allows conjunctions of queries, then, dually to the overfitting problem for disjunction, the admission of multiple negative examples becomes trivialised: if queries $\boldsymbol{q}_{\mathcal{D}}$ separate $(E^+, \{\mathcal{D}\})$ for $\mathcal{D} \in E^-$, then the conjunction $\bigwedge_{\mathcal{D}\in E^-} \boldsymbol{q}_{\mathcal{D}}$ separates (E^+, E^-). In particular, the query-by-example problem becomes polynomially reducible to its version with a single negative example. This is clearly not the case for path queries.

Example 2. Let $\mathcal{D}_1 = \{A(1), B(2)\}$, $\mathcal{D}_2 = \{B(1), A(2)\}$, $\mathcal{D}_3 = \{A(1)\}$ and $\mathcal{D}_4 = \{B(1)\}$. Then $(\{\mathcal{D}_1, \mathcal{D}_2\}, \mathcal{D}_3)$ and $(\{\mathcal{D}_1, \mathcal{D}_2\}, \mathcal{D}_4)$ are separated in $\mathcal{Q}_p[\Diamond]$ by $\Diamond B$ and $\Diamond A$, respectively; $(\{\mathcal{D}_1, \mathcal{D}_2\}, \{\mathcal{D}_3, \mathcal{D}_4\})$ is separated in $\mathcal{Q}[\Diamond]$ by $\Diamond B \land \Diamond A$, but it is not $\mathcal{Q}_p[\Diamond]$-separable. ⊣

Second, numerous natural types of query classes from applications are represented by path queries. For example, the existence of a *common subsequence* of the positive examples that is not a subsequence of any negative example corresponds to the existence of a separating query in $\mathcal{Q}_p[\Diamond]$ with $\rho_0 = \top$ and $\rho_i \neq \top$ for $i > 0$, and the existence of a *common subword* of the positive examples that is not a *subword* of any negative example corresponds to the existence of a separating query of the form $\Diamond(\rho_1 \land \bigcirc(\rho_2 \land \dots \land \bigcirc\rho_n))$. The unique characterisability and learnability of path queries is investigated in [24].

Except for $\mathcal{Q}_p[\bigcirc] = \mathcal{Q}[\bigcirc]$ (modulo logical equivalence), no nontrivial inclusion relations hold between the separation capabilities of the query languages introduced above, as illustrated by the following example.

Example 3. (1) Let $\mathcal{D}_1 = \{A(1)\}$, $\mathcal{D}_2 = \{A(2)\}$ and $E = (\{\mathcal{D}_1\}, \{\mathcal{D}_2\})$. Then $\bigcirc A$ separates E but no query in $\mathcal{Q}[\Diamond]$ does. On the other hand, E is not $\mathcal{Q}$-separable under $\mathcal{O} = \{\bigcirc A \to A\}$, for any class $\mathcal{Q}$ defined above, as $\mathcal{O}, \mathcal{D}_1 \models \boldsymbol{q}(0)$ implies $\mathcal{O}, \mathcal{D}_2 \models \boldsymbol{q}(0)$ for all $\boldsymbol{q} \in \mathcal{Q}$.

(2) Let $E = (\{\mathcal{D}_1, \mathcal{D}_2\}, \emptyset)$ with $\mathcal{D}_1$ and $\mathcal{D}_2$ as in (1). Then $\Diamond A$ separates E but no $\mathcal{Q}[\bigcirc]$-query does. Observe that at least two positive examples are needed to achieve this effect. However, $\bigcirc\bigcirc\bigcirc B$ separates E under $\mathcal{O} = \{A \to \Box B\}$.

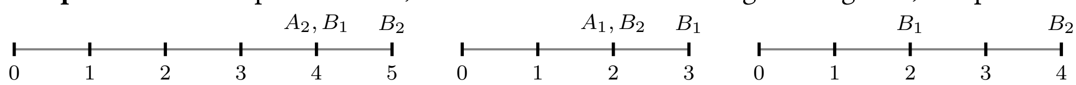
(3) Let $E = (\{\{A(1), B(2)\}, \{A(2), B(3)\}\}, \{\{A(3), B(5)\}\})$. Then $\Diamond(A \land \bigcirc B)$ separates E but no query in $\mathcal{Q}[\bigcirc]$ or $\mathcal{Q}[\Diamond]$ does.

(4) $A \mathbin{\mathsf{U}} B$ separates $(\{\{B(1)\}, \{A(1), B(2)\}\}, \{\{B(2)\}\})$ but no $\mathcal{Q}[\bigcirc, \Diamond]$-query does. ⊣

Our contribution. We now briefly present our initial results on the complexity of the separability problem for *LTL* queries, both plain and mediated by an *LTL*-ontology.

Ontology-free LTL queries. Separability in $\mathcal{Q}[\bigcirc]$ is almost trivial as it corresponds to the existence of a time point where some atom holds in all positive examples but in no negative example, which is decidable in polynomial time. For the query languages ranging from $\mathcal{Q}_p[\Diamond]$ and $\mathcal{Q}[\Diamond]$ to $\mathcal{Q}_p[\bigcirc, \Diamond]$ and $\mathcal{Q}[\bigcirc, \Diamond]$ and also $\mathcal{Q}_p[\mathsf{U}]$, separability turns out to be NP-complete. The upper bound is proved by observing that, in any of these languages, every separable example set can be separated by a query of polynomial size. The matching lower bound is established by reduction of the NP-hard problem of deciding whether the words in a given set contain a common subsequence of a given length [25]. Separability by $\mathcal{Q}[\mathsf{U}]$-queries turns out to be trickier because of the interplay of $\wedge$, the left- and the right-hand sides of the U-operator.

Example 4. The example set below, where the instance on the right is negative, is separated by



the $\mathcal{Q}[\mathsf{U}]$-query $\Diamond\big((A_1 \mathbin{\mathsf{U}} B_1) \wedge (A_2 \mathbin{\mathsf{U}} B_2)\big)$ but is not separable in any other class of queries.

We give a separability criterion in terms of U-simulations between subsets of the disjoint union of the positive examples and points of a negative example (cf. [26]). Then, using a game-theoretic variant of U-simulations, we show that a separating $\mathcal{Q}[\mathsf{U}]$-query can be constructed in PSpace. However, at the moment, we only have an NP-lower bound for separability.

Separability under LTL Ontologies. Apart from full *LTL*, we consider its fragment $\mathit{LTL}^{\Box\Diamond}$ that only uses the operators $\Box$ and $\Diamond$, also known as the *Prior logic* [27, 28, 29, 30], and the Horn fragment $\mathit{LTL}^{\Box\bigcirc}_{\mathit{horn}}$ containing axioms of the form $C_1 \wedge \cdots \wedge C_k \to C_{k+1}$, where the C_i are atoms possibly prefixed by $\Box$ and $\bigcirc$ for $i \leq k+1$, and also by $\Diamond$ for $i \leq k$. The Ontology axioms are supposed to hold at all times.

Separability by (path) $\Diamond$-queries is Σ^p_2-complete under $\mathit{LTL}^{\Box\Diamond}$ ontologies and PSpace-complete under $\mathit{LTL}^{\Box\bigcirc}_{\mathit{horn}}$ ontologies. For *LTL* ontologies, we have a NExpTime upper bound. We conjecture that exactly the same bounds can be proved for (path) $\bigcirc\Diamond$-queries. As concerns $\mathcal{Q}_p[\mathsf{U}]$-queries, separability under $\mathit{LTL}^{\Box\bigcirc}_{\mathit{horn}}$ ontologies is shown to be between ExpSpace and NExpTime; for 'branching' $\mathcal{Q}[\mathsf{U}^-]$-queries without nesting U-operators on the left of U, it can be decided in ExpTime using U-simulations. We establish the upper bounds by constructing two exponential-size transition systems S^+ and S^- from $(\mathcal{O}, E^+)$ and $(\mathcal{O}, E^-)$ such that (i) there is a *trace-based simulation* of S^+ by S^- iff (E^+, E^-) is separated in $\mathcal{Q}_p[\mathsf{U}]$ and (ii) there is a *tree-based simulation* of S^+ by S^- iff (E^+, E^-) is separated in $\mathcal{Q}[\mathsf{U}^-]$. The existence of trace-based and tree-based simulations can be decided in PSpace- and P, respectively [31].

QBE($\mathcal{L}, \mathcal{Q}$)	*LTL*	$\mathit{LTL}^{\Box\bigcirc}_{\mathit{horn}}$	$\mathit{LTL}^{\Box\Diamond}$	QBE($\mathcal{Q}$)
$\mathcal{Q}[\mathsf{U}]$		?		$\geq$ NP, $\leq$ PSpace
$\mathcal{Q}[\mathsf{U}^-]$		$\leq$ ExpTime		
$\mathcal{Q}_p[\mathsf{U}]$	?	$\geq$ NExpTime, $\leq$ ExpSpace	?	
$\mathcal{Q}[\bigcirc, \Diamond]$		$\leq$ ExpTime		
$\mathcal{Q}_p[\bigcirc, \Diamond]$		$\leq$ ExpSpace		= NP
$\mathcal{Q}[\Diamond]$	$\leq$ NExpTime	= PSpace	$= \Sigma^p_2$	
$\mathcal{Q}_p[\Diamond]$				

Acknowledgments

This research was supported by the EPSRC UK grants EP/S032207 and EP/S032282 for the joint project 'quant$^{\text{MD}}$: Ontology-Based Management for Many-Dimensional Quantitative Data'.

Appendix

2. Related Work in Concept Learning

We discuss related work in concept learning in description logic, as first proposed in [32]. Inspired by inductive logic programming, refinement operators are used to construct a concept that separates positive from negative examples in an ABox. An ontology may or may not be present. There has been significant interest in this approach [33, 34, 35, 36, 37, 38, 39]. Prominent systems include the DL Learner [40, 41], DL-Foil [42] and its extension DL-Focl [43], SPaCEL [44], YinYang [45], and pFOIL-DL [46].

3. Preliminaries

Temporal ontology-mediated queries. In our setting, the alphabet of linear temporal logic *LTL* comprises a set of *atomic concepts* A_i, $i < \omega$ (or simply *atoms*, for short). *Basic temporal concepts*, C, are defined by the grammar

$$C \quad ::= \quad A_i \quad | \quad \Box C \quad | \quad \Diamond C \quad | \quad \bigcirc C$$

with the *temporal operators* $\Box$ (always in the future), $\Diamond$ (sometime in the future) and $\bigcirc$ (at the next moment). An *LTL ontology*, $\mathcal{O}$, is a finite set of *axioms* of the form

$$C_1 \land \dots \land C_k \ \to \ C_{k+1} \lor \dots \lor C_{k+m}, \tag{3}$$

where $k, m \geq 0$, the C_i are basic temporal concepts, the empty $\land$ is $\top$, and the empty $\lor$ is $\bot$.

In this paper, we consider four types of *LTL* ontologies, adopting the nomenclature from [47] where both future- and past-time temporal operators were allowed:

$\boldsymbol{LTL^{\Box\bigcirc}_{bool}}$ admits arbitrary axioms of the form (3); $\Diamond$ is not mentioned in the fragment's name because it can be expressed via $\Box$ and $\bigcirc$. Every *LTL* ontology given by arbitrary (non-clausal) formulas of the form $\varphi \land \Box\varphi$ can be efficiently converted to an $LTL^{\Box\bigcirc}_{bool}$ ontology [47].

$\boldsymbol{LTL^{\Box}_{bool}}$ admits axioms of the form (3) in which the C_i do not contain $\bigcirc$ (with $\Diamond$ being expressible via $\Box$). This fragment is equivalent to the $\Box\Diamond$-fragment of non-clausal *LTL*, known as the *Prior logic* [27, 28, 29, 30].

$\boldsymbol{LTL^{\Box\bigcirc}_{horn}}$ admits axioms of the form (3) in which $m \leq 1$ and $\Diamond$ does not occur in C_{k+1} ($\Diamond$ on the left-hand side of (3) is expressible via $\bigcirc$).

$LTL^{\Box}_{horn}$ admits axioms of the form (3) in which $m \leq 1$ and $\Diamond$ does not occur in C_{k+1} (in this case $\Diamond$ on the left-hand side of (3) is not expressible via $\Box$ but can be expressed using the past-time counterpart $\Box_P$ of $\Box$).

A *data instance* is a finite set $\mathcal{D}$ of atoms of the form $A_i(\ell)$ with a *timestamp* $\ell \in \mathbb{N}$ together with an interval *tem*$(\mathcal{D}) = [m, n] \subseteq \mathbb{N}$, called the *active domain* of $\mathcal{D}$, such that $m \leq \ell \leq n$, for all $A_i(\ell) \in \mathcal{D}$. If $\mathcal{D} = \emptyset$, then *tem*$(\mathcal{D})$ may also be $\emptyset$. Otherwise, we assume (without loss of generality) that $m = 0$. If *tem*$(\mathcal{D})$ is not specified explicitly, it is assumed to be either empty or $[0, n]$, where n is the maximal timestamp in $\mathcal{D}$. By a *signature*, Σ, we mean any finite set of atomic concepts. An instance $\mathcal{D}$ is said to be a Σ-*data instance* if $A_i(\ell) \in \mathcal{D}$ implies $A_i \in \Sigma$.

We access data by means of (*LTL* analogues of conjunctive) *queries*, $\varkappa$, constructed from atoms, $\bot$ and $\top$ using $\wedge$ and the temporal operators $\bigcirc$, $\Diamond$ and U. The set of atoms occurring in $\varkappa$ is denoted by $\mathsf{sig}(\varkappa)$. The set of queries that only use the operators from $\Phi \subseteq \{\bigcirc, \Diamond, \mathsf{U}\}$ is denoted by $\mathcal{Q}[\Phi]$; $\mathcal{Q}^{\Sigma}[\Phi]$ is the restriction of $\mathcal{Q}[\Phi]$ to a signature Σ. We also consider the subclass $\mathcal{Q}[\mathsf{U}^-]$ of $\mathcal{Q}[\mathsf{U}]$ comprising those queries that do not contain subqueries of the form $\varkappa_1 \mathbin{\mathsf{U}} \varkappa_2$ with an occurrence of U in $\varkappa_1$. The *size* $|\varkappa|$ of $\varkappa$ is the number of symbols in $\varkappa$, and the *temporal depth* *tdp*$(\varkappa)$ of $\varkappa$ is the maximum number of nested temporal operators in $\varkappa$.

Note that under the strict semantics to be defined below, $\mathcal{Q}[\bigcirc, \Diamond]$-queries can be equivalently given as *tree-shaped* conjunctive queries (CQs) with the binary predicates *suc* and $<$ over $\mathbb{N}$, and atomic concepts as unary predicates. Each such CQ is a set $Q(t_0)$ of assertions of the form $A(t)$, *suc*(t, t'), and $t < t'$, with a distinguished variable t_0, such that, for every variable t in $Q(t_0)$, there exists exactly one path from t_0 to t along the binary predicates *suc* and $<$. The set of $\mathcal{Q}[\bigcirc, \Diamond]$-queries with *path-shaped* CQ counterparts is denoted by $\mathcal{Q}_p[\bigcirc, \Diamond]$. Such queries $\varkappa$ take the form (4), where $\boldsymbol{o}_i \in \{\bigcirc, \Diamond\}$ and ρ_i is a conjunction of atoms; similarly, path-shaped $\mathcal{Q}_p[\mathsf{U}]$-queries are of the form (5):

$$\varkappa = \rho_0 \wedge \boldsymbol{o}_1(\rho_1 \wedge \boldsymbol{o}_2(\rho_2 \wedge \dots \wedge \boldsymbol{o}_n \rho_n)), \tag{4}$$

$$\varkappa = \rho_0 \wedge (\lambda_1 \mathbin{\mathsf{U}} (\rho_1 \wedge (\lambda_2 \mathbin{\mathsf{U}} (\dots (\lambda_n \mathbin{\mathsf{U}} \rho_n) \dots)))). \tag{5}$$

Note that $\mathcal{Q}_p[\mathsf{U}] \subseteq \mathcal{Q}[\mathsf{U}^-]$.

The intended *interpretations* are structures $\mathcal{I} = (\mathbb{N}, A_0^{\mathcal{I}}, A_1^{\mathcal{I}}, \dots)$ with $A_i^{\mathcal{I}} \subseteq \mathbb{N}$, for every $i < \omega$. The *extension* $\varkappa^{\mathcal{I}}$ of a temporal concept $\varkappa$ in $\mathcal{I}$ is defined inductively as usual in *LTL* under the strict semantics [48, 18]:

$$(\bigcirc\varkappa)^{\mathcal{I}} = \{\, n \in \mathbb{Z} \mid n + 1 \in \varkappa^{\mathcal{I}} \,\},$$
$$(\Box\varkappa)^{\mathcal{I}} = \{\, n \in \mathbb{Z} \mid k \in \varkappa^{\mathcal{I}}, \text{ for all } k > n \,\},$$
$$(\Diamond\varkappa)^{\mathcal{I}} = \{\, n \in \mathbb{Z} \mid \text{there is } k > n \text{ with } k \in \varkappa^{\mathcal{I}} \,\},$$
$$(\varkappa_1 \mathbin{\mathsf{U}} \varkappa_2)^{\mathcal{I}} = \{\, n \in \mathbb{Z} \mid \text{there is } k > n \text{ with } k \in \varkappa_2^{\mathcal{I}} \text{ and } m \in \varkappa_1^{\mathcal{I}} \text{ for } n < m < k \,\}.$$

We regard $\mathcal{I}, n \models \varkappa$ as synonymous to $n \in \varkappa^{\mathcal{I}}$. We say that an axiom (3) is *true* in $\mathcal{I}$ if $C_1^{\mathcal{I}} \cap \dots \cap C_k^{\mathcal{I}} \subseteq C_{k+1}^{\mathcal{I}} \cup \dots \cup C_{k+m}^{\mathcal{I}}$, that is, if it is true at all times. An interpretation $\mathcal{I}$ is a *model* of $\mathcal{O}$ if all axioms of $\mathcal{O}$ are true in $\mathcal{I}$; it is a *model* of $\mathcal{D}$ if $A_i(\ell) \in \mathcal{D}$ implies $\ell \in A_i^{\mathcal{I}}$. $\mathcal{O}$ and $\mathcal{D}$ are *consistent* if they have a model. We write $\mathcal{O}, \mathcal{D} \models \varkappa(k)$, for $k \in \mathbb{N}$, if $k \in \varkappa^{\mathcal{I}}$ in all models $\mathcal{I}$ of $\mathcal{O}$ and $\mathcal{A}$. We say that $\varkappa$ $\mathcal{O}$-*entails* $\varkappa'$ and write $\varkappa \models_{\mathcal{O}} \varkappa'$ if $\mathcal{I}, 0 \models \varkappa$ implies

$\mathcal{I}, 0 \models \varkappa'$ for all models $\mathcal{I}$ of $\mathcal{O}$. Further, $\varkappa$ and $\varkappa'$ are *$\mathcal{O}$-equivalent*, $\varkappa \equiv_{\mathcal{O}} \varkappa'$ in symbols, if they $\mathcal{O}$-entail each other. Clearly, $\bigcirc \boldsymbol{q} \equiv \bot \cup \boldsymbol{q}$ and $\Diamond \boldsymbol{q} \equiv \top \cup \boldsymbol{q}$, where $\equiv$ stands for $\equiv_{\emptyset}$.

Recall from [47] that, for any $LTL^{\Box\bigcirc}_{horn}$-ontology $\mathcal{O}$ and any instance $\mathcal{D}$ consistent with $\mathcal{O}$, there is a *canonical model* $\mathcal{C}_{\mathcal{O},\mathcal{D}}$ of $\mathcal{O}$ and $\mathcal{D}$ such that, for any query $\varkappa$ and $k \in \mathbb{N}$,

$$\mathcal{O}, \mathcal{D} \models \varkappa(k) \quad \text{iff} \quad \mathcal{C}_{\mathcal{O},\mathcal{D}} \models \varkappa(k). \tag{6}$$

Let $\mathsf{sub}_{\mathcal{O}}$ be the set of temporal concepts in $\mathcal{O}$ and their negations. A *type for* $\mathcal{O}$ is any maximal subset $tp \subseteq \mathsf{sub}_{\mathcal{O}}$ consistent with $\mathcal{O}$. Let $\boldsymbol{T}$ be the set of all types for $\mathcal{O}$. Given an interpretation $\mathcal{I}$, we denote by $tp_{\mathcal{I}}(n)$ the type for $\mathcal{O}$ that holds in $\mathcal{I}$ at $n \in \mathbb{N}$. For $\mathcal{O}$ consistent with $\mathcal{D}$, we abbreviate $tp_{\mathcal{C}_{\mathcal{O},\mathcal{D}}}$ to $tp_{\mathcal{O},\mathcal{D}}$. The canonical model has a periodic structure in the following sense:

Proposition 5. *For any $LTL^{\Box\bigcirc}_{horn}$ ontology $\mathcal{O}$ and any instance $\mathcal{D}$ consistent with $\mathcal{O}$, there are positive integers $s_{\mathcal{O},\mathcal{D}} \leq 2^{|\mathcal{O}|}$ and $p_{\mathcal{O},\mathcal{D}} \leq 2^{2|\mathcal{O}|}$ such that*

$$tp_{\mathcal{O},\mathcal{D}}(n) = tp_{\mathcal{O},\mathcal{D}}(n + p_{\mathcal{O},\mathcal{D}}), \qquad \text{for every } n \geq \max \mathcal{D} + s_{\mathcal{O},\mathcal{D}}. \tag{7}$$

If $\mathcal{O}$ is an $LTL^{\Box}_{horn}$ ontology, then $s_{\mathcal{O},\mathcal{D}} \leq |\mathcal{O}|$ and $p_{\mathcal{O},\mathcal{D}} = 1$.

3.1. Query by Example and Strongest Separators

We now formulate the problems we are concerned with in this paper.

Query by example. By an *example set* we mean a pair $E = (E^+, E^-)$, where E^+ and E^- are finite sets of data instances. We refer to the instances in E^+ and E^- as *positive* and *negative examples*, respectively. Given an ontology $\mathcal{O}$, we say that a query $\varkappa$ *separates* E *under* $\mathcal{O}$ if $\mathcal{O}, \mathcal{D} \models \varkappa(0)$, for each $\mathcal{D} \in E^+$, and $\mathcal{O}, \mathcal{D} \not\models \varkappa(0)$, for each $\mathcal{D} \in E^-$. Let $\mathcal{Q}$ be a class of queries. We say that E is *$\mathcal{Q}$-separable under* $\mathcal{O}$ if there exists $\varkappa \in \mathcal{Q}$ that separates E under $\mathcal{O}$.

The general *query-by-example problem* $\mathsf{QBE}(\mathcal{L}, \mathcal{Q})$, for an ontology language $\mathcal{L}$ and a query language $\mathcal{Q}$, is formulated as follows:

given an $\mathcal{L}$-ontology $\mathcal{O}$ and an example set E,

decide whether E is $\mathcal{Q}$-separable under $\mathcal{O}$.

If $\mathcal{L}$ only admits the empty ontology, we shorten $\mathsf{QBE}(\emptyset, \mathcal{Q})$ to $\mathsf{QBE}(\mathcal{Q})$. We are interested in the computational complexity of deciding $\mathsf{QBE}(\mathcal{L}, \mathcal{Q})$ for various $\mathcal{L}$ and $\mathcal{Q}$.

Example 6. (1) Let $\mathcal{D}_1 = \{A(1)\}$, $\mathcal{D}_2 = \{A(2)\}$ and $E = (\{\mathcal{D}_1\}, \{\mathcal{D}_2\})$. Then $\bigcirc A$ separates E (under $\mathcal{O} = \emptyset$) but no query in $\mathcal{Q}[\Diamond]$ does. On the other hand, E is not $\mathcal{Q}$-separable under $\mathcal{O} = \{\bigcirc A \to A\}$, for any class $\mathcal{Q}$ defined above, as $\mathcal{O}, \mathcal{D}_1 \models \varkappa(0)$ implies $\mathcal{O}, \mathcal{D}_2 \models \varkappa(0)$ for all $\varkappa \in \mathcal{Q}$.

(2) Let $E = (\{\mathcal{D}_1, \mathcal{D}_2\}, \emptyset)$ with $\mathcal{D}_1$ and $\mathcal{D}_2$ as in (1). Then $\Diamond A$ separates E but no $\mathcal{Q}[\bigcirc]$-query does. Observe that at least two positive examples are needed to achieve this effect. However, $\bigcirc\bigcirc\bigcirc B$ separates E under $\mathcal{O} = \{A \to \Box B\}$.

(3) Let $E = (\{\{A(1), B(2)\}, \{A(2), B(3)\}\}, \{\{A(3), B(5)\}\})$. Then $\Diamond(A \land \bigcirc B)$ separates E but no query in $\mathcal{Q}[\bigcirc]$ or $\mathcal{Q}[\Diamond]$ does.

(4) Let $E = (\{\{B(1)\}, \{A(1), B(2)\}\}, \{\{B(2)\}\})$. Then $A \,\mathsf{U}\, B$ separates E but no $\mathcal{Q}[\bigcirc, \Diamond]$-query does.

Note that if $\mathcal{Q}$ is closed under $\wedge$, $E = (E^+, E^-)$ and $E^- = \{\mathcal{D}_1^-, \dots, \mathcal{D}_n^-\}$, then E is $\mathcal{Q}$-separable under $\mathcal{O}$ iff each $(E^+, \{\mathcal{D}_i^-\})$ is, for $1 \leq i \leq n$. Indeed, if $\varkappa_i$ separates $(E^+, \{\mathcal{D}_i^-\})$, then $\varkappa_1 \wedge \dots \wedge \varkappa_n$ separates E. For such $\mathcal{Q}$, we can therefore assume that E^- consists of a single data instance.

As there may be multiple non-$\mathcal{O}$-equivalent queries that separate E under $\mathcal{O}$, it seems natural to ask whether there is a 'strongest' separator for E in a given class $\mathcal{Q}$ in the sense that it entails all other separators for E in $\mathcal{Q}$.

Strongest separators. We call a $\mathcal{Q}$-query $\varkappa$ *a strongest $\mathcal{Q}$-separator* for an example set E under an ontology $\mathcal{O}$ if $\varkappa$ separates E under $\mathcal{O}$ and $\varkappa \models_{\mathcal{O}} \varkappa'$ for any $\varkappa' \in \mathcal{Q}$ separating E under $\mathcal{O}$. We say that $\varkappa$ is a *strongest universal $\mathcal{Q}$-separator* for a set E^+ of positive instances under $\mathcal{O}$ if $\varkappa$ is a strongest $\mathcal{Q}$-separator for (E^+, E^-) under $\mathcal{O}$, for every $\mathcal{Q}$-separable example set (E^+, E^-). The *strongest separator existence problem* $\mathsf{SSE}(\mathcal{L}, \mathcal{Q})$ is formulated as follows:

given an $\mathcal{L}$-ontology $\mathcal{O}$ and an example set E,

decide whether there exists a strongest $\mathcal{Q}$-separator for E under $\mathcal{O}$.

In the problem $\mathsf{SUSE}(\mathcal{L}, \mathcal{Q})$, where U stands for 'universal', we are interested in the existence of a strongest universal $\mathcal{Q}$-separator for a given input set E^+ of data instances.

As before we set $\mathsf{SSE}(\mathcal{Q}) = \mathsf{SSE}(\emptyset, \mathcal{L})$ and $\mathsf{SUSE}(\mathcal{Q}) = \mathsf{SUSE}(\emptyset, \mathcal{L})$), in which case we drop qualification 'under $\mathcal{O}$'.

Example 7. (1) Let $E^+ = \{\{A(0), B(0)\}\}$ and $E^- = \{\{C(0)\}\}$. Then $\varkappa = A \wedge B$ is a strongest $\mathcal{Q}$-separator for $E = (E^+, E^-)$ (under $\mathcal{O} = \emptyset$), for any query language $\mathcal{Q}$. The same $\varkappa$ is also a strongest universal $\mathcal{Q}$-separator for E^+.

(2) A strongest $\mathcal{Q}_p[\Diamond]$-separator for E with $E^+ = \{\{A(2)\}, \{A(3)\}\}$ and $E^- = \{\{C(0)\}\}$ is $\Diamond\Diamond A$, which is also a strongest universal $\mathcal{Q}_p[\Diamond]$-separator for E^+.

(3) A strongest $\mathcal{Q}_p[\mathsf{U}]$-separator for $E = (E^+, E^-)$ with $E^+ = \{\{A(1), B(2)\}, \{B(1)\}\}$ and $E^- = \{\{C(0)\}\}$ is $A \,\mathsf{U}\, B$, which is also a strongest universal $\mathcal{Q}_p[\mathsf{U}]$-separator for E^+.

(4) Consider E with $E^+ = \{\{C(1), D(2), A(3), B(4)\}, \{A(1), B(2), C(3), D(4)\}\}$ and $E^- = \{\{C(1), D(2)\}\}$. Then $\Diamond(A \wedge \Diamond B)$ is a strongest $\mathcal{Q}_p[\Diamond]$-separator for E. Now, take $E^- = \{\{A(1), B(2)\}\}$. Then a strongest $\mathcal{Q}_p[\Diamond]$-separator for E is $\Diamond(C \wedge \Diamond D)$. Finally, if $E^- = \{\{A(1)\}\}$, then there is no strongest $\mathcal{Q}_p[\Diamond]$-separator for E (though E is $\mathcal{Q}_p[\Diamond]$-separable). There is no strongest universal $\mathcal{Q}_p[\Diamond]$-separator for E^+ either

(5) Suppose $\mathcal{O} = \{A \to \bigcirc B, C \to \bigcirc B, B \to \bigcirc\bigcirc B\}$, $E^+ = \{\{A(0)\}, \{C(0)\}\}$ and $E^- = \{\{D(0)\}\}$. Then $\Diamond B$ is a strongest $\mathcal{Q}[\Diamond]$-separator for E under $\mathcal{O}$.

Strongest separators are not invesigated further in this paper and are left for future work.

4. $\mathsf{QBE}(\mathcal{L}, \mathcal{Q}[\Diamond])$ and $\mathsf{QBE}(\mathcal{L}, \mathcal{Q}[\Diamond, \circ])$

We begin by introducing a normal form for queries in $\mathcal{Q}[\bigcirc, \Diamond]$. Denote by $\mathcal{Q}_p^{\bigcirc}[\Diamond]$ the class of $\mathcal{Q}[\bigcirc, \Diamond]$-queries of the form

$$\varkappa = \rho_0 \wedge \Diamond(\rho_1 \wedge \Diamond(\rho_2 \wedge \dots \wedge \Diamond\rho_n)), \tag{8}$$

where every ρ_i is a $\mathcal{Q}[\bigcirc]$-query. The following lemma generalizes a normal form introduced in [24] and can be proved in the same way.

Lemma 8. *For every $\varkappa$ in $\mathcal{Q}[\bigcirc, \Diamond]$ one can compute in polynomial time $\varkappa_1, \dots, \varkappa_n \in \mathcal{Q}_p^{\bigcirc}[\Diamond]$ such that $\varkappa$ is equivalent to $\varkappa_1 \wedge \ldots \wedge \varkappa_n$. If $\varkappa \in \mathcal{Q}[\Diamond]$, then $\varkappa_1, \dots, \varkappa_n \in \mathcal{Q}_p[\Diamond]$.*

We obtain the following reductions.

Corollary 9. *The problems $\mathsf{QBE}(\mathcal{L}, \mathcal{Q}[\bigcirc, \Diamond])$ and $\mathsf{QBE}(\mathcal{L}, \mathcal{Q}[\Diamond])$ are polynomially reducible to $\mathsf{QBE}(\mathcal{L}, \mathcal{Q}_p^{\bigcirc}[\Diamond])$ and $\mathsf{QBE}(\mathcal{L}, \mathcal{Q}_p[\Diamond])$, respectively.*

Proof. There exists a $\boldsymbol{q} \in \mathcal{Q}[\bigcirc, \Diamond]$ separating (E^+, E^-) iff there exist $\boldsymbol{q}_{\mathcal{D}} \in \mathcal{Q}_p^{\bigcirc}[\Diamond]$ separating $(E^+, \{\mathcal{D}\})$ for $\mathcal{D} \in E^-$. ❑

We first consider the complexity of separability without a mediating ontology. Then separability turns out to be NP-complete for all languages between $\mathcal{Q}_p[\Diamond]$ and $\mathcal{Q}[\bigcirc, \Diamond]$. Observe that only two concept names are required and that we encode the following *common subsequence problem*: given a set S of words and number k, does there exist a common subsequence of all words in S of length k?

Theorem 10. $\mathsf{QBE}(\mathcal{Q}_p[\Diamond])$, $\mathsf{QBE}(\mathcal{Q}[\Diamond])$, $\mathsf{QBE}(\mathcal{Q}_p[\bigcirc, \Diamond])$, *and* $\mathsf{QBE}(\mathcal{Q}[\bigcirc, \Diamond])$ *are NP-complete. The lower bound already holds for example sets with two atomic concepts and one negative example.*

Proof. We start with the upper bound. Let $\varkappa$ be a query of the form (8) and let $\mathcal{D} \models \varkappa(0)$ for some data instance $\mathcal{D}$. Then $\varkappa$ is equivalent to a query of the form (8) in which n does not exceed the maximal timestamp in $\mathcal{D}$ and each ρ_i is a $\bigcirc$-path query whose temporal depth also does not exceed the maximal timestamp in $\mathcal{D}$. (For instance, if $\rho_n \neq \emptyset$, then $n \leq \max \mathcal{D}$; otherwise, we can remove $\Diamond\rho_n$ from $\varkappa$.)

Now assume that $E = (E^+, E^-)$ is given. By the observation above, it follows that if E is $\mathcal{Q}_p^{\bigcirc}[\Diamond]$-separable, then it is separated by a query in $\mathcal{Q}_p^{\bigcirc}[\Diamond]$ that is of polynomial size in the size of E^+. An NP-algorithm can guess such a $\mathcal{Q}_p^{\bigcirc}[\Diamond]$-query and check in polynomial time whether it separates E.

For the lower bound we provide a polynomial time reduction of the following problem (KsubS) known to be NP-complete [25]: given a set S of words over a two-element alphabet and a natural number k, decide whether there exists a common subsequence σ of the words in S of length at least k.

Suppose that an instance S, k of (KsubS) over alphabet $\{A, B\}$ is given. We define E of the form $(E^+, \{\mathcal{D}^-\})$ such that the following conditions are equivalent:

- there exists a common subsequence of S of length k;
- there exists $\boldsymbol{q} \in \mathcal{Q}_p[\Diamond]$ that separates $(E^+, \{\mathcal{D}^-\})$;
- there exists $\boldsymbol{q} \in \mathcal{Q}_p^{\bigcirc}[\Diamond]$ that separates $(E^+, \{\mathcal{D}^-\})$.

As we have only a single negative example in E, the NP-lower bound follows for all languages given in Theorem 10. We represent each word $w \in S$ as an ABox $\mathcal{D}_w$ starting at time point 1 (for example, the word $w = ABBA$ is represented as $\mathcal{D}_w = \{A(1), B(2), B(3), A(4)\}$). Now let

$$\mathcal{D}^+ = \{A(k+2), B(k+2), \dots, A(k(k+2)), B(k(k+2))\}$$

and

$$\mathcal{D}^- = \mathcal{D}^+ \setminus \{A(k(k+2)), B(k(k+2))\}$$

and let $E^+ = \{\mathcal{D}_w \mid w \in S\} \cup \{\mathcal{D}^+\}$. Assume first that there exists a common subsequence $C_1 \cdots C_k$ of S of length k. Then $\Diamond(C_1 \wedge \Diamond(C_2 \wedge \cdots \wedge \Diamond C_k))$ separates $(E^+, \{\mathcal{D}^-\})$. Now assume that a query

$$\varkappa = \rho_0 \wedge \Diamond(\rho_1 \wedge \Diamond(\rho_2 \wedge \cdots \wedge \Diamond\rho_n)),$$

where every ρ_i a $\mathcal{Q}[\bigcirc]$-query, separates $(E^+, \{\mathcal{D}^-\})$. As $\mathcal{D}_w, 0 \models \varkappa$ for some $w \in S$, we have that $n \leq k$ and all ρ_i have depth bounded by k. Then $\rho_0 = \top$ and also, as there are 'gaps' of length $k+1$ between any two entries in $\mathcal{D}^+$ and since $\mathcal{D}^+, 0 \models \varkappa$ we may assume that each ρ_i, $i > 0$, is of the form $\bigcirc^{m_i}\rho_i'$ with $0 \leq m_i \leq k$ and ρ_i' a conjunction of atoms. Observe that we can satisfy, in $\mathcal{D}^+$,

- ρ_1 in the interval $\{1, \dots, k+2\}$;
- ρ_2 in the interval $\{(k+1)+1, \dots, 2(k+2)\}$;
- and so on, with ρ_n satisfied in the interval $\{(n-1)(k+2)+1, \dots, n(k+2)\}$.

In particular, if ρ_i is a conjunction of atoms, then it can be satisfied in $i(k+2)$. If $n < k$, then it follows directly that $\mathcal{D}^-, 0 \models \varkappa$, and we have derived a contradiction. Hence $n = k$. Then, as the depth of $\varkappa$ is bounded by k, ρ_k is a conjunction of atoms. In fact, one can now show by induction starting with ρ_{k-1} that all ρ_i, $i > 0$, are nonempty conjunctions of atoms. Otherwise a shift to the left shows that $\mathcal{D}^-, 0 \models \varkappa$ and we have derived a contradiction. Thus $\varkappa$ takes the form $\Diamond(\rho_1 \wedge \Diamond(\rho_2 \wedge \cdots \wedge \Diamond\rho_k))$ with all ρ_i non-empty. It follows from $\mathcal{D}_w \models \varkappa$ for all $w \in S$ that $\varkappa$ defines a common subsequence of S of length k, as required. ❑

Suppose next that the language $\mathcal{L}$ in $\mathsf{QBE}(\mathcal{L}, \mathcal{Q}_p[\Diamond])$ is an $LTL_{bool}^{\Box}$-ontology.

Theorem 11. *(i) If an example set E is $\mathcal{Q}_p[\Diamond]$-separable under an $LTL_{bool}^{\Box}$ ontology $\mathcal{O}$, then E can be separated under $\mathcal{O}$ by a $\mathcal{Q}_p[\Diamond]$-query of polynomial size in E and $\mathcal{O}$.*

(ii) $\mathsf{QBE}(LTL_{bool}^{\Box}, \mathcal{Q}_p[\Diamond])$ and $\mathsf{QBE}(LTL_{bool}^{\Box}, \mathcal{Q}[\Diamond])$ are both Σ_2^p-complete.

(iii) $\mathsf{QBE}(LTL_{horn}^{\Box}, \mathcal{Q}[\Diamond])$ and $\mathsf{QBE}(LTL_{horn}^{\Box}, \mathcal{Q}[\Diamond])$ are both NP-complete.

Proof. (i) Suppose $E = (E^+, E^-)$ is separated by a $\mathcal{Q}_p[\Diamond]$-query $\varkappa$ under $\mathcal{O}$. Then, for every $\mathcal{D} \in E^-$, there is a model $\mathcal{J}_\mathcal{D}$ of $\mathcal{O}$ and $\mathcal{D}$ such that $\mathcal{J}_\mathcal{D} \not\models \varkappa(0)$. As follows from [28] and since

$\varkappa$ is a positive existential query, we can assume that the types (in the vocabulary of $\mathcal{O}$ and $\mathcal{D}$) in $\mathcal{J}_\mathcal{D}$ form a sequence

$$tp_0, \dots, tp_k, tp_{k+1}, \dots, tp_{k+m}, tp_{k+1}, \dots, tp_{k+m}, \dots, tp_{k+1}, \dots, tp_{k+m}, \dots \tag{9}$$

with k and m polynomial in $\mathcal{D}$ and $\mathcal{O}$. Let K be the maximal such k over all $\mathcal{D} \in E^-$. If the depth n of $\varkappa$ of the form (4) does not exceed K, then $\varkappa$ is as required. If $K > n$, we shorten $\varkappa$ as follows. Consider first the 'prefix' $\varkappa'$ of $\varkappa$ formed by $\rho_0, \dots, \rho_K$. If $\mathcal{J}_\mathcal{D} \not\models \varkappa'(0)$, for all $\mathcal{D} \in E^-$, then $\varkappa'$ is as required. Otherwise, for each $\mathcal{D} \in E^-$, we pick some ρ_i with $i > K$ such that $\rho_i \not\subseteq tp_{k+j}$, for any j, $1 \leq j \leq k$, which must exist since $\mathcal{J}_\mathcal{D} \not\models \varkappa(0)$. Then we construct $\varkappa''$ by omitting from $\varkappa$ all ρ_l that are different from those in $\varkappa'$ and the chosen ρ_i with $i > K$. Clearly, $\varkappa''$ separates E and is of polynomial size in $\mathcal{D}$ and $\mathcal{O}$.

(ii) A Σ_2^p upper bound follows immediately from (i): we guess polynomial-size $\varkappa$ and $\mathcal{J}_\mathcal{D}$, for $\mathcal{D} \in E^-$ and then check in polynomial time that $\mathcal{J}_\mathcal{D} \models \mathcal{O}, \mathcal{D}$ and $\mathcal{J}_\mathcal{D} \not\models \varkappa(0)$ and in coNP [28] that $\mathcal{O}, \mathcal{D} \models \varkappa(0)$ for all $\mathcal{D} \in E^+$.

We establish a matching lower bound by reduction of the validity problem for fully quantified Boolean formulas of the form

$$\exists \boldsymbol{p}\, \forall \boldsymbol{q}\, \psi,$$

where ψ is a propositional formula, and $\boldsymbol{p} = p_1, \dots, p_k$ and $\boldsymbol{q} = q_1, \dots, q_m$ are lists of propositional variables. We assume w.l.o.g. that ψ is not a tautology. We also assume that $\neg\psi \not\models x$ for $x \in \{p_i, \neg p_i, q_j, \neg q_j \mid 1 \leq i \leq k, 1 \leq j \leq m\}$. Indeed, if $\neg\psi \models x$ then $\psi \equiv \neg x \vee \psi'$, for some ψ', and when $x \in \{p_i, \neg p_i\}$ the QBF formula $\exists \boldsymbol{p}\, \forall \boldsymbol{q}\, \psi$ is vacuously valid whereas when $x \in \{q_j \neg q_j\}$ the QBF formula $\exists \boldsymbol{p}\, \forall \boldsymbol{q}\, \psi$ is valid iff $\exists \boldsymbol{p}\, \forall \boldsymbol{q'}\, \psi'$ is, where $\boldsymbol{q'}$ is obtained from $\boldsymbol{q}$ by removing q_j. We regard propositional variables as atomic concepts and also use fresh atoms $A_1, \dots, A_k, \bar{A}_1, \dots, \bar{A}_k$ and B.

Let $E = (E^+, E^-)$ with $E^+ = \{\mathcal{D}_1, \mathcal{D}_2\}$, $E^- = \{\mathcal{D}_3\}$, where

$$\mathcal{D}_1 = \{B_1(0)\}, \quad \mathcal{D}_2 = \{B_2(0)\}, \quad \mathcal{D}_3 = \{q_1(0), q_2(0), \dots, q_m(0)\},$$

and let $\mathcal{O}$ contain (the normal forms of) the following axioms, for all $i = 1, \dots, k$:

$$B_1 \to \neg\psi, \quad B_2 \to \neg\psi, \tag{10}$$

$$p_i \to \Diamond\big(\bar{A}_i \wedge \bigwedge_{j \neq i}(A_j \wedge \bar{A}_j)\big), \qquad \neg p_i \to \Diamond\big(A_i \wedge \bigwedge_{j \neq i}(A_j \wedge \bar{A}_j)\big), \tag{11}$$

We show that $\exists \boldsymbol{p}\, \forall \boldsymbol{q}\, \psi$ is valid iff E is $\mathcal{Q}_p[\Diamond]$-separable under $\mathcal{O}$.

$(\Rightarrow)$ Suppose $\exists \boldsymbol{p}\, \forall \boldsymbol{q}\, \psi$ is valid. Take an assignment $\mathfrak{a}$ for the variables $\boldsymbol{p}$ such that under all assignments $\mathfrak{b}$ for the variables $\boldsymbol{q}$ formula ψ is true. Let C be the conjunction of all A_i with $\mathfrak{a}(p_i) = 1$ and all $\bar{A}_i$ with $\mathfrak{a}(p_i) = 0$, and let $\varkappa = \Diamond C$. We show that $\varkappa$ separates E. Define an interpretation $\mathcal{J}$ by taking

- $\mathcal{J}, 0 \models p_i$ iff $\mathfrak{a}(p_i) = 1$, for $i = 1, \dots, k$ and $\mathcal{J}, 0 \models q_j$, for $j = 1, \dots, m$;
- if $\mathcal{J}, 0 \models p_i$, then $\mathcal{J}, i \models \bar{A}_i \wedge \bigwedge_{j \neq i}(A_j \wedge \bar{A}_j)$;
- if $\mathcal{J}, 0 \not\models p_i$, then $\mathcal{J}, i \models A_i \wedge \bigwedge_{j \neq i}(A_j \wedge \bar{A}_j)$.

By the definition, $\mathcal{J}$ is a model of $\mathcal{O}$ and $\mathcal{D}_3$ with $\mathcal{J}, 0 \not\models \varkappa$. On the other hand, let $\mathcal{I}$ be a model of $\mathcal{O}$ and some $\mathcal{D}_l$, $l = 1, 2$. By (10), $\mathcal{I}, 0 \not\models \psi$. Then the truth values of the p_i in $\mathcal{I}$ at 0 cannot reflect the truth values of the p_i under $\mathfrak{a}$ (for otherwise ψ would be true at 0 in $\mathcal{I}$). Take some i_0 for which these truth values of p_{i_0} differ, say $\mathfrak{a}(p_{i_0}) = 1$ but $\mathcal{I}, 0 \not\models p_{i_0}$. Then $\mathcal{I}, 0 \models \Diamond(A_{i_0} \wedge \bigwedge_{j \neq i_0}(A_j \wedge \bar{A}_j))$, and so $\mathcal{I}, 0 \models \varkappa$.

($\Leftarrow$) Suppose a $\mathcal{Q}_p[\Diamond]$-query $\varkappa$ separates E but $\exists \boldsymbol{p} \forall \boldsymbol{q}\, \psi$ is not valid. From our conditions on ψ, it is easy to see by considering possible models of $\mathcal{O}$ and $\mathcal{D}_l$, $l = 1, 2, 3$, that $\varkappa$ does not contain occurrences of B_1, B_2, p_i, q_j, $1 \leq i \leq k$, $1 \leq j \leq m$. Let $\mathcal{J}$ be a model of $\mathcal{O}$ and $\mathcal{D}_3$ such that $\mathcal{J}, 0 \not\models \varkappa$. Let $\mathfrak{a}$ be the assignment for $\boldsymbol{p}$ given by $\mathcal{J}$ at 0. As $\exists \boldsymbol{p} \forall \boldsymbol{q}\, \psi$ is not valid, there is an assignment $\mathfrak{b}$ for $\boldsymbol{q}$ such that ψ is false under $\mathfrak{a}$ and $\mathfrak{b}$. Consider an interpretation $\mathcal{I}$ such that $\mathcal{I}, 0 \models B_1$, the truth values of $\boldsymbol{p}$ and $\boldsymbol{q}$ at 0 are given by $\mathfrak{a}$ and $\mathfrak{b}$, and all other atoms are interpreted as in $\mathcal{J}$. Then $\mathcal{I}$ is a model of $\mathcal{O}$ and $\mathcal{D}_1$, and so $\mathcal{I}, 0 \models \varkappa$. But then $\mathcal{J} \models \varkappa$, as $\varkappa$ can only contain atoms A_i and $\bar{A}_i$, which is a contradiction showing that $\exists \boldsymbol{p} \forall \boldsymbol{q}\, \psi$ is valid.

(iii) Follows from (i) and the fact that query answering is in polynomial time [49]. ❑

Extending $LTL^{\Box}_{horn}$ with the next-time operator $\bigcirc$ leads to the following result:

Theorem 12. *$\mathsf{QBE}(LTL^{\Box\bigcirc}_{horn}, \mathcal{Q}_p[\Diamond])$ and $\mathsf{QBE}(LTL^{\Box\bigcirc}_{horn}, \mathcal{Q}[\Diamond])$ are both PSPACE-complete.*

Proof. The lower bound follows from [50]. To obtain the upper one, we first recall from [47] that k and m in (9) for the canonical model $\mathcal{C}_{\mathcal{O},\mathcal{D}}$ of a given $LTL^{\Box\bigcirc}_{horn}$ ontology $\mathcal{O}$ and $\mathcal{D}$ are exponential in $|\mathcal{D}|$ and $|\mathcal{O}|$, and so a given $E = (E^+, E^-)$ is $\mathcal{Q}_p[\Diamond]$-separable under $\mathcal{O}$ iff it is separated by a $\mathcal{Q}_p[\Diamond]$-query $\varkappa$ of exponential size. Our nondeterministic PSPACE-algorithm incrementally guesses the ρ_i in (4) (in the signature of $\mathcal{O}$ and E) and checks whether they are satisfiable in some relevant part of the relevant $\mathcal{C}_{\mathcal{O},\mathcal{D}}$, which can be done in PSPACE. Details of the algorithm are given below.

Let $E^+ = \{\mathcal{D}^+_1, \dots, \mathcal{D}^+_n\}$ and $E^- = \{\mathcal{D}^-_1, \dots, \mathcal{D}^-_l\}$. Let K (respectively, M) be the maximum of all k (respectively, m) for $\mathcal{C}_{\mathcal{O},\mathcal{D}}$ with $\mathcal{D} \in E^+ \cup E^-$. The nondeterministic algorithm starts by guessing a conjunction of atoms ρ_0 and checking in PSPACE that $\mathcal{O}, \mathcal{D}^+_i \models \rho_0(0)$ for all $i \in [1, n]$. We use numbers $d^+_i, d^-_j \leq K + M$, for $i \in [1, n]$, $j \in [1, l]$, and a set $N \subseteq [1, l]$ that will keep track of the negative examples yet to be separated. Initially, we set all $d^+_i, d^-_j = 0$ and $N = \{j \in [1, l] \mid \mathcal{O}, \mathcal{D}^-_j \models \rho_0(0)\}$. Then we repeat the following steps until $N = \emptyset$, in which case the algorithm terminates accepting the input:

- Guess a conjunction ρ of atoms in the signature of $\mathcal{O}$ and E.
- For every $i \in [1, n]$, check in PSPACE that $\mathcal{O}, \mathcal{D}^+_i \models \Diamond\rho(d^+_i)$ and reject if this is not so.
- Guess $d^{+\prime}_i$ such that $\min(d^+_i, K) < d^{+\prime}_i \leq K + M$ and $\mathcal{O}, \mathcal{D}^+_i \models \rho(d^{+\prime}_i)$.
- For each, $j \in N$ check that $\mathcal{O}, \mathcal{D}^-_j \models \Diamond\rho(d^-_i)$. If no, remove j from N. Otherwise, find in PSPACE the smallest $d^{-\prime}_i$ such that $\min(d^-_i, K) < d^{-\prime}_i \leq K + M$ and $\mathcal{O}, \mathcal{D}^-_i \models \rho(d^{-\prime}_i)$.
- Set $d^+_i := d^{+\prime}_i$ and, for all j still in N, set $d^-_j := d^{-\prime}_i$.

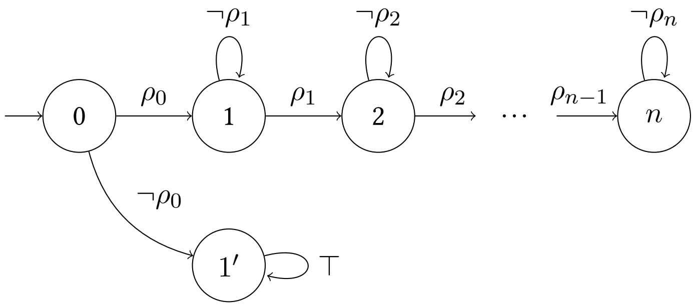
Let $\varphi_i = \rho_0 \wedge \Diamond(\rho_1 \wedge \Diamond(\dots(\rho_{i-1} \wedge \Diamond\rho_i)\dots))$, where ρ_i is the conjunction of atoms guessed in the i-th iteration. Let N_i be the set N after the i-th iteration. Then, for all $j \in [1, n]$, we have $\mathcal{O}, \mathcal{D}^+_j \models \varphi_i(0)$, for all $j \in N_i$ we have $\mathcal{O}, \mathcal{D}^-_j \models \varphi_i(0)$, and for all $j \in [1, l] \setminus N_i$ we have $\mathcal{O}, \mathcal{D}^-_j \not\models \varphi_i(0)$. So the algorithm accepts after the ℓ-th iteration iff φ_ℓ separates (E^+, E^-). ❑

For the full ontology language $LTL^{\Box\bigcirc}_{bool}$, we only obtain an upper bound:

Theorem 13. *$\mathsf{QBE}(LTL^{\Box\bigcirc}_{bool}, \mathcal{Q}_p[\Diamond])$ and $\mathsf{QBE}(LTL^{\Box\bigcirc}_{bool}, \mathcal{Q}[\Diamond])$ is in NExpTime.*

Proof. Similarly to the case of $\mathsf{QBE}(LTL^{\Box}_{bool}, \mathcal{Q}_p[\Diamond])$, one can show that if $E = (E^+, E^-)$ is $\mathcal{Q}_p[\Diamond]$-separable under $\mathcal{O}$, then there exists a separating $\mathcal{Q}_p[\Diamond]$-query of exponential size. We show that, having guessed a $\mathcal{Q}_p[\Diamond]$-query $\varkappa$ of exponential size, one can check in exponential time whether $\varkappa$ separates E under $\mathcal{O}$.

For every data instance $\mathcal{D}$, there is an automaton $\mathfrak{A}_{\mathcal{O},\mathcal{D}}$ over the alphabet of types for $\mathcal{O}$ and $\mathcal{D}$, which is computable and traversable in polynomial space (and therefore of exponential size), such that each infinite run of it corresponds to a model of $\mathcal{O}$ and $\mathcal{D}$ (see, e.g., [51, 47]). There is also an exponential-size automaton $\mathfrak{A}_{\neg\varkappa}$ (constructible and traversable in PSpace) whose infinite runs correspond to interpretations that do not satisfy $\varkappa(0)$. For instance, assuming that $\varkappa = \rho_0 \land \Diamond(\rho_1 \land \Diamond(\rho_2 \land \dots \land \Diamond\rho_n))$, one can take the automaton shown below:



Here, each transition ρ_i means the set of transitions that correspond to all of the types satisfying ρ_i, and the transitions $\neg\rho_i$ mean the set of transitions that correspond to all of the types not satisfying ρ_i. Then $\varkappa$ separates (E^+, E^-) under $\mathcal{O}$ iff

- for all $\mathcal{D} \in E^+$, there are no models (infinite runs) common to both $\mathfrak{A}_{\mathcal{O},\mathcal{D}}$ and $\mathfrak{A}_{\neg\varkappa}$;
- for all $\mathcal{D} \in E^-$, there is a model (infinite run) common to both $\mathfrak{A}_{\mathcal{O},\mathcal{D}}$ and $\mathfrak{A}_{\neg\varkappa}$.

This can be decided in exponential time because all of the automata can be traversed using only polynomial space. ❑

5. $\mathsf{QBE}(\mathcal{L}, \mathcal{Q}[\mathsf{U}^-])$

We next consider separability by $\mathcal{Q}[\mathsf{U}^-]$-queries, possibly mediated by Horn ontologies. We start by establishing upper complexity bounds.

5.1. Upper bounds

Theorem 14. *$\mathsf{QBE}(\mathcal{Q}_p[\mathsf{U}])$ and $\mathsf{QBE}(LTL^{\Box}_{horn}, \mathcal{Q}_p[\mathsf{U}])$ are in NP.*

Proof. We only need to show the result for $\mathsf{QBE}(LTL^{\Box}_{horn}, \mathcal{Q}_p[\mathsf{U}])$. Let M be the maximum of $\max \mathcal{D} + s_{\mathcal{O},\mathcal{D}}$ (see Proposition 5) over all $\mathcal{D} \in E^+ \cup E^-$. We observe that if E is separable under some $LTL^{\Box}_{horn}$ ontology $\mathcal{O}$ by a query $\varkappa$ of the form (5), then it is separable by $\varkappa$ with

$n \leq M$. Indeed, suppose $\mathcal{C}_{\mathcal{O},\mathcal{D}} \models \varkappa(0)$ for all $\mathcal{D} \in E^+$ and $\mathcal{C}_{\mathcal{O},\mathcal{D}} \not\models \varkappa(0)$ for all $\mathcal{D} \in E^-$. Let $X = \{A \in \Sigma \mid \mathcal{C}_{\mathcal{O},\mathcal{D}} \models A(M) \text{ for all } \mathcal{D} \in E^+\}$. It follows that $\rho_i \subseteq X$, $\lambda_{i+1} \subseteq X \cup \{\bot\}$ for all $i \geq M$. Clearly, if we remove the subquery $\lambda_{M+1} \mathsf{U} (\rho_{M+1} \wedge (\dots (\lambda_n \mathsf{U} \rho_n) \dots))$ from $\varkappa$, it is still the case that $\mathcal{C}_{\mathcal{O},\mathcal{D}} \models \varkappa(0)$ for all $\mathcal{D} \in E^+$ and $\mathcal{C}_{\mathcal{O},\mathcal{D}} \not\models \varkappa(0)$ for all $\mathcal{D} \in E^-$. An NP-algorithm can then guess a $\mathcal{Q}_p[\mathsf{U}]$-query of depth $\leq M$ and check in polynomial time (using dynamic programming) whether it separates E. ❑

To obtain our results for $\mathsf{QBE}(\mathit{LTL}^{\Box\bigcirc}_{\mathit{horn}}, \mathcal{Q}_p[\mathsf{U}])$ and $\mathsf{QBE}(\mathit{LTL}^{\Box\bigcirc}_{\mathit{horn}}, \mathcal{Q}[\mathsf{U}^-])$, we require some technical definitions. Let $\boldsymbol{\mathcal{M}} = \{\mathcal{I}_i \mid i \in I\}$ be a set of *LTL* interpretations. We assume that each $\mathcal{I}_i$ has a domain $(\mathbb{N}_i, <_i)$ isomorphic to $(\mathbb{N}, <)$ with $\mathbb{N}_i \cap \mathbb{N}_j = \emptyset$ for any $i \neq j$. We denote the copy of 0 in $\mathbb{N}_i$ by 0_i and set $\mathbf{0} = \bigcup_{i \in \boldsymbol{I}} \{0_i\}$ Let $\Delta \subseteq \bigcup_{i \in \boldsymbol{I}} \mathbb{N}_i$ be such that $\Delta \cap \mathbb{N}_i$ is finite and convex in $(\mathbb{N}_i, <_i)$. We call Δ an *arena* for $\boldsymbol{\mathcal{M}}$. For an arena Δ for $\boldsymbol{\mathcal{M}}$, we say that $\boldsymbol{d}$ is in Δ if $\boldsymbol{d} \subseteq \Delta$ and we say that $\boldsymbol{d}$ is in $\boldsymbol{\mathcal{M}}$ if there exists an arena Δ for $\boldsymbol{\mathcal{M}}$ such that $\boldsymbol{d} \subseteq \Delta$. For any $\boldsymbol{d}$ in $\boldsymbol{\mathcal{M}}$, we set $\boldsymbol{d}_i = \boldsymbol{d} \cap \mathbb{N}_i$. We call $\boldsymbol{d}$ *singleton* if $|\boldsymbol{d}_i| = 1$, $i \in I$. For a pair $\boldsymbol{d}, \boldsymbol{s}$ of singletons in $\boldsymbol{\mathcal{M}}$, we define $\boldsymbol{d} + \boldsymbol{s}$ as a component-wise sum of $\boldsymbol{d}$ and $\boldsymbol{s}$. By definition, it is a singleton in $\boldsymbol{\mathcal{M}}$. For singleton $\boldsymbol{d}$ and $\boldsymbol{d}'$, we write $\boldsymbol{d} \lessdot \boldsymbol{d}'$ if $\boldsymbol{d}_i <_i \boldsymbol{d}'_i$, $i \in I$. For singleton $\boldsymbol{d}$ and $\boldsymbol{d}'$ such that $\boldsymbol{d} \lessdot \boldsymbol{d}'$, we set $\nabla(\boldsymbol{d}, \boldsymbol{d}') = \bigcup_{i \in I} \{d \in \mathbb{N}_i \mid \boldsymbol{d}_i < d < \boldsymbol{d}'_i\}$. Observe that $\nabla(\boldsymbol{d}, \boldsymbol{d}')$ is in $\boldsymbol{\mathcal{M}}$, but not necessarily singleton. We write $\boldsymbol{\mathcal{M}}, \boldsymbol{d} \models \varkappa$ if $\mathcal{I}_i, d_i \models \varkappa$ for *all* $d_i \in \boldsymbol{d}_i$ and $i \in \boldsymbol{I}$. By $\mathcal{I}_i, \boldsymbol{d}_i \models \varkappa$ we mean that $\mathcal{I}_i, d_i \models \varkappa$ for *all* $d_i \in \boldsymbol{d}_i$.

Given an $\mathit{LTL}^{\Box\bigcirc}_{\mathit{horn}}$-ontology $\mathcal{O}$ and an example set $E = (E^+, E^-)$, let $E^+ = \{\mathcal{D}_i \mid i \in I\}$ and $E^- = \{\mathcal{D}_i \mid i \in J\}$, assuming that I and J are disjoint. We set $\boldsymbol{\mathcal{M}}^+_{\mathcal{O}} = \{\mathcal{C}_{\mathcal{O},\mathcal{D}_i} \mid i \in I\}$ and $\boldsymbol{\mathcal{M}}^-_{\mathcal{O}} = \{\mathcal{C}_{\mathcal{O},\mathcal{D}_i} \mid i \in J\}$. Consider an infinite tree $\mathfrak{T}^+$ (respectively, $\mathfrak{T}^-$) whose vertices are non-empty finite sequences $\mathfrak{s}$ of the form $\mathbf{0}\boldsymbol{s}^1 \dots \boldsymbol{s}^n$ such that $\mathbf{0} \lessdot \boldsymbol{s}^k$ are singletons in $\boldsymbol{\mathcal{M}}^+_{\mathcal{O}}$ (respectively, $\boldsymbol{\mathcal{M}}^-_{\mathcal{O}}$). The edges of $\mathfrak{T}^+$ ($\mathfrak{T}^-$) are defined by taking $\mathfrak{s} \to \mathfrak{s}'$ if $\mathfrak{s}' = \mathfrak{s}\boldsymbol{s}$. We denote by $\sum \mathfrak{s}$ the sum $\mathbf{0} + \boldsymbol{s}^1 + \dots + \boldsymbol{s}^n$, which is a singleton. For any finite subtree $\mathfrak{T}^+_2$ of $\mathfrak{T}^+$ and any subtree $\mathfrak{T}^-_2$ of $\mathfrak{T}^-$, we say that $\mathfrak{T}^+_2$ is *homomorphically embeddable into* $\mathfrak{T}^-_2$ if there is a map h such that $\mathfrak{s} \to \mathfrak{s}'$ in $\mathfrak{T}^+_2$ implies $h(\mathfrak{s}) \to h(\mathfrak{s}')$ in $\mathfrak{T}^-_2$ and

(at′) $\boldsymbol{\mathcal{M}}^+_{\mathcal{O}}, \sum \mathfrak{s} \models A$ implies $\boldsymbol{\mathcal{M}}^-_{\mathcal{O}}, \sum h(\mathfrak{s}) \models A$, for all $A \in \Sigma$;

(nxt′) $\boldsymbol{\mathcal{M}}^+_{\mathcal{O}}, \nabla(\sum \mathfrak{s}, \sum \mathfrak{s}') \models A$ implies $\boldsymbol{\mathcal{M}}^-_{\mathcal{O}}, \nabla(\sum h(\mathfrak{s}), \sum h(\mathfrak{s}')) \models A$, for all $A \in \Sigma \cup \{\bot\}$.

For a subtree of $\mathfrak{T}^+$, we say that it is *finitely homomorphically embeddable into* $\mathfrak{T}^-_2$ if every finite subtree of $\mathfrak{T}^+$ is homomorphically embeddable into $\mathfrak{T}^-_2$.

Theorem 15. *(i) E is not $\mathcal{Q}[\mathsf{U}^-]$-separable under an $\mathit{LTL}^{\Box\bigcirc}_{\mathit{horn}}$-ontology $\mathcal{O}$ iff $\mathfrak{T}^+$ is finitely homomorphically embeddable into $\mathfrak{T}^-$.*

(ii) E is not $\mathcal{Q}_p[\mathsf{U}]$-separable under $\mathcal{O}$ iff every finite path in $\mathfrak{T}^+$ is homomorphically embeddable into $\mathfrak{T}^-$.

Proof. The proof is a straightforward modification of the proof of Theorem 25. ❑

Let $\mathfrak{T}^+_1$ be the restriction of $\mathfrak{T}^+$ to $\mathfrak{s}$ with $\boldsymbol{s}^k_i \in [1, M_i)$, where $M_i = \max \mathcal{D}_i + s_{\mathcal{O},\mathcal{D}_i} + 2p_{\mathcal{O},\mathcal{D}_i}$ for $i \in I$. Similarly, let $\mathfrak{T}^-_1$ be the tree defined as $\mathfrak{T}^-$ but with the restriction that $\boldsymbol{s}^k_i \in [1, M_i)$, where $M_i = \max \mathcal{D}_i + s_{\mathcal{O},\mathcal{D}_i} + p_{\mathcal{O},\mathcal{D}_i}$ for $i \in J$. The following lemma is the first step towards an algorithm for checking the criterion of Theorem 15 (i):

Lemma 16. (i) *$\mathfrak{T}^+$ is finitely homomorphically embeddable into $\mathfrak{T}^-$ iff $\mathfrak{T}^+_1$ is finitely homomorphically embeddable into $\mathfrak{T}^-_1$.*

(ii) *Every finite path of $\mathfrak{T}^+$ is homomorphically embeddable into $\mathfrak{T}^-$ iff every finite path of $\mathfrak{T}^+_1$ is homomorphically embeddable into $\mathfrak{T}^-_1$.*

Proof. (i) First, we show that $\mathfrak{T}^+$ is finitely homomorphically embeddable into $\mathfrak{T}^-$ iff $\mathfrak{T}^+_1$ is finitely homomorphically embeddable into $\mathfrak{T}^-$. ($\Rightarrow$) is straightforward because $\mathfrak{T}^+_1 \subseteq \mathfrak{T}^+$. ($\Leftarrow$) Suppose $\mathfrak{T}^+_1$ is finitely homomorphically embeddable into $\mathfrak{T}^-$. Let $\mathfrak{T}'$ be a finite subtree of $\mathfrak{T}^+$ and let n be the longest sequence in $\mathfrak{T}'$. We define $\mathfrak{T}''$ as a tree with all sequences $\mathbf{0}\boldsymbol{s}^1 \dots \boldsymbol{s}^m$, $m \leq n$, and $\boldsymbol{s}^j_i \in [1, M_i)$. Clearly, $\mathfrak{T}''$ is a finite subtree of $\mathfrak{T}^+_1$; take its embedding h into $\mathfrak{T}^-$. We now show how to construct an embedding h' of $\mathfrak{T}'$ into $\mathfrak{T}^-$. We set $h'(\mathbf{0}) = h(\mathbf{0})$. Now, consider any sequence $\mathbf{0}\boldsymbol{s}^1$ in $\mathfrak{T}'$. If $\boldsymbol{s}^1_i \in [1, M_i)$ for all i, we set $h'(\mathbf{0}\boldsymbol{s}^1) = h(\mathbf{0}\boldsymbol{s}^1)$. Suppose now $\boldsymbol{s}^1_i \geq M_i$. Then let $D_i = \boldsymbol{s}^1_i - \max \mathcal{D}_i - s_{\mathcal{O},\mathcal{D}_i} \pmod{p_{\mathcal{O},\mathcal{D}_i}}$ and take s_i equal to $\max \mathcal{D}_i + s_{\mathcal{O},\mathcal{D}_i} + p_{\mathcal{O},\mathcal{D}_i} + D_i$. Clearly, $s_i \in [1, M_i)$. We now construct $\boldsymbol{e}^1$ by taking $\boldsymbol{e}^1_i = \boldsymbol{s}^1_i$, if $\boldsymbol{s}^1_i \in [1, M_i)$ and taking $\boldsymbol{e}^1_i = s_i$ otherwise. We set $h'(\mathbf{0}\boldsymbol{s}^1) = h(\mathbf{0}\boldsymbol{e}^1)$. It can be readily verified that **(at′)** holds for h' and $\mathfrak{s} = \mathbf{0}\boldsymbol{s}^1$ while **(nxt′)** holds for $\mathfrak{s}' = \mathbf{0}$ and $\mathfrak{s}' = \mathbf{0}\boldsymbol{s}^1$. Now, consider any sequence $\mathbf{0}\boldsymbol{s}^1\boldsymbol{s}^2$ in $\mathfrak{T}'$. We show how to define h' for it. If $\boldsymbol{s}^2_i \in [1, M_i)$ for all i, we set $h'(\mathbf{0}\boldsymbol{s}^1\boldsymbol{s}^2) = h(\mathbf{0}\boldsymbol{e}^1\boldsymbol{s}^2)$. Suppose now $\boldsymbol{s}^2_i \geq M_i$. Then let $D_i = (\boldsymbol{s}^2_i - \max \mathcal{D}_i - s_{\mathcal{O},\mathcal{D}_i}) \mod p_{\mathcal{O},\mathcal{D}_i}$ and take s_i equal to $\max \mathcal{D}_i + s_{\mathcal{O},\mathcal{D}_i} + p_{\mathcal{O},\mathcal{D}_i} + D_i$. Clearly, $s_i \in [1, M_i)$. We now construct $\boldsymbol{e}^2$ by taking $\boldsymbol{e}^2_i = \boldsymbol{s}^2_i$ if $\boldsymbol{s}^2_i \in [1, M_i)$ and $\boldsymbol{e}^2_i = s_i$ otherwise. Set $h'(\mathbf{0}\boldsymbol{s}^1\boldsymbol{s}^2) = h(\mathbf{0}\boldsymbol{e}^1\boldsymbol{e}^2)$. It can be readily verified that **(at′)** and **(nxt′)** hold for the corresponding $\mathfrak{s}$.

To complete the proof of Lemma 16 (i), we show that for any finite subtree of $\mathfrak{T}^+_1$, there is a homomorphic embedding of it into $\mathfrak{T}^-$ iff there exists an embedding of it into $\mathfrak{T}^-_1$. The direction ($\Leftarrow$) is trivial, we only need to show ($\Rightarrow$). Let $\mathfrak{T}'$ be a finite subtree of $\mathfrak{T}^+_1$ and h' its homomorphic embedding into $\mathfrak{T}^-$. Consider the maximal subtree of $\mathfrak{T}^+_1$ genetated by $\mathbf{0}$ and satisfying $h'(\mathfrak{s}')_i - h'(\mathfrak{s})_i < M_i$, for all $i \in J$ and each edge $\mathfrak{s} \to \mathfrak{s}'$. For every $\mathfrak{s}$ in it, set $h(\mathfrak{s}) = h'(\mathfrak{s})$. Take $\mathfrak{s} \to \mathfrak{s}'$ closest to $\mathbf{0}$ such that $h'(\mathfrak{s}')_i - h'(\mathfrak{s})_i \geq M_i$, for some $i \in J$. Let $D_i = h'(\mathfrak{s}')_i - h'(\mathfrak{s})_i - \max \mathcal{D}_i - s_{\mathcal{O},\mathcal{D}_i} \pmod{p_{\mathcal{O},\mathcal{D}_i}}$ and set $h(\mathfrak{s}')_i = h'(\mathfrak{s})_i + \max \mathcal{D}_i + s_{\mathcal{O},\mathcal{D}_i} + D_i$ (recall that $h(\mathfrak{s}) = h'(\mathfrak{s})$). Clearly, $h(\mathfrak{s}')_i - h(\mathfrak{s})_i < M_i$ and conditions **(at′)** and **(nxt′)** hold for $\mathfrak{s}, \mathfrak{s}'$. Now, consider $\mathfrak{s}' \to \mathfrak{s}''$ and suppose $h'(\mathfrak{s}'')_i - h(\mathfrak{s}')_i \geq M_i$. Let $D_i = h'(\mathfrak{s}'')_i - h(\mathfrak{s}')_i - \max \mathcal{D}_i - s_{\mathcal{O},\mathcal{D}_i} \pmod{p_{\mathcal{O},\mathcal{D}_i}}$ and set $h(\mathfrak{s}'')_i = h(\mathfrak{s}')_i + \max \mathcal{D} + i + s_{\mathcal{O},\mathcal{D}_i} + D_i$. It can be readily verified that **(at′)** and **(nxt′)** hold for the corresponding $\mathfrak{s}$.

The proof of (ii) is similar and left to the reader. ❑

By a *transition system* we mean a structure $S = (\Sigma_1, \Sigma_2, W, L, R, s_0)$, where Σ_1 (Σ_2) is an alphabet of state (respectively, transition) labels, W is a set of states s, L is a set of labelled states (s, a), $s \in W$, $a \in \Sigma_1$, R is a set of labelled transitions (s, s', b), $s, s' \in W$, $b \in \Sigma_2$ between states, and $s_0 \in W$ is an initial state. A *run* of S is a finite sequence $(s_0, a_0)(s_0, s_1, b_1)(s_1, a_1)(s_1, s_2, b_2) \dots (s_{n-1}, a_{n-1})(s_{n-1}, s_n, b_n)(s_n, a_n)$ such that $(s_i, a_i) \in L$ and $(s_i, s_{i+1}, b_{i+1}) \in R$. Given two transition systems $S_1 = (\Sigma_1, \Sigma_2, W, L, R, s_0)$ and $S_2 = (\Sigma_1, \Sigma_2, W', L', R', s'_0)$ (with the same state and transition alphabets), we say that S_1 *is finitely contained in* S_2 if, for every run $(s_0, a_0)(s_0, s_1, b_1)(s_1, a_1)(s_1, s_2, b_2) \dots (s_{n-1}, a_{n-1})(s_{n-1}, s_n, b_n)(s_n, a_n)$ of S_1, there exists a run $(s'_0, a_0)(s'_0, s'_1, b_1)(s'_1, a_1)(s'_1, s'_2, b_2)\dots(s'_{n-1}, a_{n-1})(s'_{n-1}, s'_n, b_n)(s'_n, a_n)$ of S_2. We say that S_1 *is finitely simulated by* S_2 if, for each finite $W_1 \subseteq W$, there is a relation $H \subseteq W_1 \times W'$ such that the following conditions are satisfied:

- $(s_0, s'_0) \in H$;
- $(s, a) \in L$ implies $(s', a) \in L'$, for all $(s, s') \in H$ and $a \in \Sigma_1$;
- if $(s, t, b) \in R$ and $(s, s') \in H$, then there exists $t' \in W'$ such that $(s', t', b) \in R'$ and $(t, t') \in H$, for all $s, t \in W_1$ and $s' \in W'$.

We say that S_1 *is simulated by* S_2 if the above condition is satisfied for $W_1 = W$. (It is easy to see that if W is finite, then S_1 is finitely simulated by S_2 iff S_1 is simulated by S_2.) We require the following known results (see, e.g., [31]):

Theorem 17. *Let S_1 and S_2 be* finite *transitions systems. Then it can be checked in PSPACE if S_1 is finitely contained in S_2, and it can be checked in P if S_1 is (finitely) simulated by S_2.*

We next convert the trees $\mathfrak{T}_1^\circ$, for $\circ \in \{+, -\}$, into infinite transition systems $S^\circ = (\Sigma_1, \Sigma_2, W^\circ, L^\circ, R^\circ, s_0^\circ)$. The state label alphabet Σ_1 of S° is 2^Σ and the transition label alphabet is $\Sigma_2 = 2^{\Sigma \cup \{\bot\}}$. The states of S° are the nodes $\mathfrak{s}$ of $\mathfrak{T}_1^\circ$ and the initial state is $\mathbf{0}$. We define L^+ as the set of pairs $(\mathfrak{s}, X_\mathfrak{s}^+)$, where $\mathfrak{s}$ is in $\mathfrak{T}_1^+$ and $X_\mathfrak{s}^\circ = \{A \in \Sigma \mid \boldsymbol{\mathcal{M}}_\mathcal{O}^\circ, \sum \mathfrak{s} \models A\}$. We define R^+ as the set of triples $(\mathfrak{s}, \mathfrak{s}', X_{\mathfrak{s},\mathfrak{s}'}^+)$, where $\mathfrak{s} \to \mathfrak{s}'$ in $\mathfrak{T}_1^+$ and $X_{\mathfrak{s},\mathfrak{s}'}^\circ$ (for $\mathfrak{s} \to \mathfrak{s}'$ in $\mathfrak{T}_1^\circ$) equals $\{A \in \Sigma \cup \{\bot\} \mid \boldsymbol{\mathcal{M}}_\mathcal{O}^\circ, \nabla(\sum \mathfrak{s}, \sum \mathfrak{s}') \models A\}$. Finally, we define L^- as the set of pairs $(\mathfrak{s}, X)$, for $\mathfrak{s}$ in $\mathfrak{T}_1^-$ and $X \subseteq X_\mathfrak{s}^-$ and R^- as the set of triples $(\mathfrak{s}, \mathfrak{s}', X)$, for $\mathfrak{s} \to \mathfrak{s}'$ in $\mathfrak{T}_1^-$ and $X \subseteq X_{\mathfrak{s},\mathfrak{s}'}^-$. The following result is an immediate consequence of Lemma 16 and Theorem 15:

Theorem 18. (i) *E is not $\mathcal{Q}[\mathsf{U}^-]$-separable under $\mathcal{O}$ iff S^+ is finitely simulated by S^-.*
(ii) *E is not $\mathcal{Q}_p[\mathsf{U}]$-separable under $\mathcal{O}$ iff S^+ is finitely contained in S^-.*

As the last step towards the algorithms for $\mathsf{QBE}(\mathit{LTL}^{\Box\bigcirc}_{\mathit{horn}}, \mathcal{Q}_p[\mathsf{U}])$ and $\mathsf{QBE}(\mathit{LTL}^{\Box\bigcirc}_{\mathit{horn}}, \mathcal{Q}[\mathsf{U}^-])$, we replace S°, for $\circ \in \{+, -\}$, by finite transition systems $S_1^\circ = (\Sigma_1, \Sigma_2, W_1^\circ, L_1^\circ, R_1^\circ, t_0^\circ)$. Denote I by I^+ and J by I^-. Let $N_i = \max \mathcal{D}_i + s_{\mathcal{O},\mathcal{D}_i} + p_{\mathcal{O},\mathcal{D}_i}$ for $i \in I^+ \cup I^-$ (thus, $N_i < M_i$ for $i \in I^+$ and $N_i = M_i$ for $i \in I^-$). The set of states W_1° of S_1° contains all $\boldsymbol{d}$ in $\boldsymbol{\mathcal{M}}_\mathcal{O}^\circ$ such that $\boldsymbol{d}_i \in [0, N_i)$. The initial states t_0° are $\mathbf{0}$. To define the transitions and labels, we require several technical definitions. For each $i \in I^+ \cup I^-$, we define the function $r_i \colon \mathbb{N}_i \to \mathbb{N}_i$ such that $r_i(d) = d$, if $d \in [0, M_i)$ and $r_i(d) = [d - (\max \mathcal{D}_i + s_{\mathcal{O},\mathcal{D}_i})] \pmod{p_{\mathcal{O},\mathcal{D}_i}} + \max \mathcal{D}_i + s_{\mathcal{O},\mathcal{D}_i}$ otherwise. For $\boldsymbol{d}$ as above and $\boldsymbol{s}$ with $\boldsymbol{s}_i \in [0, M_i)$, we define $\boldsymbol{d} \oplus \boldsymbol{s}$ equal to $\boldsymbol{d}'$ such that $\boldsymbol{d}'_i = r_i(\boldsymbol{d}_i + \boldsymbol{s}_i)$. Denote by per_i^k, for $k \in \mathbb{N}$, the interval

$$[\max \mathcal{D}_i + s_{\mathcal{O},\mathcal{D}_i} + p_{\mathcal{O},\mathcal{D}_i} k, \max \mathcal{D}_i + s_{\mathcal{O},\mathcal{D}_i} + p_{\mathcal{O},\mathcal{D}_i}(k+1)).$$

We define $\nabla'(\boldsymbol{d}, \boldsymbol{s})$ as the set $\boldsymbol{S} = \bigcup_{i \in I} \boldsymbol{S}_i$, where $\boldsymbol{S}_i \subseteq \mathbb{N}_i^+$ and

$$\boldsymbol{S}_i = \begin{cases} (\boldsymbol{d}_i, \boldsymbol{d}_i + \boldsymbol{s}_i), & \text{if } \boldsymbol{d}_i, \boldsymbol{d}_i + \boldsymbol{s}_i < \max \mathcal{D}_i + s_{\mathcal{O},\mathcal{D}_i} \text{ or} \\ & \boldsymbol{d}_i < \max \mathcal{D}_i + s_{\mathcal{O},\mathcal{D}_i}, \boldsymbol{d}_i + \boldsymbol{s}_i \in \mathit{per}_i^0; \\ (\boldsymbol{d}_i, \max \mathcal{D}_i + s_{\mathcal{O},\mathcal{D}_i}) \cup \mathit{per}_i^0, & \text{if } \boldsymbol{d}_i < \max \mathcal{D}_i + s_{\mathcal{O},\mathcal{D}_i}, \boldsymbol{d}_i + \boldsymbol{s}_i \in \mathit{per}_i^k, \\ & \text{for some } k \geq 1; \\ (r_i(\boldsymbol{d}_i), r_i(\boldsymbol{d}_i + \boldsymbol{s}_i)), & \text{if } \boldsymbol{d}_i, \boldsymbol{d}_i + \boldsymbol{s}_i \in \mathit{per}_i^k, \text{ for some } k \in \mathbb{N}; \\ \mathit{per}_i^0, & \text{if } \boldsymbol{d}_i \in \mathit{per}_i^k, \boldsymbol{d}_i + \boldsymbol{s}_i \in \mathit{per}_i^{k+2} \text{ for some } k \in \mathbb{N}; \\ (r_i(\boldsymbol{d}_i), \max \mathcal{D}_i + s_{\mathcal{O},\mathcal{D}_i} + p_{\mathcal{O},\mathcal{D}_i}) \cup & \\ \quad [\max \mathcal{D}_i + s_{\mathcal{O},\mathcal{D}_i}, r_i(\boldsymbol{d}_i + \boldsymbol{s}_i)), & \text{if } \boldsymbol{d}_i \in \mathit{per}_i^k, \boldsymbol{d}_i + \boldsymbol{s}_i \in \mathit{per}_i^{k+1} \text{ for some } k \in \mathbb{N}. \end{cases}$$

Now, define L_1^+ as the set of pairs $(\boldsymbol{d}, X_{\boldsymbol{d}}^+)$, where $\boldsymbol{d} \in W_1^+$ and $X_{\boldsymbol{d}}^\circ = \{A \in \Sigma \mid \boldsymbol{\mathcal{M}}_{\mathcal{O}}^\circ, \boldsymbol{d} \models A\}$, for $\boldsymbol{d}$ in $\boldsymbol{\mathcal{M}}_{\mathcal{O}}^\circ$. We define R_1^+ as the set of triples $(\boldsymbol{d}, \boldsymbol{d} \oplus \boldsymbol{s}, X_{\boldsymbol{d},\boldsymbol{s}}^+)$, where $\boldsymbol{d}$ is as above, $\boldsymbol{s}$ is such that $\boldsymbol{s}_i \in [0, M_i)$, and $X_{\boldsymbol{d},\boldsymbol{s}}^\circ = \{A \in \Sigma \cup \{\bot\} \mid \boldsymbol{\mathcal{M}}_{\mathcal{O}}^\circ, \nabla'(\boldsymbol{d}, \boldsymbol{s}) \models A\}$ for $\boldsymbol{d}$ in $\boldsymbol{\mathcal{M}}_{\mathcal{O}}^\circ$ and $\boldsymbol{s}$ as above. Finally, we define L_1^- as the set of pairs $(\boldsymbol{d}, X)$, where $\boldsymbol{d} \in W_1^-$ and $X \subseteq X_{\boldsymbol{d}}^-$, and define R_1^- as the set of triples $(\boldsymbol{d}, \boldsymbol{d} \oplus \boldsymbol{s}, X)$, where $\boldsymbol{d}$ is in W_1^-, $\boldsymbol{s}$ is such that $\boldsymbol{s}_i \in [0, M_i)$, and $X \subseteq X_{\boldsymbol{d},\boldsymbol{s}}^-$. Using the periodic structure of the canonical models and so the transitions in S°, we can show the following:

Theorem 19. (i) *S^+ is finitely simulated by S^- iff S_1^+ is simulated by S_1^-.*
(ii) *S^+ is finitely contained in S^- iff S_1^+ is finitely contained in S_1^-.*

It can be easily verified that S_1^+ and S_1^- are of the exponential size in $|\mathcal{O}| + |E^-|$ and $|\mathcal{O}| + |E^+|$, respectively, and can be constructed in ExpTime. Therefore, using the above result together with Theorems 17 and 18, we obtain:

Theorem 20. $\mathsf{QBE}(\mathit{LTL}_{\mathit{horn}}^{\Box\bigcirc}, \mathcal{Q}[\mathsf{U}^-])$ *is in ExpTime and* $\mathsf{QBE}(\mathit{LTL}_{\mathit{horn}}^{\Box\bigcirc}, \mathcal{Q}_p[\mathsf{U}])$ *is in ExpSpace.*

5.2. Lower bounds

Theorem 21. $\mathsf{QBE}(\mathcal{Q}_p[\mathsf{U}])$ *is NP-complete.*

Proof. The proof is by reduction of KsubS. Suppose an instance S, k of (KsubS) over the alphabet $\{A, B\}$ is given. We represent each word $w \in S$ as an ABox $\mathcal{D}_w$ as in the proof of Theorem 10. Let $\mathcal{D}_n = \{A(1), B(1), \dots, A(n), B(n)\}$ and $\mathcal{D}_n^+ = \{A(2i), B(2i) \mid 1 \leq i \leq n\}$.

Define $E = (E^+, E^-)$ by taking $E^+ = \{\mathcal{D}_w \mid w \in S\} \cup \{\mathcal{D}_k, \mathcal{D}_k^+\}$ and $E^- = \{\mathcal{D}_{k-1}, \mathcal{D}_{k-1}^+\}$. We show that E is $\mathcal{Q}_p[\mathsf{U}]$-separable iff there exists a common subsequence σ of the words in S of length at least k.

($\Leftarrow$) If $\sigma = C_1 \dots C_k \in \{A, B\}^k$ is a common subsequence of $w \in S$, then E is separated by the query $\top \,\mathsf{U}\, (C_1 \wedge \top \,\mathsf{U}\, (C_2 \wedge \top \,\mathsf{U}\, (\dots \wedge (\top \,\mathsf{U}\, C_k) \dots)))$.

($\Rightarrow$) Suppose a $\mathcal{Q}_p[\mathsf{U}]$-query $\varkappa = \lambda_0 \,\mathsf{U}\, (\rho_1 \wedge \lambda_2 \,\mathsf{U}\, (\rho_2 \wedge \lambda_3 \,\mathsf{U}\, (\dots \wedge (\lambda_l \,\mathsf{U}\, \rho_l) \dots)))$ with $\rho_l \neq \emptyset$ separates E. As $\mathcal{D}_k \in E^+$, we have $l \leq k$. As $\mathcal{D}_{k-1} \in E^-$, $l \geq k$, and so $l = k$.

For any ABox $\mathcal{D}$, we have $\mathcal{D} \models \varkappa(0)$ iff there is $f\colon [0,k] \to \mathbb{N}$ such that $f(0)=0$, $\mathcal{D} \models \rho_i(f(i))$, for $0 < i \leq k$,

$$\text{and } \mathcal{D} \models \lambda_i(j) \text{ for all } j \in (f(i-1), f(i)). \quad (12)$$

Consider such a map f for $\mathcal{D}_k^+ \in E^+$. Since $\rho_k \neq \top$, we have $f(k) \leq 2k$. If $f(k) < 2k$, then $\mathcal{D}_{k-1}^+ \models \varkappa(0)$, contrary to $\mathcal{D}_{k-1}^+ \in E^-$. So $f(k) = 2k$. Suppose there is $i \leq k$ with $f(i) - f(i-1) \geq 3$. Take f' with $f'(j) = f(j)$, for $j < i$, and $f'(j) = f(j) - 2$, for $j \geq i$. The map f' satisfies (12), and so $\mathcal{D}_{k-1}^+ \models \varkappa(0)$, which is again a contradiction. It follows that $f(i) = 2i$, for all $i \in [0,k]$, and $\lambda_i = \top$, for all $i \in [0,k]$. Now, if there is $\rho_i = \top$, then take the map f' with $f'(j) = f(j)$ for $j < i$, $f'(i) = 2i - 1$, and $f'(j) = 2j - 2$ for $j > i$. It satisfies (12), and so $\mathcal{D}_{k-1}^+ \models \varkappa(0)$. Thus, $\rho_i \neq \top$ for all i. Since $\mathcal{D}_w \in E^+$ for all $w \in S$, each ρ_i is either A or B, and each w contains the subsequence $\rho_1, \dots, \rho_k$. ❑

Theorem 22. $\mathsf{QBE}(LTL^{\bigcirc}_{horn}, \mathcal{Q}_p[\mathsf{U}])$ *is NExpTime-hard.*

Proof. Let $\boldsymbol{M}$ be a non-deterministic Turing machine that accepts words $\boldsymbol{x}$ over its tape alphabet in at most $N = 2^{p(|\boldsymbol{x}|)}$ steps, for some polynomial p. Given such an $\boldsymbol{M}$ and an input $\boldsymbol{x}$, our aim is to define an $LTL^{\bigcirc}_{horn}$ ontology $\mathcal{O}$ and an example set $E = (E^+, E^-)$ of size polynomial in $\boldsymbol{M}$ and $\boldsymbol{x}$ such that E is separated by a $\mathcal{Q}_p[\mathsf{U}]$-query under $\mathcal{O}$ iff $\boldsymbol{M}$ accepts $\boldsymbol{x}$.

Suppose $\boldsymbol{M} = (Q, \Sigma, \delta, \mathsf{b}, q_0, q_{acc})$ with a set Q of states, tape alphabet Σ with b for blank, transitional relation $\delta \subseteq (Q \times \Sigma) \times (Q \times \Sigma \times \{-1, 0, 1\})$, initial state q_0 and accepting state q_{acc}. Without loss of generality we assume that $\boldsymbol{M}$ erases the tape before accepting, its head is at the left-most cell in any accepting configuration, and if $\boldsymbol{M}$ does not accept the input, it runs forever. Given an input word $\boldsymbol{x} = x_1 \dots x_n$ over Σ, we represent configurations $\mathfrak{c}$ of a computation of $\boldsymbol{M}$ on $\boldsymbol{x}$ by the $(N-1)$-long word written on the tape (with sufficiently many blanks at the end), in which the symbol y in the active cell is replaced by the pair (q, y) with the current state q. An accepting computation of $\boldsymbol{M}$ on $\boldsymbol{x}$ is encoded by the word $w = \sharp \mathfrak{c}_1 \sharp \mathfrak{c}_2 \sharp \dots \sharp \mathfrak{c}_{N-1} \sharp \mathfrak{c}_N$ over the alphabet $\Sigma' = \Sigma \cup (Q \times \Sigma) \cup \{\sharp\}$, where $\mathfrak{c}_1, \mathfrak{c}_2, \dots, \mathfrak{c}_N$ are the subsequent configurations in the computation. In particular, $\mathfrak{c}_1$ is the initial configuration $(q_0, x_1)x_2 \dots x_n \mathsf{b} \dots \mathsf{b}$, and $\mathfrak{c}_N$ is the accepting configuration $(q_{acc}, \mathsf{b})\mathsf{b} \dots \mathsf{b}$. Thus, any accepting computation is encoded by a word of length N^2.

Call a tuple $\mathfrak{t} = (a, b, c, d, e, f) \in (\Sigma')^6$ *legal* [52, Theorem 7.37] if there exist two consecutive configurations $\mathfrak{c}_1$ and $\mathfrak{c}_2$ of $\boldsymbol{M}$ and a number i such that

$$abcdef = \mathfrak{c}_1[i]\mathfrak{c}_1[i+1]\mathfrak{c}_1[i+2]\mathfrak{c}_2[i]\mathfrak{c}_2[i+1]\mathfrak{c}_2[i+2].$$

Let $\mathfrak{L} \subseteq (\Sigma')^6$ be the set of all legal tuples (plus a few additional 6-tuples to take care of $\sharp$). Thus, a word w encodes an accepting computation iff it starts with the initial configuration preceded by $\sharp$, ends with the accepting configuration, and

$$(w[i], w[i+1], w[i+2], w[i+N], w[i+N+1], w[i+N+2]) \in \mathfrak{L},$$

for any $i \leq N^2 - N - 2$. Let $\bar{\mathfrak{L}} = (\Sigma')^6 \setminus \mathfrak{L}$.

For any $k > 0$, by a *k-counter* we mean a set $\mathbb{A} = \{A_j^i \mid i = 0, 1,\ j = 1, \dots, k\}$ of atomic concepts that will be used to store values between 0 and $2^k - 1$, which can be different at different time points. The counter $\mathbb{A}$ is *well-defined* at a time point $n \in \mathbb{N}$ in an interpretation $\mathcal{I}$

if $\mathcal{I}, n \models A_j^0 \wedge A_j^1 \to \bot$ and $\mathcal{I}, n \models A_j^0 \vee A_j^1$, for any $j = 1, \ldots, k$. In this case, the *value of* $\mathbb{A}$ at n in $\mathcal{I}$ is given by the unique binary number $b_k \ldots b_1$ for which $\mathcal{I}, n \models A_1^{b_1} \wedge \cdots \wedge A_k^{b_k}$. We require the following formulas, for $c = b_k \ldots b_1$ (provided that $\mathbb{A}$ is well-defined):

- $[\mathbb{A} = c] = A_1^{b_1} \wedge \cdots \wedge A_k^{b_k}$, for which $\mathcal{I}, n \models [\mathbb{A} = c]$ iff the value of $\mathbb{A}$ is c;
- $[\mathbb{A}< c] = \bigvee_{\substack{k \geq i \geq 1 \\ b_i = 1}} \big(A_i^0 \wedge \bigwedge_{j=i+1}^k A_j^{b_j}\big)$ with $\mathcal{I}, n \models [\mathbb{A} < c]$ iff the value of $\mathbb{A}$ is $< c$;
- $[\mathbb{A}> c] = \bigvee_{\substack{k \geq i \geq 1 \\ b_i = 0}} \big(A_i^1 \wedge \bigwedge_{j=i+1}^k A_j^{b_j}\big)$ with $\mathcal{I}, n \models [\mathbb{A} > c]$ iff the value of $\mathbb{A}$ is $> c$.

We regard the set $(\bigcirc_F \mathbb{A}) = \{\bigcirc_F A_j^i \mid i = 0, 1,\ j = 1, \ldots, k\}$ as another counter that stores at n in $\mathcal{I}$ the value stored by $\mathbb{A}$ at $n+1$ in $\mathcal{I}$. Thus, we can use formulas like $[\mathbb{A} > c_1] \to [(\bigcirc_F \mathbb{A}) = c_2]$, which says that if the value of $\mathbb{A}$ at n in $\mathcal{I}$ is greater than c_1, then the value of $\mathbb{A}$ at $n+1$ in $\mathcal{I}$ is c_2. Also, for $l \leq k$, we can use formulas like $[\mathbb{A} = i \,(\mathrm{mod}\, 2^l)]$ with self-explaining meaning.

To define $\mathcal{O}$ and $E = (E^+, E^-)$ for given $\boldsymbol{M}$ and $\boldsymbol{x} = x_1 \ldots x_n$, we fix $k = \lceil \log |\Sigma'| \rceil + 1$ and $m = \lceil 2k \log N \rceil + 1$, and let $\Sigma' = \{a_0, \ldots, a_l\}$. We use the following atomic concepts in $\mathcal{O}$ and E: the symbols in Σ', the atoms B, S, C, T, D, G, and F_{t}, for $\mathsf{t} \in \bar{\mathfrak{L}}$, and those atoms that are needed in m-counters $\mathbb{S}$, $\mathbb{C}$, $\mathbb{T}$, $\mathbb{D}$, $\mathbb{G}$, $\mathbb{F}_{\mathsf{t}}$.

We define $E = (E^+, E^-)$ by taking

- E^+ with the ABoxes $\{T(0)\}$, $\{S(0)\}$, and $\{C(0)\}$;
- E^- with the ABoxes $\{D(0)\}$, $\{G(0)\}$, and $\{F_{\mathsf{t}}(0)\}$, for all $\mathsf{t} \in \bar{\mathfrak{L}}$.

The following axioms, for $A \in \{S, C, T, D, G, F_{\mathsf{t}}\}$ and $\mathsf{t} \in \bar{\mathfrak{L}}$, serve to initialise the corresponding m-counters:

$$A \to [(\bigcirc \mathbb{A}) = 0].$$

(These and all other axioms of $\mathcal{O}$ can be easily transformed to equivalent sets of $\mathit{LTL}_{\mathit{horn}}^{\bigcirc}$ axiooms.) The behaviour of each counter is specified by the axioms below whose meaning is illustrated by the structure of the canonical model of the corresponding example restricted to B and Σ'.

The T-axioms

$$[\mathbb{T} < N^2] \to [\bigcirc \mathbb{T} = \mathbb{T} + 1], \quad [\mathbb{T} = 0] \to \sharp, \quad [\mathbb{T} = 1] \to (q_1, x_1),$$
$$[\mathbb{T} = 2] \to x_2, \ldots, [\mathbb{T} = n] \to x_n, \quad [\mathbb{T} > n] \wedge [\mathbb{T} < N] \to \mathsf{b}, \quad [\mathbb{T} = N] \to \sharp,$$
$$[\mathbb{T} > N] \wedge [\mathbb{T} < N^2 - N] \to X, \qquad \text{for all } X \in \Sigma',$$
$$[\mathbb{T} = N^2 - N] \to \sharp, \quad [\mathbb{T} = N^2 - N + 1] \to (q_{acc}, \mathsf{b}), \quad [\mathbb{T} > N^2 - N + 1] \to \mathsf{b}$$

together with the ABox $\{T(0)\}$ give rise to the canonical model of the form

$\underline{\mathcal{C}_{\mathcal{O},\{T(0)\}}}$: $\quad \emptyset, \sharp, (q_1, x_1), x_2, \ldots, x_n, \mathsf{b}^{N-n-1}, \sharp, (\Sigma')^{N^2-2N-1}, \sharp, (q_{acc}, \mathsf{b}), \mathsf{b}^{N-2}, \emptyset, \emptyset, \ldots.$

The D-axioms

$$[\mathbb{D} < N^2 - 1] \to [\bigcirc \mathbb{D} = \mathbb{D} + 1], \qquad [\mathbb{D} < N^2 - 1] \to X, \text{ for all } X \in \Sigma'$$

and $\{D(0)\}$ give the canonical model

$\underline{\mathcal{C}_{\mathcal{O},\{D(0)\}}}$: $\emptyset, (\Sigma')^{N^2-1}, \emptyset, \emptyset, \dots.$

The S-axioms

$$[\mathbb{S} < 2N^2] \to [\bigcirc\mathbb{S} = \mathbb{S}+1], \quad [\mathbb{S} < 2N^2] \wedge S_0^0 \to B, \quad [\mathbb{S} < 2N^2] \wedge S_0^1 \to X, \text{ for all } X \in \Sigma',$$

and $\{S(0)\}$ generate the canonical model

$\underline{\mathcal{C}_{\mathcal{O},\{S(0)\}}}$: $\emptyset, (B, \Sigma')^{N^2}, \emptyset, \emptyset, \dots,$

The G-axioms

$$[\mathbb{G} < 2N^2 - 2] \to [\bigcirc\mathbb{G} = \mathbb{S} + 1], \quad [\mathbb{G} < 2N^2 - 2] \wedge G_0^0 \to B,$$
$$[\mathbb{G} < 2N^2 - 2] \wedge G_0^1 \to X, \text{ for all } X \in \Sigma',$$

and $\{G(0)\}$ generate

$\underline{\mathcal{C}_{\mathcal{O},\{G(0)\}}}$: $\emptyset, (B, \Sigma')^{N^2-1}, \emptyset, \emptyset, \dots$

and the C-axioms

$$[\mathbb{C} < 2^k N^2] \to [\bigcirc\mathbb{C} = \mathbb{C}+1], \quad [\mathbb{C} < 2^k N^2] \to B, \quad [\mathbb{C} = i \,(\mathrm{mod}\, 2^k)] \to a_i, \text{ for all } i \in [0, l]$$

together with $\{C(0)\}$ generate

$\underline{\mathcal{C}_{\mathcal{O},\{C(0)\}}}$: $\emptyset, (a_0 B, a_1 B, \dots, a_l B, B^{2^k - l - 1})^{N^2}, \emptyset, \emptyset, \dots.$

Finally, the $F_{\mathfrak{t}}$-axioms, for $\mathfrak{t} = (a, b, c, d, e, f) \in \bar{\mathfrak{L}}$,

$$\begin{aligned}
&[\mathbb{F}_{\mathfrak{t}} < 2N^2] \to [\bigcirc\mathbb{F}_{\mathfrak{t}} = \mathbb{F}_{\mathfrak{t}} + 1],\\
&[\mathbb{F}_{\mathfrak{t}} < N^2 - N - 2] \to X, \qquad \text{for all } X \in \Sigma' \cup \{B\},\\
&[\mathbb{F}_{\mathfrak{t}} = N^2 - N - 2] \to a, \quad [\mathbb{F}_{\mathfrak{t}} = N^2 - N - 1] \to b, \quad [\mathbb{F}_{\mathfrak{t}} = N^2 - N] \to c,\\
&[\mathbb{F}_{\mathfrak{t}} > N^2 - N] \wedge [\mathbb{F}_{\mathfrak{t}} < N^2 - 2] \to X, \qquad \text{for all } X \in \Sigma',\\
&[\mathbb{F}_{\mathfrak{t}} = N^2 - 2] \to d, \quad [\mathbb{F}_{\mathfrak{t}} = N^2 - 1] \to e, \quad [\mathbb{F}_{\mathfrak{t}} = N^2] \to f,\\
&[\mathbb{F}_{\mathfrak{t}} > N^2] \to X, \qquad \text{for all } X \in \Sigma',
\end{aligned}$$

and the ABox $\{F_{\mathfrak{t}}(0)\}$ give the canonical model

$\underline{\mathcal{C}_{\mathcal{O},\{F_{\mathfrak{t}}(0)\}}}$: $\emptyset, (\Sigma' \cup \{B\})^{N^2-N-2}, a, b, c, (\Sigma')^{N-2}, d, e, f, (\Sigma')^{N^2}, \emptyset, \emptyset, \dots.$

We denote the set of the axioms above by $\mathcal{O}$ and show that E is separated by a $\mathcal{Q}_p[\mathsf{U}]$-query $\varkappa$ under $\mathcal{O}$ iff $\boldsymbol{M}$ accepts $\boldsymbol{x}$.

($\Leftarrow$) Suppose $\rho_1 \dots \rho_{N^2}$ encodes an accepting computation of $\boldsymbol{M}$ on $\boldsymbol{x}$. Consider the $\mathcal{Q}_p[\mathsf{U}]$-query

$$\varkappa = B \,\mathsf{U}\, (\rho_1 \wedge B \,\mathsf{U}\, (\rho_2 \dots (\rho_{N^2-1} \wedge (B \,\mathsf{U}\, \rho_{N^2})) \dots)).$$

It is not hard to show by inspecting the respective canonical models described above that

$$\mathcal{C}_{\mathcal{O},\{T(0)\}} \models \varkappa(0), \quad \mathcal{C}_{\mathcal{O},\{S(0)\}} \models \varkappa(0), \quad \mathcal{C}_{\mathcal{O},\{C(0)\}} \models \varkappa(0),$$
$$\mathcal{C}_{\mathcal{O},\{D(0)\}} \not\models \varkappa(0), \quad \mathcal{C}_{\mathcal{O},\{G(0)\}} \not\models \varkappa(0), \quad \mathcal{C}_{\mathcal{O},\{F_{\mathfrak{t}}(0)\}} \not\models \varkappa(0).$$

To prove the last one, notice that $\mathcal{C}_{\mathcal{O},\{F_{\mathfrak{t}}(0)\}} \not\models B(i)$ for all $i \geq N^2 - N - 2$. Therefore, if $\mathcal{C}_{\mathcal{O},\{F_{\mathfrak{t}}(0)\}} \models \varkappa(0)$, there is $i \leq N^2 - N - 2$ such that $\mathcal{C}_{\mathcal{O},\{F_{\mathfrak{t}}(0)\}} \models \rho_{i+j}(N^2 - N - 2 + j)$ for all $j \in [0, N^2 - i]$. But then $\rho_i \rho_{i+1} \rho_{i+2} = abc$ and, since $\rho_{i+N} \rho_{i+N+1} \rho_{i+N+2} \neq def$, we have $\mathcal{C}_{\mathcal{O},\{F_{\mathfrak{t}}(0)\}} \not\models \rho_{i+N}(N^2 - 2) \land \rho_{i+N+1}(N^2 - 1) \land \rho_{i+N+2}(N^2)$, which is a contradiction.

($\Rightarrow$) Suppose the query

$$\varkappa = \lambda_1 \,\mathsf{U}\, (\rho_1 \land \lambda_2 \,\mathsf{U}\, (\rho_2 \ldots (\rho_{K-1} \land (\lambda_K \,\mathsf{U}\, \rho_K)) \ldots))$$

with $\rho_K \neq \top$ separates E under $\mathcal{O}$. Since $\mathcal{C}_{\mathcal{O},\{T(0)\}} \models \varkappa(0)$, we have $K \leq N^2$ and $\rho_i \subseteq \Sigma'$ for all i. Since $\mathcal{C}_{\mathcal{O},\{D(0)\}} \not\models \varkappa(0)$, we have $K > N^2 - 1$, and so $K = N^2$. Since $\mathcal{C}_{\mathcal{O},\{C(0)\}} \models \varkappa(0)$ and $B \notin \Sigma'$, we have $|\rho_i| \leq 1$ for all i. Since $\mathcal{C}_{\mathcal{O},\{S(0)\}} \models \varkappa(0)$ and $\mathcal{C}_{\mathcal{O},\{G(0)\}} \not\models \varkappa(0)$, we have $|\rho_i| \geq 1$ and $\lambda_i \subseteq \{B\}$ for all i. So $|\rho_i| = 1$.

Suppose $\lambda_i = \top$ for some i. Let $y_j = j$ for $j \in [1, i-1]$, $y_j = N^2 + j - i + 1$ for $j \in [i, N^2]$, let $abcde = \rho_{N^2-N-2} \rho_{N^2-N-1} \rho_{N^2-N} \rho_{N^2-2} \rho_{N^2-1}$, and choose f so that $\mathfrak{t} = (a, b, c, d, e, f) \in \bar{\mathfrak{L}}$. Then $\mathcal{C}_{\mathcal{O},\{F_{\mathfrak{t}}(0)\}} \models \rho_j(y_j)$ for all j, and so $\mathcal{C}_{\mathcal{O},\{F_{\mathfrak{t}}(0)\}} \models \varkappa(0)$, which is a contradiction. Thus, $\lambda_i = B$ for all i.

Suppose $\rho_1 \ldots \rho_{N^2}$ does not encode an accepting computation of $\boldsymbol{M}$ on $\boldsymbol{x}$. In view of $\mathcal{C}_{\mathcal{O},\{T(0)\}} \models \varkappa(0)$, we have $\rho_1 \ldots \rho_{N+1} = \sharp(q_1, x_1) x_2 \ldots x_n \mathsf{b}^{N^2-N-1} \sharp$ and $\rho_{N^2-N} \ldots \rho_{N^2} = (\boldsymbol{q}_{acc}, \mathsf{b}) \mathsf{b}^{N-2}$, so there is some i such that $(\rho_i, \rho_{i+1}, \rho_{i+2}, \rho_{N+i}, \rho_{N+i+1}, \rho_{N+i+2}) \in \bar{\mathfrak{L}}$. Let $\mathfrak{t} = \rho_i \rho_{i+1} \rho_{i+2} \rho_{N+i} \rho_{N+i+1} \rho_{N+i+2}$. Let $y_j = j$ for $j \in [1, i-1]$ and $y_j = N^2 - N - 2 + j - i$ for $j \in [i, N^2]$. We have $\mathcal{C}_{\mathcal{O},\{F_{\mathfrak{t}}(0)\}} \models \rho_j(y_j)$, and so $\mathcal{C}_{\mathcal{O},\{F_{\mathfrak{t}}(0)\}} \models \varkappa(0)$, which is impossible. ❑

6. QBE($\mathcal{L}$, $\mathcal{Q}$[U])

To deal with arbitrary $\mathcal{Q}$[U]-queries (with possibly nested U-operators on the left-hand side of U), we require a few more technical definitions.

Let $\boldsymbol{\mathcal{M}} = \{\mathcal{I}_i \mid i \in I\}$ be a set of *LTL* interpretations and Δ an arena for $\boldsymbol{\mathcal{M}}$ (see the previous section). Let $\boldsymbol{d}^1, \boldsymbol{d}^2 \subseteq \Delta$ be nonempty. For any $d \in \boldsymbol{d}^1_i$, let $\mu_i(d) = \min\{d' \in \boldsymbol{d}^2_i \mid d <_i d'\}$. If μ_i is a surjective $\boldsymbol{d}^1_i \to \boldsymbol{d}^2_i$ function, for every $i \in \boldsymbol{I}$, we write $\boldsymbol{d}^1 \lessdot \boldsymbol{d}^2$ and set

$$\nabla(\boldsymbol{d}^1, \boldsymbol{d}^2) \;=\; \bigcup_{i \in \boldsymbol{I}} \bigcup_{d \in \boldsymbol{d}^1_i} \{d' \in \Delta \mid d <_i d' <_i \mu_i(d)\}.$$

We thereby generalise the definition of $\lessdot$ and ∇ from the previous section to possibly non-singleton $\boldsymbol{d}$. For singleton $\boldsymbol{d}$, the old and new definitions coincide.

Example 23. Let $\boldsymbol{\mathcal{M}}$ consist of one interpretation, $\Delta = \{1, 2, 3, 4\}$, $\boldsymbol{d}^1 = \{1, 2, 3\}$ and $\boldsymbol{d}^2 = \{3, 4\}$. Then $\boldsymbol{d}^1 \lessdot \boldsymbol{d}^2$ with $\nabla(\boldsymbol{d}^1, \boldsymbol{d}^2) = \{2\}$. However, for $\boldsymbol{d}^1 = \{1, 2\}$ and $\boldsymbol{d}^2 = \{3, 4\}$, we have neither $\boldsymbol{d}^1 \lessdot \boldsymbol{d}^2$ (because μ is not a surjection) nor $\boldsymbol{d}^2 \lessdot \boldsymbol{d}^1$ (because μ is not defined).

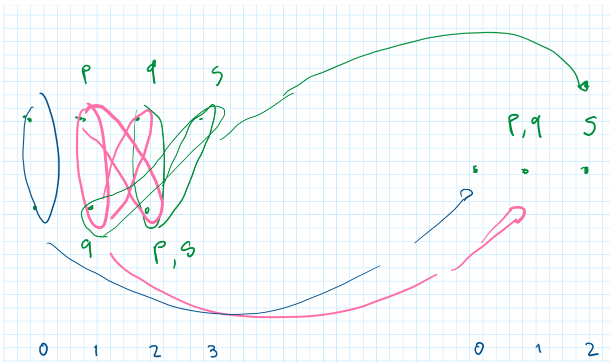
Given an arena Δ for $\boldsymbol{\mathcal{M}}$ and a set Σ of atoms, a $\Sigma\Delta$-U-*simulation* between $\boldsymbol{\mathcal{M}}$ and an *LTL* interpretation $\mathcal{J}$ is a binary relation $S \neq \emptyset$ between $\wp(\Delta) \setminus \{\emptyset\}$ and the domain $\Delta^{\mathcal{J}} = \mathbb{N}$ of $\mathcal{J}$ such that the following conditions hold (cf. [26]):

(at) if $(\boldsymbol{d}, e) \in S$ and $\boldsymbol{\mathcal{M}}, \boldsymbol{d} \models A$, then $\mathcal{J}, e \models A$, for every $A \in \Sigma$;

(nxt) if $(\boldsymbol{d}, e) \in S$ and $\boldsymbol{d} \lessdot \boldsymbol{d}' \in \Delta$, then there exists $e' > e$ such that $(\boldsymbol{d}', e') \in S$ and $(\nabla(\boldsymbol{d}, \boldsymbol{d}'), e'') \in S$, for every e'' with $e < e'' < e'$.

If $(\boldsymbol{d}, e) \in S$, we say that S is *via* $(\boldsymbol{d}, e)$.

Example 24. Suppose $\boldsymbol{\mathcal{M}}$ consists of the interpretations whose 'meaningful parts' are shown on the left-hand side of the picture below (with no atom being true at any of the omitted time points) and suppose the 'meaningful part' of an interpretation $\mathcal{J}$ looks like that on the right-hand side. Let $\Delta = \{0_1, 1_1, 2_1, 3_1, 0_2, 1_2, 2_2\}$ and $\Sigma = \{p, q, r\}$. We show how to



construct a $\Sigma\Delta$-U-simulation S between $\boldsymbol{\mathcal{M}}$ and $\mathcal{J}$ via $(\mathbf{0}, 0)$. First, we add $(\mathbf{0}, 0)$ to $S = \emptyset$ and consider all $\boldsymbol{d} \subseteq \Delta$ with $\mathbf{0} \lessdot \boldsymbol{d}$ and then all $\boldsymbol{d}' \subseteq \Delta$ with $\boldsymbol{d} \lessdot \boldsymbol{d}'$:

- $\mathbf{0} \lessdot \mathbf{1}$ with $\nabla(\mathbf{0}, \mathbf{1}) = \emptyset$, and we add $(\mathbf{1}, 1)$ to S;
- $\mathbf{0} \lessdot \{1_1, 2_2\}$ with $\nabla(\mathbf{0}, \{1_1, 2_2\}) = \{1_2\}$, and we add $(\{1_1, 2_2\}, 1)$ to S;
- $\mathbf{0} \lessdot \{2_1, 1_2\}$ with $\nabla(\mathbf{0}, \{2_1, 1_2\}) = \{1_1\}$, and we add $(\{2_1, 1_2\}, 1)$ to S;
- $\mathbf{0} \lessdot \{2_1, 1_2\}$ with $\nabla(\mathbf{0}, \{2_1, 1_2\}) = \{1_1\}$, and we add $(\{2_1, 1_2\}, 1)$ to S;
- $\mathbf{0} \lessdot \mathbf{2}$ with $\nabla(\mathbf{0}, \mathbf{2}) = \mathbf{1}$, and we add $(\mathbf{2}, 2)$ to S, with $(\mathbf{1}, 1)$ being in S already;
- $\mathbf{0} \lessdot \{3_1, 1_2\}$ with $\nabla(\mathbf{0}, \{3_1, 1_2\}) = \{1_1, 2_1, 1_2\}$, and we add both $(\{3_1, 1_2\}, 2)$ and $(\{1_1, 2_1\}, 1)$ to S;
- $\mathbf{0} \lessdot \{3_1, 2_2\}$ with $\nabla(\mathbf{0}, \{3_1, 2_2\}) = \{1_1, 2_1, 1_2\}$, and we add both $(\{3_1, 2_2\}, 2)$ and $(\{1_1, 2_1, 1_2\}, 1)$ to S;
- $\mathbf{1} \lessdot \mathbf{2}$ with $\nabla(\mathbf{1}, \mathbf{2}) = \emptyset$, and we already have $(\mathbf{2}, 2) \in S$;
- $\mathbf{1} \lessdot \{3_1, 2_2\}$ with $\nabla(\mathbf{1}, \{3_1, 2_2\}) = \{1_1\}$, and we already have $(\{3_1, 2_2\}, 2) \in S$;
- $\{1_1, 2_1\} \lessdot \{3_1\}$ with $\nabla(\{1_1, 2_1\}, \{3_1\}) = \{2_1\}$, so we add $(\{3_1\}, 2)$ to S;
- $\{2_1, 1_2\} \lessdot \{3_1, 2_2\}$ with $\nabla(\{2_1, 1_2\}, \{3_1, 2_2\}) = \{2_1\}$ and $(\{3_1, 2_2\}, 2) \in S$;
- $\{1_1, 2_1, 1_2\} \lessdot \{3_1, 2_2\}$ with $\nabla(\{1_1, 2_1, 1_2\}, \{3_1, 2_2\}) = \{2_1\}$, $(\{3_1, 2_2\}, 2) \in S$.

Theorem 25. *Let $E = (E^+, \{\mathcal{D}_N\})$ be an example set, $\mathcal{O}$ an LTL ontology, and Σ the set of atoms that occur in E and $\mathcal{O}$. Suppose $\boldsymbol{\mathcal{M}}_\mathcal{D}$, for each $\mathcal{D} \in E^+$, is a set of models of $\mathcal{O}$ and $\mathcal{D}$ such that, for all $\mathcal{Q}[\mathsf{U}]$-queries $\varkappa$ with atoms from Σ,*

$$\mathcal{O}, \mathcal{D} \models \varkappa(0) \qquad \textit{iff} \qquad \mathcal{I} \models \varkappa(0) \quad \textit{for all } \mathcal{I} \in \boldsymbol{\mathcal{M}}_\mathcal{D}. \tag{13}$$

Then the following conditions are equivalent:

(i) *there is no $\varkappa \in \mathcal{Q}[\mathsf{U}]$ separating E under $\mathcal{O}$*;

(ii) *for any model $\mathcal{J}$ of $\mathcal{O}$ and $\mathcal{D}_N$ and any arena Δ for $\boldsymbol{\mathcal{M}} = \bigcup_{\mathcal{D} \in E^+} \boldsymbol{\mathcal{M}}_\mathcal{D}$, there exists a $\Sigma\Delta$-U-simulation between $\boldsymbol{\mathcal{M}}$ and $\mathcal{J}$ via $(\mathbf{0}, 0)$.*

Proof. We require a few definitions. Let $\boldsymbol{\mathcal{M}} = \{\mathcal{I}_i \mid i \in \boldsymbol{I}\}$. For any $i \in \boldsymbol{I}$ and any $\varkappa = \varphi \,\mathsf{U}\, \psi$ with $\mathcal{I}_i, k \models \varkappa$, for some $k \in \mathbb{N}$, let $\mathsf{mw}_i(\varkappa, k)$ be the *minimal* $m > k$ witnessing $\mathcal{I}_i, k \models \varkappa$ in the sense that $\mathcal{I}_i, m \models \psi$ and $\mathcal{I}_i, l \models \varphi$ for all l with $k < l < m$. Note that if $k < k'$, $\mathcal{I}_i, k \models \varkappa$, $\mathcal{I}_i, k' \models \varkappa$ and $\mathsf{mw}_i(\varkappa, k) > k'$, then $\mathsf{mw}_i(\varkappa, k) = \mathsf{mw}_i(\varkappa, k')$. Now, for any $\varkappa \in \mathcal{Q}[\mathsf{U}]$, we define inductively a number $\mathsf{reach}_i(\varkappa, n)$ by taking

- $\mathsf{reach}_i(A, k) = k$, for any atom A,
- $\mathsf{reach}_i(\varphi \wedge \psi, k) = \max\{\mathsf{reach}_i(\varphi, k), \mathsf{reach}_i(\psi, k)\}$,
- $\mathsf{reach}_i(\varphi \,\mathsf{U}\, \psi, k) = \max\{\mathsf{reach}_i(\varphi, \mathsf{mw}_i(\varphi \,\mathsf{U}\, \psi, k) - 1), \mathsf{reach}_i(\psi, \mathsf{mw}_i(\varphi \,\mathsf{U}\, \psi, k))\}$.

Lemma 26. *Suppose $\varkappa \in \mathcal{Q}[\mathsf{U}]$, $\boldsymbol{d} \subseteq \bigcup_{i \in \boldsymbol{I}} \mathbb{N}_i$ with finite $\boldsymbol{d}_i$, and $\boldsymbol{\mathcal{M}}, \boldsymbol{d} \models \varkappa$. Let $\Delta^{\varkappa, \boldsymbol{d}} = \bigcup_{i \in \boldsymbol{I}} \bigcup_{d \in \boldsymbol{d}_i} \{k \in \mathbb{N} \mid d \le k \le \mathsf{reach}_i(\varkappa, d)\}$. If there exists a $\Sigma\Delta^{\varkappa, \boldsymbol{d}}$-$\mathsf{U}$-simulation S between $\boldsymbol{\mathcal{M}}$ and an interpretation $\mathcal{J}$ via $(\boldsymbol{d}, e)$, then $\mathcal{J}, e \models \varkappa$.*

Proof. The proof is by induction on the construction of $\varkappa$. The basis follows immediately from the definition, and the case $\varkappa = \varphi \wedge \psi$ follows from IH because S is also a $\Sigma\Delta^{\varphi, \boldsymbol{d}}$-$\mathsf{U}$- and $\Sigma\Delta^{\psi, \boldsymbol{d}}$-$\mathsf{U}$-simulation. So let $\varkappa = \varphi \,\mathsf{U}\, \psi$. For any $i \in \boldsymbol{I}$ and $d \in \boldsymbol{d}_i$, we have $\mathcal{I}_i, \mathsf{mw}_i(\varkappa, d) \models \psi$. Let $\boldsymbol{d}' = \bigcup_{i \in \boldsymbol{I}} \bigcup_{d \in \boldsymbol{d}_i} \{\mathsf{mw}_i(\varkappa, d)\} \subseteq \Delta^{\varkappa, \boldsymbol{d}}$. Since $\boldsymbol{d} \lessdot \boldsymbol{d}'$ because $\mu_i(d) = \mathsf{mw}_i(\varkappa, d)$, there is $e' > e$ such that $(\boldsymbol{d}', e') \in S$ and $(\nabla(\boldsymbol{d}, \boldsymbol{d}'), e'') \in S$, for every e'' with $e < e'' < e'$. By the definition, $\boldsymbol{\mathcal{M}}, \boldsymbol{d}' \models \psi$. It also follows from the definitions of simulation and the functions reach_i that the restriction of S to $\Delta^{\psi, \boldsymbol{d}'}$ is a $\Sigma\Delta^{\psi, \boldsymbol{d}'}$-$\mathsf{U}$-simulation between $\boldsymbol{\mathcal{M}}$ and $\mathcal{J}$ via $(\boldsymbol{d}', e')$. We also have $\mathcal{I}_i, d'' \models \varphi$ whenever $d < d'' < \mathsf{mw}_i(\varkappa, d)$, and so $\boldsymbol{\mathcal{M}}, \nabla(\boldsymbol{d}, \boldsymbol{d}') \models \varphi$. And we have a $\Sigma\Delta^{\varphi, \nabla(\boldsymbol{d}, \boldsymbol{d}')}$-$\mathsf{U}$-simulation between $\boldsymbol{\mathcal{M}}$ and $\mathcal{J}$ via $(\nabla(\boldsymbol{d}, \boldsymbol{d}'), e'')$, for every e'' with $e < e'' < e'$. By IH, $\mathcal{J}, e' \models \psi$ and $\mathcal{J}, e'' \models \varphi$ for all such e'', from which $\mathcal{J}, e \models \varkappa$. ❑

We now proceed with the proof of our theorem.

$(ii) \Rightarrow (i)$ Suppose on the contrary that $\varkappa \in \mathcal{Q}[\mathsf{U}]$ separates E under $\mathcal{O}$. Then $\mathcal{I}_i, 0_i \models \varkappa$ for all $I_i \in \boldsymbol{\mathcal{M}}$ and there is a model $\mathcal{J}$ of $\mathcal{O}$ and $\mathcal{D}_N$ with $\mathcal{J}, 0 \not\models \varkappa$. Now, take a $\Sigma\Delta^{\varkappa, \boldsymbol{n}}$-$\mathsf{U}$-simulation S between $\boldsymbol{\mathcal{M}}$ and $\mathcal{J}$ via $(\mathbf{0}, 0)$. By Lemma 26, we then have $\mathcal{J}, 0 \models \varkappa$, which is impossible.

$(i) \Rightarrow (ii)$ Suppose we have a model $\mathcal{J}$ of $\mathcal{O}$ and $\mathcal{D}_N$ and an arena Δ for $\boldsymbol{\mathcal{M}}$ such that there is no $\Sigma\Delta$-U-simulation between $\boldsymbol{\mathcal{M}}$ and $\mathcal{J}$ via $(\mathbf{0}, 0)$. We construct a query $\varkappa \in \mathcal{Q}[\mathsf{U}]$ that separates E under $\mathcal{O}$ as follows. For any $\boldsymbol{d} \subseteq \Delta$, we set inductively, using the fact that $\Delta \cap \mathbb{N}_i$ is finite, and so any chain $\boldsymbol{d} \lessdot \boldsymbol{d}^1 \lessdot \dots \lessdot \boldsymbol{d}^l \lessdot \dots$ with the $\boldsymbol{d}^j \subseteq \Delta$ is finite, too:

$$\varphi_{\boldsymbol{d}} \;=\; \bigwedge_{\substack{A \in \Sigma \\ \boldsymbol{\mathcal{M}}, \boldsymbol{d} \models A}} A \;\wedge\; \bigwedge_{\substack{\boldsymbol{d}' \subseteq \Delta \\ \boldsymbol{d} \lessdot \boldsymbol{d}'}} \varphi_{\nabla(\boldsymbol{d},\boldsymbol{d}')} \,\mathsf{U}\, \varphi_{\boldsymbol{d}'}.$$

Aiming to show that $\varkappa = \varphi_{\boldsymbol{0}}$ separates E under $\mathcal{O}$, we first prove by induction that $\mathcal{I}_i, \boldsymbol{d}_i \models \varphi_{\boldsymbol{d}}$, for any $\boldsymbol{d} \subseteq \Delta$. The basis of induction is the obvious case when there is no $\boldsymbol{d}' \subseteq \Delta$ with $\boldsymbol{d} \lessdot \boldsymbol{d}'$. Suppose now that our statement holds for all subformulas $\varphi_{\boldsymbol{d}^1}$ of $\varphi_{\boldsymbol{d}}$ and show it for $\varphi_{\boldsymbol{d}}$. To prove that $\mathcal{I}_i, \boldsymbol{d}_i \models \varphi_{\nabla(\boldsymbol{d},\boldsymbol{d}')} \,\mathsf{U}\, \varphi_{\boldsymbol{d}'}$ for each $\boldsymbol{d}' \subseteq \Delta$ with $\boldsymbol{d} \lessdot \boldsymbol{d}'$, consider any $d \in \boldsymbol{d}_i$. Since $\mu_i(d) \in \boldsymbol{d}'_i$, the IH yields $\mathcal{I}_i, \mu_i(d) \models \varphi_{\boldsymbol{d}'}$. Now, take any c with $d <_i c <_i \mu_i(d)$. Then $c \in \nabla(\boldsymbol{d}, \boldsymbol{d}')_i$ and, by IH, $\mathcal{I}_i, c \models \varphi_{\nabla(\boldsymbol{d},\boldsymbol{d}')}$. Thus, we obtain $\mathcal{I}_i, \boldsymbol{d}_i \models \varphi_{\nabla(\boldsymbol{d},\boldsymbol{d}')} \,\mathsf{U}\, \varphi_{\boldsymbol{d}'}$. It follows that $\mathcal{I}_i, 0_i \models \varphi_{\boldsymbol{0}}$, as required.

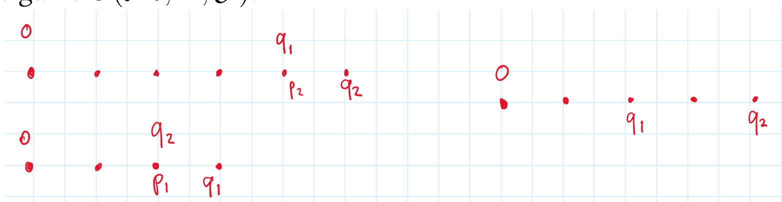
It remains to show that $\mathcal{J}, 0 \not\models \varphi_{\boldsymbol{0}}$. Suppose otherwise and construct a $\Sigma\Delta$-U-simulation S between $\boldsymbol{\mathcal{M}}$ and $\mathcal{J}$ via $(\boldsymbol{0}, 0)$, leading to a contradiction. We start by taking $S_0 = \{(\boldsymbol{0}, 0)\}$. Condition **(at)** holds for the pair in S_0 because $\mathcal{J}, 0 \models A$ for all $A \in \Sigma$ with $\boldsymbol{\mathcal{M}}, \boldsymbol{0} \models A$. On the other hand, for any $\boldsymbol{d}' \subseteq \Delta$ with $\boldsymbol{0} \lessdot \boldsymbol{d}'$, there is $n_{\boldsymbol{d}'} > 0$ such that $\mathcal{J}, n_{\boldsymbol{d}'} \models \varphi_{\boldsymbol{d}'}$ and $\mathcal{J}, n_{\nabla(\boldsymbol{0},\boldsymbol{d}')} \models \varphi_{\nabla(\boldsymbol{0},\boldsymbol{d}')}$ for all $n_{\nabla(\boldsymbol{0},\boldsymbol{d}')}$ with $0 < n_{\nabla(\boldsymbol{0},\boldsymbol{d}')} < n_{\boldsymbol{d}'}$. We extend S_0 by adding to it $(\boldsymbol{d}', n_{\boldsymbol{d}'})$, for each such $\boldsymbol{d}'$, together with all the pairs $(\nabla(\boldsymbol{0}, \boldsymbol{d}'), n_{\nabla(\boldsymbol{0},\boldsymbol{d}')})$, and denote the result by S_1. Condition **(at)** holds for all $(\boldsymbol{d}^1, n_{\boldsymbol{d}^1}) \in S_1$ in view of $\mathcal{J}, n_{\boldsymbol{d}^1} \models \varphi_{\boldsymbol{d}^1}$. To satisfy **(nxt)** for the newly added pairs, we use the second conjunct of $\varphi_{\boldsymbol{d}^1}$ and extend S_1 to S_2 in the same way as we did it for S_0. We proceed in this way till there are no $(\boldsymbol{d}, e) \in S_j \setminus S_{j-1}$ with $\boldsymbol{d} \lessdot \boldsymbol{d}'$ and $\boldsymbol{d}' \subseteq \Delta$, which must be the case for some j in view of the finiteness of the Δ_i and the definition of $\lessdot$. The resulting $S = S_j$ is the required $\Sigma\Delta$-U-simulation between $\boldsymbol{\mathcal{M}}$ and $\mathcal{J}$ via $(\boldsymbol{0}, 0)$. ❑

To understand the complexity of checking the existence of simulations, we re-define them in game-theoretic terms. We begin by giving a brief abstract description of the games we need. Every game G is played by two players, Player 1 and Player 2, and defined by a set $\mathfrak{S}$ of states, a set $\mathfrak{C}$ of challenges, and two functions $\chi\colon \mathfrak{S} \to 2^{\mathfrak{C}}$ and $\rho\colon \mathfrak{S} \times \mathfrak{C} \to 2^{\mathfrak{S}}$, where $\chi(\mathfrak{s})$ is the set of challenges Player 1 can choose from in any state $\mathfrak{s}$ and $\rho(\mathfrak{s}, \mathfrak{c})$ is the set of responses available to Player 2 in order to respond to any challenge $\mathfrak{c}$ made by Player 1 in state $\mathfrak{s}$. The game starts in an initial state $\mathfrak{s}_0 \in \mathfrak{S}$ and is played in rounds. In each round i, for $i > 0$, the current state is $\mathfrak{s}_{i-1} \in \mathfrak{S}$. If $\chi(\mathfrak{s}_{i-1}) = \emptyset$, then Player 1 loses. Otherwise, Player 1 challenges Player 2 by choosing $\mathfrak{c}_i \in \chi(\mathfrak{s}_{i-1})$. If $\rho(\mathfrak{s}_{i-1}, \mathfrak{c}_i) = \emptyset$, then Player 2 loses. Otherwise, Player 1 responds with $\mathfrak{s}_i \in \rho(\mathfrak{s}_{i-1}, \mathfrak{c}_i)$, which becomes the current state for the next round $i+1$. A play of length n starting from $\mathfrak{s}_0 \in \mathfrak{S}$ is any sequence $\mathfrak{s}_0, \dots, \mathfrak{s}_n$ of states obtained as described above. For any ordinal $\lambda \leq \omega$, we say that Player 2 has a λ-*winning strategy* in the game G starting from $\mathfrak{s}_0$ if, for any play $\mathfrak{s}_0, \dots, \mathfrak{s}_n$ with $n < \lambda$ that is played according to this strategy, Player 2 has a response to any challenge of Player 1 in the final state $\mathfrak{s}_n$.

Let $\boldsymbol{\mathcal{M}} = \{\mathcal{I}_i \mid i \in \boldsymbol{I}\}$ be a set of *LTL* interpretations with $\boldsymbol{0} = \bigcup_{i \in \boldsymbol{I}} \{0_i\}$, Δ an arena for $\boldsymbol{\mathcal{M}}$, and $\mathcal{J}$ an *LTL* interpretation. We define a game $G(\boldsymbol{\mathcal{M}}, \Delta, \mathcal{J})$ in the table below, where, intuitively, Player 1 aims to show that there is no $\Sigma\Delta$-U-simulation between $\boldsymbol{\mathcal{M}}$ and $\mathcal{J}$, while Player 2 wants to show that such a simulation exists.

Game $G(\mathcal{M}, \Delta, \mathcal{J})$	
states	$(\mathbf{0}, 0)$ and $(\boldsymbol{d}^1, \boldsymbol{d}^2, e^1, e^2)$, where $\boldsymbol{d}^1$ and $\boldsymbol{d}^2$ are non-empty in Δ with $\boldsymbol{d}^1 \lessdot \boldsymbol{d}^2$, $e^1, e^2 \in \mathbb{N}$ with $e^1 < e^2$, and $\mathcal{M}, \boldsymbol{d}^j \models A$ implies $\mathcal{J}, e^j \models A$, for all $A \in \Sigma$ and $j = 1, 2$
initial state	$(\mathbf{0}, 0)$
challenges, $i = 0$	$(\mathbf{0}, 0) \to (\mathbf{0}, \boldsymbol{d}^2, 0)$ with $\mathbf{0} \lessdot \boldsymbol{d}^2$ (type A)
challenges, $i > 0$	$(\boldsymbol{d}^1, \boldsymbol{d}^2, e^1, e^2) \to (\boldsymbol{d}^2, \boldsymbol{d}^3, e^2)$ with $\boldsymbol{d}^2 \lessdot \boldsymbol{d}^3$ (type A) $(\boldsymbol{d}^1, \boldsymbol{d}^2, e^1, e^2) \to (\nabla(\boldsymbol{d}^1, \boldsymbol{d}^2), e^3)$ with $e^1 < e^3 < e^2$ (type B)
responses	$(\boldsymbol{d}^1, \boldsymbol{d}^2, e^1) \to (\boldsymbol{d}^1, \boldsymbol{d}^2, e^1, e^2)$ provided that $\mathcal{M}, \nabla(\boldsymbol{d}^1, \boldsymbol{d}^2) \models A$ implies $\mathcal{J}, e^3 \models A$ for all e^3 with $e^1 < e^3 < e^2$ and $A \in \Sigma \cup \{\bot\}$

Example 27. Suppose $\mathcal{M}$ consists of two interpretations whose 'meaningful parts' are shown on the left-hand side of the picture below and suppose the 'meaningful part' of an interpretation $\mathcal{J}$ looks like the one on the right-hand side of the picture. Let Δ be the set of points in $\mathcal{M}$ shown in the picture and $\Sigma = \{p_1, p_2, q_1, q_2\}$. We claim that player 2 does not have a winning strategy in the game $G(\mathcal{M}, \Delta, \mathcal{J})$.



Indeed, let Player 1 start with the challenge $(\{0_1, 0_2\}, 0) \to (\{0_1, 0_2\}, \{3_1, 1_2\}, 0)$. Suppose Player 2's strategy is to respond with the node $(\{0_1, 0_2\}, \{3_1, 1_2\}, 0, 1)$, for which $\nabla(\{0_1, 0_2\}, \{3_1, 1_2\}) = \{1_1, 2_1\}$. Then Player 1 can challenge with the node $(\{3_1, 1_2\}, \{5_1, 2_2\}, 1)$ to which Player 2 does not have a response because the only choice of $(\{3_1, 1_2\}, \{5_1, 2_2\}, 1, 4)$ does not satisfy the extra condition for p_1, q_2 that are true at $\nabla(\{3_1, 1_2\}, \{5_1, 2_2\}) = \{4_1\}$ in $\mathcal{M}$ but not at 2 and 3 in $\mathcal{J}$. Suppose Player 2 responds to the first challenge with $(\{0_1, 0_2\}, \{3_1, 1_2\}, 0, 2)$. Then Player 1 can challenge with $(\{3_1, 1_2\}, \{4_1, 3_2\}, 2)$ to which Player 2 has no good response. The case when Player 2 responds to the first challenge with $(\{0_1, 0_2\}, \{3_1, 1_2\}, 0, 3)$ is similar.

Theorem 28. *Let $\mathcal{M}$ be a set of LTL models and $\mathcal{J}$ a model. Then, for every arena Δ for $\mathcal{M}$, there exists a $\Sigma\Delta$-U-simulation between $\mathcal{M}$ and $\mathcal{J}$ via $(\mathbf{0}, 0)$ iff Player 2 has an ω-winning strategy for $G(\mathcal{M}, \Delta, \mathcal{J})$.*

Proof. ($\Rightarrow$) Suppose S is a $\Sigma\Delta$-U-simulation between $\mathcal{M}$ and $\mathcal{J}$ via $(\mathbf{0}, 0)$. We describe how to construct a winning strategy for Player 2. For any Player 1's challenge $(\mathbf{0}, \boldsymbol{d}', 0)$, Player 2 responds with any $(\mathbf{0}, \boldsymbol{d}', 0, e')$ such that $(\boldsymbol{d}', e') \in S$ and also $(\nabla(\boldsymbol{d}, \boldsymbol{d}'), e'') \in S$, for every e'' with $e < e'' < e'$. Clearly, such e' exists by the definition of U-simulation. Now, for the node $(\mathbf{0}, \boldsymbol{d}', 0, e')$ of Player 2, Player 1's move can be to $(\boldsymbol{d}', \boldsymbol{d}'', e')$ or to $(\nabla(\boldsymbol{d}, \boldsymbol{d}'), \boldsymbol{d}'', e'')$. In the former case, Player 2 responds with $(\boldsymbol{d}', \boldsymbol{d}'', e', e'')$, such that $(\boldsymbol{d}'', e'') \in S$ (which exists by the definition of U-simulation). In the latter case, Player 2 responds with $(\boldsymbol{d}'', e''') \in S$, for the appropriate e'''. We proceed further in the same way obtaining a winning strategy for Player 2 because there is an outgoing edge in each of their nodes. The proof ($\Leftarrow$) is similar. ❑

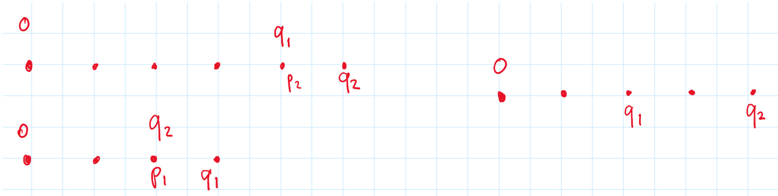
Theorem 29. $\mathsf{QBE}(\mathcal{Q}[\mathsf{U}])$ *is in PSpace.*

Proof. It suffices to consider example sets of the form $E = (E^+, \{\mathcal{D}_N\})$. Let Σ consist of the atoms that occur in E. Take $\boldsymbol{\mathcal{M}} = \{\mathcal{C}_{\emptyset,\mathcal{D}} \mid \mathcal{D} \in E^+\}$ (assuming that different $\mathcal{C}_{\emptyset,\mathcal{D}}$ have disjoint domains) and let $\mathcal{J} = \mathcal{C}_{\emptyset,\mathcal{D}_N}$. By Theorems 25 and 28, we need to show that we can check in PSpace if Player 2 has a winning strategy in $G(\boldsymbol{\mathcal{M}}, \Delta, \mathcal{J})$, for any arena Δ for $\boldsymbol{\mathcal{M}}$. First, we show that this is the case iff Player 2 has a winning strategy in $G(\boldsymbol{\mathcal{M}}, \Delta^*, \mathcal{J})$, where $\Delta^* = \bigcup_{i\in I}\{k_i \mid 0 \le k \le \max \mathcal{D}_i\}$. Indeed, suppose Player 2 has a winning strategy in $G(\boldsymbol{\mathcal{M}}, \Delta^*, \mathcal{J})$. It follows that Player 2 has a winning strategy in $G(\boldsymbol{\mathcal{M}}, \Delta, \mathcal{J})$, for each $\Delta \subseteq \Delta^*$, as the restriction of the Δ^*-winning strategy to Δ is a Δ-winning strategy. Now, consider $\Delta \supsetneq \Delta^*$ and show how to construct a Δ-winning strategy. It is enough to explain Player 2's responses from the nodes of the form $(\boldsymbol{d}, \boldsymbol{d}', e)$ with $\boldsymbol{d}' \cap (\Delta \setminus \Delta^*) \neq \emptyset$. We observe that $\boldsymbol{\mathcal{M}}, \boldsymbol{d}' \not\models A$, for all such $\boldsymbol{d}'$ and all atoms A. Therefore, Player 2 can respond with $(\boldsymbol{d}, \boldsymbol{d}', e, e+1)$. This strategy is clearly winning for Player 2 as there is an outgoing edge in each of their nodes.

Let $\max \Delta = \max_{i\in I} \Delta_i$. We show that Player 2 has a winning strategy in $G(\boldsymbol{\mathcal{M}}, \Delta, \mathcal{J})$ iff Player 2 has a winning strategy that only uses nodes with $e, e' \le \max \mathcal{D}_N + \max \Delta$. Indeed, if a Δ-winning strategy of Player 2 at $(\boldsymbol{0}, \boldsymbol{d}', 0)$ is to respond with $(\boldsymbol{0}, \boldsymbol{d}', 0, e')$, for $e > \max \mathcal{D}_N + \max \Delta$, we could change the response to $(\boldsymbol{0}, \boldsymbol{d}', 0, \mathcal{D}_N + \max \Delta)$ (in the sense that if the play after $(\boldsymbol{0}, \boldsymbol{d}', 0, e')$ was winning for Player 2, the play after the updated response will be winning as well). Further, if a Δ-winning strategy of Player 2 at $(\boldsymbol{d}, \boldsymbol{d}', e)$, for $e \ge \max \mathcal{D}_N + \max \Delta$, is to respond with $(\boldsymbol{d}, \boldsymbol{d}', e, e')$, for $e' > e+1$, then we could change the response of Player 2 to $(\boldsymbol{d}, \boldsymbol{d}', e, e+1)$.

Now we show how to check in PSpace the existence of Player 2's winning strategy in $G(\boldsymbol{\mathcal{M}}, \Delta^*, \mathcal{J})$ only using nodes with $e, e' \le \max \mathcal{D}_N + \max \Delta^*$. Since Δ^* is of polynomial size in E, all the required nodes in the graph of $G(\boldsymbol{\mathcal{M}}, \Delta^*, \mathcal{J})$ can be stored in PSpace. We observe that, in each play, the node $(\boldsymbol{n}, n)$ changes to the node $(\boldsymbol{n}, \boldsymbol{d}', n)$ with $\boldsymbol{n} \lessdot \boldsymbol{d}'$ and each node $(\boldsymbol{d}, \boldsymbol{d}', e, e')$ changes to a node of the form $(\boldsymbol{f}, \boldsymbol{d}'', e'')$, where $\boldsymbol{d}' \lessdot \boldsymbol{d}''$. Therefore, the required winning strategy of Player 2 exists iff it exists for the plays of the length up to $2 \max \Delta^*$, which is polynomial in E. Therefore, we can check the existence of the required winning strategy in PSpace. ❑

Example 30. Suppose $\boldsymbol{\mathcal{M}}$ comprises two interpretations whose 'meaningful parts' are shown on the left-hand side of the picture below and suppose the 'meaningful part' of an interpretation $\mathcal{J}$ looks like on the right-hand side of the picture. We claim that Player 2 does not have a winning strategy in the game $G(\boldsymbol{\mathcal{M}}, \mathcal{J}, \{0_1, 0_2\}, 0)$. Indeed, consider the plays that start with



the edge $(\{0_1, 0_2\}, 0) \rightarrow (\{0_1, 0_2\}, \{3_1, 1_2\}, 0)$. Suppose Player 2's strategy is to continue with $(\{0_1, 0_2\}, \{3_1, 1_2\}, 0, 1)$, which has an edge to $(\{3_1, 1_2\}, \{5_1, 2_2\}, 1)$. There are three outgoing edges from that node: one to $(\{3_1, 1_2\}, \{5_1, 2_2\}, 1, 4)$, for which we have $\nabla(\{3_1, 1_2\}, \{5_1, 2_2\}) = \{4_1\}$, another to $(\{4_1\}, 2)$ and the third one to $(\{4_1\}, 3)$. Suppose Player 2 responds with $(\{0_1, 0_2\}, \{3_1, 1_2\}, 0, 2)$. Then Player 1 moves to $(\{3_1, 1_2\}, \{4_1, 3_2\}, 2)$, where the possible responses of Player 2 are $(\{3_1, 1_2\}, \{4_1, 3_2\}, 2, 3)$ and $(\{3_1, 1_2\}, \{4_1, 3_2\}, 2, 4)$, and both of such plays are wins of Player 1. The case when Player 2 responds with $(\{0_1, 0_2\}, \{3_1, 1_2\}, 0, 3)$ is similar. Finally, we note that $\mathcal{M}$ and $\mathcal{J}$ can be separated by the query $\Diamond\big((p_1 \mathbin{\mathsf{U}} q_1) \land (p_2 \mathbin{\mathsf{U}} q_2)\big)$.

Acknowledgments

This research was supported by the EPSRC UK grants EP/S032207 and EP/S032282. for the joint project 'quant^{MD}: Ontology-Based Management for Many-Dimensional Quantitative Data'.

References

[1] D. M. L. Martins, Reverse engineering database queries from examples: State-of-the-art, challenges, and research opportunities, Inf. Syst. 83 (2019) 89–100.

[2] M. Zhang, H. Elmeleegy, C. M. Procopiuc, D. Srivastava, Reverse engineering complex join queries, in: Proc. of SIGMOD, 2013, pp. 809–820.

[3] Y. Y. Weiss, S. Cohen, Reverse engineering SPJ-queries from examples, in: Proc. of PODS, 2017, pp. 151–166.

[4] D. V. Kalashnikov, L. V. Lakshmanan, D. Srivastava, Fastqre: Fast query reverse engineering, in: Proc. of SIGMOD, 2018, pp. 337–350.

[5] D. Deutch, A. Gilad, Reverse-engineering conjunctive queries from provenance examples., in: Proc. of EDBT, 2019, pp. 277–288.

[6] S. Staworko, P. Wieczorek, Learning twig and path queries, in: Proc. of ICDT, 2012, pp. 140–154.

[7] P. Barceló, M. Romero, The complexity of reverse engineering problems for conjunctive queries, in: Proc. of ICDT, 2017, pp. 7:1–7:17.

[8] S. Cohen, Y. Y. Weiss, The complexity of learning tree patterns from example graphs, ACM Trans. Database Syst. 41 (2016) 14:1–14:44. URL: https://doi.org/10.1145/2890492. doi:10.1145/2890492.

[9] M. Arenas, G. I. Diaz, E. V. Kostylev, Reverse engineering SPARQL queries, in: Proc. of WWW, 2016, pp. 239–249.

[10] V. Gutiérrez-Basulto, J. C. Jung, L. Sabellek, Reverse engineering queries in ontology-

enriched systems: The case of expressive Horn description logic ontologies, in: Proc. of IJCAI-ECAI, 2018.

[11] M. Ortiz, Ontology-mediated queries from examples: a glimpse at the DL-Lite case, in: Proc. of GCAI, 2019, pp. 1–14.

[12] J. C. Jung, C. Lutz, H. Pulcini, F. Wolter, Logical separability of incomplete data under ontologies, in: Proc. of KR, 2020.

[13] G. Cima, F. Croce, M. Lenzerini, Query definability and its approximations in ontology-based data management, in: G. Demartini, G. Zuccon, J. S. Culpepper, Z. Huang, H. Tong (Eds.), CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021, ACM, 2021, pp. 271–280. URL: https://doi.org/10.1145/3459637.3482466. doi:10.1145/3459637.3482466.

[14] J. C. Jung, C. Lutz, H. Pulcini, F. Wolter, Separating data examples by description logic concepts with restricted signatures, in: Proc. of KR, 2021.

[15] R. Willard, Testing expressibility is hard, in: D. Cohen (Ed.), Principles and Practice of Constraint Programming - CP 2010 - 16th International Conference, CP 2010, St. Andrews, Scotland, UK, September 6-10, 2010. Proceedings, volume 6308 of *Lecture Notes in Computer Science*, Springer, 2010, pp. 9–23. URL: https://doi.org/10.1007/978-3-642-15396-9_4. doi:10.1007/978-3-642-15396-9_4.

[16] B. ten Cate, V. Dalmau, The product homomorphism problem and applications, in: Proc. of ICDT, 2015, pp. 161–176.

[17] M. Funk, J. C. Jung, C. Lutz, H. Pulcini, F. Wolter, Learning description logic concepts: When can positive and negative examples be separated?, in: Proc. of IJCAI, 2019, pp. 1682–1688.

[18] S. Demri, V. Goranko, M. Lange, Temporal Logics in Computer Science, Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, 2016.

[19] C. Lemieux, D. Park, I. Beschastnikh, General LTL specification mining (T), in: Proc. of ASE, IEEE, 2015, pp. 81–92.

[20] D. Neider, I. Gavran, Learning linear temporal properties, in: Proc. of FMCAD 2018, IEEE, 2018, pp. 1–10. URL: https://doi.org/10.23919/FMCAD.2018.8603016. doi:10.23919/FMCAD.2018.8603016.

[21] A. Camacho, S. A. McIlraith, Learning interpretable models expressed in linear temporal logic, in: Proc. of ICAPS 2018, AAAI Press, 2019, pp. 621–630. URL: https://aaai.org/ojs/index.php/ICAPS/article/view/3529.

[22] N. Fijalkow, G. Lagarde, The complexity of learning linear temporal formulas from examples, in: J. Chandlee, R. Eyraud, J. Heinz, A. Jardine, M. Zaanen (Eds.), Proceedings of the 15th International Conference on Grammatical Inference, 23-27 August 2021, Virtual Event, volume 153 of *Proceedings of Machine Learning Research*, PMLR, 2021, pp. 237–250. URL: https://proceedings.mlr.press/v153/fijalkow21a.html.

[23] R. Raha, R. Roy, N. Fijalkow, D. Neider, Scalable anytime algorithms for learning fragments of linear temporal logic, in: D. Fisman, G. Rosu (Eds.), Tools and Algorithms for the Construction and Analysis of Systems - 28th International Conference, TACAS 2022, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2022, Munich, Germany, April 2-7, 2022, Proceedings, Part I, volume 13243 of

Lecture Notes in Computer Science, Springer, 2022, pp. 263–280. URL: https://doi.org/10.1007/978-3-030-99524-9_14. doi:10.1007/978-3-030-99524-9_14.
[24] M. Fortin, B. Konev, V. Ryzhikov, Y. Savateev, F. Wolter, M. Zakharyaschev, Unique characterisability and learnability of temporal instance queries, in: Proc. of KR, 2022.
[25] D. Maier, The complexity of some problems on subsequences and supersequences, J. ACM 25 (1978) 322–336. URL: https://doi.org/10.1145/322063.322075. doi:10.1145/322063.322075.
[26] N. Kurtonina, M. de Rijke, Bisimulations for temporal logic, J. Log. Lang. Inf. 6 (1997) 403–425. URL: https://doi.org/10.1023/A:1008223921944. doi:10.1023/A:1008223921944.
[27] A. Prior, Time and Modality, Oxford University Press, 1956.
[28] H. Ono, A. Nakamura, On the size of refutation Kripke models for some linear modal and tense logics, Studia Logica (1980) 325–333.
[29] J. P. Burgess, Basic tense logic, in: Handbook of Philosophical Logic: Volume II: Extensions of Classical Logic, Reidel, Dordrecht, 1984, pp. 89–133.
[30] M. Y. Vardi, From Church and Prior to PSL, in: 25 Years of Model Checking - History, Achievements, Perspectives, volume 5000 of *LNCS*, Springer, 2008, pp. 150–171.
[31] O. Kupferman, M. Y. Vardi, Verification of fair transisiton systems, in: R. Alur, T. A. Henzinger (Eds.), Computer Aided Verification, 8th International Conference, CAV '96, New Brunswick, NJ, USA, July 31 - August 3, 1996, Proceedings, volume 1102 of *Lecture Notes in Computer Science*, Springer, 1996, pp. 372–382. URL: https://doi.org/10.1007/3-540-61474-5_84. doi:10.1007/3-540-61474-5_84.
[32] L. Badea, S. Nienhuys-Cheng, A refinement operator for description logics, in: Proc. of ILP, 2000, pp. 40–59.
[33] J. Lehmann, C. Haase, Ideal downward refinement in the $\mathcal{EL}$ description logic, in: Proc. of ILP, 2009, pp. 73–87.
[34] J. Lehmann, P. Hitzler, Concept learning in description logics using refinement operators, Machine Learning 78 (2010) 203–250.
[35] F. A. Lisi, U. Straccia, Learning in description logics with fuzzy concrete domains, Fundamenta Informaticae 140 (2015) 373–391.
[36] M. K. Sarker, P. Hitzler, Efficient concept induction for description logics, in: Proc. of AAAI, 2019, pp. 3036–3043.
[37] N. Fanizzi, C. d'Amato, F. Esposito, DL-FOIL concept learning in description logics, in: Proc. of ILP, 2008, pp. 107–121.
[38] F. A. Lisi, A formal characterization of concept learning in description logics, in: Proc. of DL, 2012.
[39] G. Rizzo, N. Fanizzi, C. d'Amato, Class expression induction as concept space exploration: From dl-foil to dl-focl, Future Gener. Comput. Syst. 108 (2020) 256–272.
[40] L. Bühmann, J. Lehmann, P. Westphal, S. Bin, DL-learner - structured machine learning on semantic web data, in: Proc. of WWW, 2018, pp. 467–471.
[41] L. Bühmann, J. Lehmann, P. Westphal, DL-learner - A framework for inductive learning on the semantic web, J. Web Sem. 39 (2016) 15–24.
[42] N. Fanizzi, G. Rizzo, C. d'Amato, F. Esposito, DLFoil: Class expression learning revisited, in: Proc. of EKAW, 2018, pp. 98–113.
[43] G. Rizzo, N. Fanizzi, C. d'Amato, F. Esposito, A framework for tackling myopia in concept

learning on the web of data, in: Proc. of EKAW, Springer, 2018, pp. 338–354.

[44] A. C. Tran, J. Dietrich, H. W. Guesgen, S. Marsland, Parallel symmetric class expression learning, J. Mach. Learn. Res. 18 (2017) 64:1–64:34.

[45] L. Iannone, I. Palmisano, N. Fanizzi, An algorithm based on counterfactuals for concept learning in the semantic web, Appl. Intell. 26 (2007) 139–159.

[46] U. Straccia, M. Mucci, pFOIL-DL: Learning (fuzzy) $\mathcal{EL}$ concept descriptions from crisp OWL data using a probabilistic ensemble estimation, in: Proc. of SAC, 2015, pp. 345–352.

[47] A. Artale, R. Kontchakov, A. Kovtunova, V. Ryzhikov, F. Wolter, M. Zakharyaschev, First-order rewritability of ontology-mediated queries in linear temporal logic, Artif. Intell. 299 (2021) 103536. URL: https://doi.org/10.1016/j.artint.2021.103536. doi:10.1016/j.artint.2021.103536.

[48] D. Gabbay, A. Kurucz, F. Wolter, M. Zakharyaschev, Many-Dimensional Modal Logics: Theory and Applications, volume 148 of *Studies in Logic*, Elsevier, 2003.

[49] A. Artale, R. Kontchakov, V. Ryzhikov, M. Zakharyaschev, The complexity of clausal fragments of LTL, in: Proc. of the 19th Int. Conf. on Logic for Programming, Artificial Intelligence and Reasoning, LPAR 2013, volume 8312 of *LNCS*, Springer, 2013, pp. 35–52.

[50] C.-C. Chen, I.-P. Lin, The computational complexity of the satisfiability of modal Horn clauses for modal propositional logics, Theor. Comp. Sci. 129 (1994) 95–121.

[51] M. Y. Vardi, Automata-theoretic techniques for temporal reasoning, in: P. Blackburn, J. F. A. K. van Benthem, F. Wolter (Eds.), Handbook of Modal Logic, volume 3 of *Studies in logic and practical reasoning*, North-Holland, 2007, pp. 971–989. URL: https://doi.org/10.1016/s1570-2464(07)80020-6. doi:10.1016/s1570-2464(07)80020-6.

[52] M. Sipser, Introduction to the theory of computation, PWS Publishing Company, 1997.