Reverse Engineering of Temporal Queries Mediated by LTL Ontologies

Marie Fortin[1] , **Boris Konev**[2] , **Vladislav Ryzhikov**[3] , **Yury Savateev**[4] ,
Frank Wolter[2] and **Michael Zakharyaschev**[3]
[1]Université Paris Cité, CNRS, IRIF, France
[2]Department of Computer Science, University of Liverpool, UK
[3]Department of Computer Science and Information Systems, Birkbeck, University of London, UK
[4]School of Electronics and Computer Science, University of Southampton, UK
mfortin@irif.fr, {boris.konev,wolter}@liverpool.ac.uk, {vlad,michael}@dcs.bbk.ac.uk,
y.savateev@soton.ac.uk

Abstract

In reverse engineering of database queries, we aim to construct a query from a given set of answers and non-answers; it can then be used to explore the data further or as an explanation of the answers and non-answers. We investigate this query-by-example problem for queries formulated in positive fragments of linear temporal logic *LTL* over timestamped data, focusing on the design of suitable query languages and the combined and data complexity of deciding whether there exists a query in the given language that separates the given answers from non-answers. We consider both plain *LTL* queries and those mediated by *LTL*-ontologies.

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 since the 2000s [Martins, 2019]. In the database community, research has focussed on standard query languages such as SQL, graph query languages, and SPARQL [Zhang *et al.*, 2013; Weiss and Cohen, 2017; Kalashnikov *et al.*, 2018; Deutch and Gilad, 2019; Staworko and Wieczorek, 2012; Barceló and Romero, 2017; Cohen and Weiss, 2016; Arenas *et al.*, 2016]. The KR community has been concerned with constructing queries from examples under the open world semantics and with background knowledge given by an ontology [Gutiérrez-Basulto *et al.*, 2018; Ortiz, 2019; Cima *et al.*, 2021; Jung *et al.*, 2021; Jung *et al.*, 2022]. A fundamental problem that has been investigated by both communities is known as *separability* or *query-by-example* (QBE), a term coined by Zloof [1977]:

Given: sets E^+ and E^- of pairs $(\mathcal{D}, \boldsymbol{d})$ with a database instance $\mathcal{D}$ and a tuple $\boldsymbol{d}$ in $\mathcal{D}$, a (possibly empty) ontology $\mathcal{O}$, and a query language $\mathcal{Q}$.

Problem: decide whether there exists a query $\boldsymbol{q} \in \mathcal{Q}$ *separating* (E^+, E^-) in the sense that $\mathcal{O}, \mathcal{D} \models \boldsymbol{q}(\boldsymbol{d})$ 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 such a $\boldsymbol{q}$ exists, then (E^+, E^-) is often called *satisfiable* w.r.t. $\mathcal{Q}$ under $\mathcal{O}$, and the construction of $\boldsymbol{q}$ is called *learning*.

In many applications, the input data is timestamped and queries are naturally formulated in languages with temporal operators. In this paper, we investigate temporal query-by-example by focusing on the basic but very useful case where data $\mathcal{D}$ is a set of timestamped atomic propositions. 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 [Demri *et al.*, 2016]. To enforce generalisation, we do not admit $\vee$. Our most expressive query language $\mathcal{Q}[\mathsf{U}]$ is thus defined as the set of formulas constructed from atoms using $\wedge$ and U (via which $\bigcirc$ and $\Diamond$ are expressible); the fragments $\mathcal{Q}[\Diamond]$ and $\mathcal{Q}[\bigcirc, \Diamond]$ are defined analogously. Ontologies can be given in full *LTL* or its fragments $\mathit{LTL}^{\Box\Diamond}$ (known as the *Prior logic* [Prior, 1956]), which only uses the operators $\Box$ (always in the future) and $\Diamond$, and the Horn fragment $\mathit{LTL}^{\Box\bigcirc}_{horn}$ 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 $i \leq k$. Ontology axioms are supposed to hold at all times. In fact, already this basic 'one-dimensional' temporal ontology-mediated querying formalism provides enough expressive power in those real-world situations where the interaction among individuals in the object domain is not important and can be disregarded in data modelling; see [Artale *et al.*, 2021] and also Example 1 and the references before it.

Within this temporal setting, we take a broad view of the potential applications of the QBE problem. On the one hand, there are non-expert 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. QBE 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 *LTL* formulas in explainable planning and program synthesis [Lemieux *et al.*, 2015; Neider and Gavran, 2018; Camacho and McIlraith, 2019; Fijalkow and Lagarde, 2021;

Omitted details and proofs are available in [Fortin *et al.*, 2023].

Raha *et al.*, 2022; Fortin *et al.*, 2022].

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' (T), 'strong vibration' (V), etc. Suppose the engine stopped after the runs $\mathcal{D}_1^+$ and $\mathcal{D}_2^+$ 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 moments $0, 1, 2, \dots$:

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

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

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

is explained by the U-query $T \,\mathsf{U}\, V$. Using background knowledge, we can compensate for sensor failures resulting in incomplete data. To illustrate, suppose $\mathcal{E}_1^+ = \{H(3), V(4)\}$, where H means 'heater is on'. If an ontology $\mathcal{O}$ has 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 $\boldsymbol{q}$ separates $\{\mathcal{E}_1^+, \mathcal{D}_2^+\}$ from $\{\mathcal{D}_1^-, \mathcal{D}_2^-, \mathcal{D}_3^-\}$ under $\mathcal{O}$. $\dashv$

Query $\boldsymbol{q}$ in Example 1 is of a particular 'linear' form, in which the order of atoms is fixed and not left open as, for instance, in the 'branching' $\Diamond T \land \Diamond V$. More precisely, *path $\bigcirc\Diamond$-queries* in the class $\mathcal{Q}_p[\bigcirc, \Diamond]$ take the form

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

where $\boldsymbol{o}_i \in \{\bigcirc, \Diamond\}$ and ρ_i is a conjunction of atoms; $\mathcal{Q}_p[\Diamond]$ restricts $\boldsymbol{o}_i$ to $\{\Diamond\}$; and *path* U*-queries* $\mathcal{Q}_p[\mathsf{U}]$ look like

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

where λ_i is a conjunction of atoms or $\bot$. Path queries are motivated by two observations. First, if a query language admits conjunctions of queries—unlike our classes of path queries—then, dually to overfitting for $\lor$, multiple negative examples become redundant: if $\boldsymbol{q}_\mathcal{D}$ separates $(E^+, \{\mathcal{D}\})$, for each $\mathcal{D} \in E^-$, then $\bigwedge_{\mathcal{D} \in E^-} \boldsymbol{q}_\mathcal{D}$ separates (E^+, E^-). Second, numerous natural query types known from applications can be captured by path queries. For example, the existence of a *common subsequence* of the positive examples (regarded as words) that is not a subsequence of any negative one corresponds to the existence of a separating $\mathcal{Q}_p[\Diamond]$-query 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 one corresponds to the existence of a separating query of the form $\Diamond(\rho_1 \land \bigcirc(\rho_2 \land \cdots \land \bigcirc\rho_n))$. These and similar queries are the basis of data comparison programs with numerous applications in computational linguistics, bioinformatics, and revision control systems [Bergroth *et al.*, 2000; Chowdhury *et al.*, 2010; Blum *et al.*, 2021].

While path queries express the intended separating pattern of events in many applications, branching queries are needed if the order of events is irrelevant for separation.

Example 2. In the setting of Example 1, the positive examples $\{T(2), V(4)\}$ and $\{V(1), T(4)\}$ are separated from the negative $\{T(1)\}$ and $\{V(4)\}$ by the branching $\mathcal{Q}[\Diamond]$-query $\Diamond T \land \Diamond V$ while no path query is capable of doing this. $\dashv$

Branching $\mathcal{Q}[\bigcirc, \Diamond]$-queries express transparent existential conditions and can be regarded as *LTL* CQs. However, branching $\mathcal{Q}[\mathsf{U}]$-queries with nestings of U on the left-hand side correspond to complex first-order formulas with multiple alternations of quantifiers $\exists$ and $\forall$, which are hard to comprehend. So we also consider the language $\mathcal{Q}[\mathsf{U}_s] \supseteq \mathcal{Q}_p[\mathsf{U}]$ of 'simple' $\mathcal{Q}[\mathsf{U}]$-queries without such nestings.

In this paper, we take the first steps towards understanding the complexity and especially feasibility of the query-by-example problems $\mathsf{QBE}(\mathcal{L}, \mathcal{Q})$ with $\mathcal{L}$ an ontology and $\mathcal{Q}$ a query language. We are particularly interested in whether there is a difference in complexity between path and branching queries and whether it can be reduced by bounding the number of positive or negative examples. Our results in the ontology-free case are summarised in Table 1, where

QBE for	b+, b−	b+	b− or unbounded
$\mathcal{Q}_p[\Diamond]/\mathcal{Q}_p[\bigcirc, \Diamond]$	≤P	=NP	=NP
$\mathcal{Q}[\Diamond]/\mathcal{Q}[\bigcirc, \Diamond]$	≤P	≤P	=NP
$\mathcal{Q}_p[\mathsf{U}]$	=NP		
$\mathcal{Q}[\mathsf{U}_s]$	≤P	≤P	≥NP, ≤PSPACE
$\mathcal{Q}[\mathsf{U}]$	≤PSPACE		

Table 1: Complexity in the ontology-free case.

b+ / b− indicate that the number of positive / negative examples is bounded[1]. Note that path queries are indeed harder than branching ones when the number of positive examples is bounded but not in the unbounded case. Our proof techniques range from reductions to common subsequence existence problems [Maier, 1978; Fraser, 1996] and dynamic programming to mimicking separability by path and branching U-queries in terms of containment and simulation of transition systems [Kupferman and Vardi, 1996]. The key to NP upper bounds is the *polynomial separation property* (PSP) of the respective languages: any separable example set is separated by a polynomial-size query. The complexity for $\mathcal{Q}_p[\Diamond]$, $\mathcal{Q}[\Diamond]$ can also be obtained from [Fijalkow and Lagarde, 2021] who studied separability by $\mathcal{Q}[\Diamond]$-queries of bounded size.

In the presence of ontologies, we distinguish between the combined complexity of $\mathsf{QBE}(\mathcal{L}, \mathcal{Q})$, when both data and ontology are regarded as input, and the data complexity, when the ontology is deemed fixed or negligibly small compared with the data. We obtain encouraging results: $\mathcal{Q}_p[\Diamond]$- and $\mathcal{Q}[\Diamond]$-queries mediated by $\mathit{LTL}^{\Box\Diamond}$-ontologies and all of our queries mediated by $\mathit{LTL}^{\Box\bigcirc}_{\mathit{horn}}$-ontologies enjoy the same data complexity as in Table 1. The combined complexity results for queries with $\mathit{LTL}^{\Box\bigcirc}_{\mathit{horn}}$-ontologies we have obtained so far are given in Table 2. Interestingly, QBE for query classes with $\Diamond$ and $\bigcirc$ only is PSPACE-complete— not harder than satisfiability. The upper bound is proved by establishing the *exponential separation property* for all of these classes of queries and using the canonical (aka minimal) model property of

[1] We do not consider queries with $\bigcirc$ only as separability is trivially in P and does not detect any useful patterns.

$\mathcal{Q}[\Diamond]$ / $\mathcal{Q}_p[\Diamond]$ $\mathcal{Q}[\bigcirc, \Diamond]$ / $\mathcal{Q}_p[\bigcirc, \Diamond]$	= PSPACE
$\mathcal{Q}[\mathsf{U}_s]$	$\geq$ PSPACE, $\leq$ EXPTIME
$\mathcal{Q}_p[\mathsf{U}]$	$\geq$ NEXPTIME, $\leq$ EXPSPACE
$\mathcal{Q}[\mathsf{U}]$	$\geq$ PSPACE, $\leq$ 2EXPTIME

Table 2: Combined complexity of $\mathsf{QBE}(\mathit{LTL}^{\Box\bigcirc}_{\mathit{horn}}, \mathcal{Q})$ in both bounded and unbounded cases.

Horn *LTL*. The upper bounds for U-queries are by reduction to the simulation and containment problems for exponential-size transition systems. For arbitrary *LTL*-ontologies, this technique only gives a 2EXPTIME upper bound for $\mathcal{Q}[\mathsf{U}_s]$ and a 2EXPSPACE one for $\mathcal{Q}_p[\mathsf{U}]$. Separability by (path) $\Diamond$-queries under $\mathit{LTL}^{\Box\Diamond}$ ontologies turns out to be Σ^p_2-complete, where the upper bound is shown by establishing the PSP.

Compared with non-temporal QBE, our results are very encouraging: QBE is CONEXPTIME-complete for conjunctive queries (CQs) over standard relational databases [Willard, 2010; ten Cate and Dalmau, 2015] and even undecidable for CQs under $\mathcal{ELI}$ or $\mathcal{ALC}$ ontologies [Funk *et al.*, 2019; Jung *et al.*, 2020].

2 Further Related Work

We now briefly comment on a few other related research areas. One of them is concept learning in description logic (DL), as proposed by [Badea and Nienhuys-Cheng, 2000] who, inspired by inductive logic programming, used refinement operators to construct a concept separating positive and negative examples in a DL ABox. There has been significant interest in this approach [Lehmann and Haase, 2009; Lehmann and Hitzler, 2010; Lisi and Straccia, 2015; Sarker and Hitzler, 2019; Lisi, 2012; Rizzo *et al.*, 2020]. Prominent systems include the DL LEARNER [Bühmann *et al.*, 2016], DL-FOIL [Fanizzi *et al.*, 2018] and its extension DL-FOCL [Rizzo *et al.*, 2018], SPaCEL [Tran *et al.*, 2017], YINYANG [Iannone *et al.*, 2007], pFOIL-DL [Straccia and Mucci, 2015], and EVOLEARNER [Heindorf *et al.*, 2022]. However, this work has not considered the complexity of separability. Also closely related is the work on the separability of two formal (e.g., regular) languages using a weaker (e.g., FO-definable) language [Place and Zeitoun, 2016; Hofman and Martens, 2015; Place and Zeitoun, 2022]. When translated into a logical separability problem, the main difference to our results is that one demands $\mathcal{O}, \mathcal{D} \models \neg \boldsymbol{q}(\boldsymbol{d})$—and not just $\mathcal{O}, \mathcal{D} \not\models \boldsymbol{q}(\boldsymbol{d})$—for all $(\mathcal{D}, \boldsymbol{d}) \in E^-$.

3 Preliminaries

LTL-formulas are built from *atoms* A_i, $i < \omega$, using the Booleans and (future-time) temporal operators $\bigcirc$, $\Diamond$, $\Box$, U, which we interpret under the *strict semantics* [Gabbay *et al.*, 2003; Demri *et al.*, 2016]. An *LTL-interpretation* $\mathcal{I}$ identifies those atoms A_i that are *true* at each time instant $n \in \mathbb{N}$, written $\mathcal{I}, n \models A_i$. The truth-relation for atoms is extended inductively to *LTL*-formulas by taking $\mathcal{I}, n \models \varphi \mathbin{\mathsf{U}} \psi$ iff $\mathcal{I}, m \models \psi$, for some $m > n$, and $\mathcal{I}, k \models \varphi$ for all $k \in (n, m)$, and using the standard clauses for the Booleans and equivalences $\bigcirc\varphi \equiv \bot \mathbin{\mathsf{U}} \varphi$, $\Diamond\varphi \equiv \top \mathbin{\mathsf{U}} \varphi$ and $\Box\varphi \equiv \neg\Diamond\neg\varphi$ with Boolean *constants* $\bot$ and $\top$ for 'false' and 'true'.

An *LTL-ontology*, $\mathcal{O}$, is any finite set of *LTL*-formulas, called the *axioms* of $\mathcal{O}$. An interpretation $\mathcal{I}$ is a *model* of $\mathcal{O}$ if all axioms of $\mathcal{O}$ are true at *all times* in $\mathcal{I}$. As mentioned in the introduction, apart from full *LTL* we consider its Prior $\Box\Diamond$-fragment $\mathit{LTL}^{\Box\Diamond}$ and $\mathit{LTL}^{\Box\bigcirc}_{\mathit{horn}}$ whose axioms take the form

$$C_1 \wedge \dots \wedge C_k \;\to\; C_{k+1} \tag{3}$$

with C_i given by $C ::= A_i \mid \bot \mid \Box C \mid \bigcirc C$. In fact, we could allow $\Diamond$ on the left-hand side of (3) as $\Diamond C \to C'$ can be replaced by $\bigcirc C \to A$, $\bigcirc A \to A$, $A \to C'$ with fresh A.

A *data instance* is a finite set $\mathcal{D}$ of atoms $A_i(\ell)$ with a *timestamp* $\ell \in \mathbb{N}$; $\max \mathcal{D}$ is the maximal timestamp in $\mathcal{D}$. We access data by means of *LTL* analogues of conjunctive queries: our *queries*, $\varkappa$, are constructed from atoms, $\bot$ and $\top$ using $\wedge$, $\bigcirc$, $\Diamond$ and U. The class of queries that only use operators from $\Phi \subseteq \{\bigcirc, \Diamond, \mathsf{U}\}$ is denoted by $\mathcal{Q}[\Phi]$; $\mathcal{Q}_p[\Phi]$ is its subclass of *path-queries*, which take the form (1) or (2); and $\mathcal{Q}[\mathsf{U}_s]$ comprises *simple queries* in $\mathcal{Q}[\mathsf{U}]$ that do not contain subqueries $\varkappa_1 \mathbin{\mathsf{U}} \varkappa_2$ with an occurrence of U in $\varkappa_1$. Note that $\mathcal{Q}_p[\mathsf{U}] \subseteq \mathcal{Q}[\mathsf{U}_s]$. The *temporal depth* $\mathit{tdp}(\varkappa)$ of $\varkappa$ is the maximum number of nested temporal operators in $\varkappa$.

An interpretation $\mathcal{I}$ is a *model* of a data instance $\mathcal{D}$ if $\mathcal{I}, \ell \models A_i$ for all $A_i(\ell) \in \mathcal{D}$. $\mathcal{O}$ and $\mathcal{D}$ are *consistent* if they have a model. We call $k \leq \max \mathcal{D}$ a (*certain*) *answer* to the *ontology-mediated query* $(\mathcal{O}, \varkappa)$ over $\mathcal{D}$ and write $\mathcal{O}, \mathcal{D} \models \varkappa(k)$ if $\mathcal{I}, k \models \varkappa$ in all models $\mathcal{I}$ of $\mathcal{O}$ and $\mathcal{D}$.

Let $\mathcal{L}$ and $\mathcal{Q}$ be an ontology and query language defined above. The *query-by-example problem* $\mathsf{QBE}(\mathcal{L}, \mathcal{Q})$ we are concerned with in this paper is formulated as follows:

given an $\mathcal{L}$-ontology $\mathcal{O}$ and an *example set* $E = (E^+, E^-)$ with finite sets E^+ and E^- of *positive* and, respectively, *negative* data instances,

decide whether E is $\mathcal{Q}$-*separable* under $\mathcal{O}$ in the sense that there is a $\mathcal{Q}$-query $\varkappa$ with $\mathcal{O}, \mathcal{D} \models \varkappa(0)$ for all $\mathcal{D} \in E^+$ and $\mathcal{O}, \mathcal{D} \not\models \varkappa(0)$ for all $\mathcal{D} \in E^-$.

If $\mathcal{L} = \emptyset$, we shorten $\mathsf{QBE}(\emptyset, \mathcal{Q})$ to $\mathsf{QBE}(\mathcal{Q})$. We also consider the QBE problems with the input example sets having a *bounded* number of positive and/or negative examples, denoted $\mathsf{QBE}^{\mathsf{b+}}(\mathcal{L}, \mathcal{Q})$, $\mathsf{QBE}_{\mathsf{b-}}(\mathcal{L}, \mathcal{Q})$, or $\mathsf{QBE}^{\mathsf{b+}}_{\mathsf{b-}}(\mathcal{L}, \mathcal{Q})$. Notations like $\mathsf{QBE}^{2+}_{1-}(\mathcal{L}, \mathcal{Q})$ should be self-explanatory. The size of $\mathcal{O}$, E, $\varkappa$, denoted $|\mathcal{O}|$, $|E|$, $|\varkappa|$, respectively, is the number of symbols in it with the timestamps given in unary.

The next example illustrates the definitions and relative expressive power of queries with different temporal operators.

Example 3. (a) Let $E = (\{\mathcal{D}_1\}, \{\mathcal{D}_2\})$ with $\mathcal{D}_1 = \{A(1)\}$, $\mathcal{D}_2 = \{A(2)\}$. Then $\bigcirc A$ separates E but no $\mathcal{Q}[\Diamond]$-query does. E is not separable under $\mathcal{O} = \{\bigcirc A \to A\}$ by any query $\varkappa$ as $\mathcal{O}, \mathcal{D}_1 \models \varkappa(0)$ implies $\mathcal{O}, \mathcal{D}_2 \models \varkappa(0)$.

(b) Let $E = (\{\mathcal{D}_1, \mathcal{D}_2\}, \{\mathcal{D}_3\})$ with $\mathcal{D}_1 = \{A(1), B(2)\}$, $\mathcal{D}_2 = \{A(2), B(3)\}$, $\mathcal{D}_3 = \{A(3), B(5)\}$. Then the query $\Diamond(A \wedge \bigcirc B)$ separates E but no query in $\mathcal{Q}[\Diamond]$ does.

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

We now establish a few important *polynomial-time reductions*, $\leq_p$, among the QBE-problems for various query classes, including $\mathcal{Q}_p^\circ[\Diamond]$-*queries* of the form

$$\varkappa = \rho_0 \wedge \Diamond(\rho_1 \wedge \Diamond(\rho_2 \wedge \cdots \wedge \Diamond\rho_n)), \qquad (4)$$

where each ρ_i is a $\mathcal{Q}_p[\bigcirc]$-query (i.e., $\Diamond$-free $\mathcal{Q}_p[\bigcirc,\Diamond]$-query).

Theorem 4. *The following polynomial-time reductions hold*:

(*i*.1) $\mathsf{QBE}(\mathcal{L},\mathcal{Q}) \leq_p \mathsf{QBE}_{1\text{-}}(\mathcal{L},\mathcal{Q})$*, for any* $\mathcal{Q}$ *closed under* $\wedge$*, and any* $\mathcal{L}$ (*including* $\mathcal{L} = \emptyset$)*,*

(*i*.2) $\mathsf{QBE}(\mathcal{L},\mathcal{Q}) \leq_p \mathsf{QBE}^{2+}(\mathcal{L},\mathcal{Q})$*, for* $\mathcal{L} \in \{\mathit{LTL}, \mathit{LTL}^{\Box\Diamond}\}$*,*

(*i*.3) $\mathsf{QBE}(\mathcal{L},\mathcal{Q}[\bigcirc,\Diamond]) \leq_p \mathsf{QBE}(\mathcal{L},\mathcal{Q}_p^\circ[\Diamond]))$ *and* $\mathsf{QBE}(\mathcal{L},\mathcal{Q}[\Diamond]) \leq_p \mathsf{QBE}(\mathcal{L},\mathcal{Q}_p[\Diamond])$*, for any* $\mathcal{L}$*,*

(*ii*.1) $\mathsf{QBE}(\mathcal{Q}_p[\Diamond]) \leq_p \mathsf{QBE}(\mathcal{Q}_p[\bigcirc,\Diamond])$ *and* $\mathsf{QBE}(\mathcal{Q}_p[\Diamond]) \leq_p \mathsf{QBE}(\mathcal{Q}_p[\mathsf{U}]) \leq_p \mathsf{QBE}_{1\text{-}}(\mathcal{Q}_p[\mathsf{U}])$*,*

(*ii*.2) $\mathsf{QBE}(\mathcal{Q}[\bigcirc,\Diamond]) =_p \mathsf{QBE}(\mathcal{Q}[\Diamond]) \leq_p \mathsf{QBE}(\mathcal{Q}[\mathsf{U}_s])$.

Reductions $(i.1)$–$(i.3)$ *work for combined complexity*; $(i.1)$, $(i.3)$ *also work for data complexity. The reductions preserve boundedness of the number of positive/negative examples.*

Proof. In $(i.1)$, (E^+, E^-) with $E^- = \{\mathcal{D}_1, \dots, \mathcal{D}_n\}$ is $\mathcal{Q}$-separable under $\mathcal{O}$ iff each $(E^+, \{\mathcal{D}_i\})$ is because if $\varkappa_i$ separates $(E^+, \{\mathcal{D}_i\})$, then $\varkappa_1 \wedge \cdots \wedge \varkappa_n$ separates (E^+, E^-).

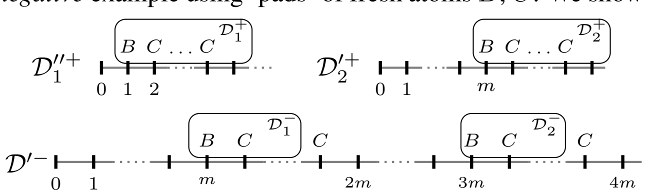
In $(i.2)$, (E^+, E^-) with $E^+ = \{\mathcal{D}_1, \dots, \mathcal{D}_n\}$, $n > 1$, is $\mathcal{Q}$-separable under $\mathcal{O}$ iff (E'^+, E^-) is $\mathcal{Q}$-separable under $\mathcal{O}'$ that extends $\mathcal{O}$ with the following axioms simulating E^+:

$$S_1 \to A_1 \vee \cdots \vee A_n, \quad S_2 \to A_1 \vee \cdots \vee A_n,$$
$$C_i \wedge \Diamond A_j \to X, \quad D_i \wedge \Diamond A_j \to X, \text{ for } X(i) \in \mathcal{D}_j,$$

where S_1, S_2, A_k, C_l, D_l, for $l \leq n' = \max_i \max \mathcal{D}_i$, are fresh and E'^+ consists of $\{C_0(0), \dots, C_k(n'), S_1(n'+1)\}$ and $\{D_0(0), \dots, D_k(n'), S_2(n'+1)\}$.

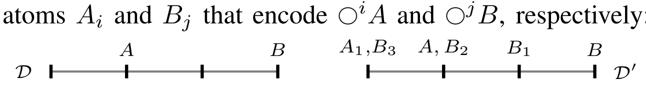
$(i.3)$ Using $[\rho_0 \wedge \Diamond(\rho_1 \wedge \bigwedge_i \Diamond\varkappa_i)] \equiv [\rho_0 \wedge \bigwedge_i \Diamond(\rho_1 \wedge \Diamond\varkappa_i)]$, $\bigcirc\Diamond\varkappa \equiv \Diamond\bigcirc\varkappa$ and $\bigcirc(\varkappa \wedge \varkappa') \equiv (\bigcirc\varkappa \wedge \bigcirc\varkappa')$ we convert, in polytime, each $\mathcal{Q}[\bigcirc,\Diamond]$-query to an equivalent conjunction of $\mathcal{Q}_p^\circ[\Diamond]$-queries. Thus, there is $\boldsymbol{q} \in \mathcal{Q}[\bigcirc,\Diamond]$ separating (E^+, E^-) iff there are polysize $\boldsymbol{q}_{\mathcal{D}} \in \mathcal{Q}_p^\circ[\Diamond]$ separating $(E^+, \{\mathcal{D}\})$, for each $\mathcal{D} \in E^-$.

$(ii.1)$ The first two reductions are shown by adding to $E^+ \ni \mathcal{D}$, for some $\mathcal{D}$, the data instance $\mathcal{D}' = \{A(mn) \mid A(n) \in \mathcal{D}\}$ with $m = \max\mathcal{D} + 2$. Now, if $\mathcal{D} \models \varkappa(0)$ and $\mathcal{D}' \models \varkappa(0)$, for $\varkappa \in \mathcal{Q}_p[\mathsf{U}]$, then $\varkappa$ is equivalent to a $\mathcal{Q}_p[\Diamond]$-query. The third reduction, illustrated below for $E^+ = \{\mathcal{D}_1^+, \mathcal{D}_2^+\}$ and $E^- = \{\mathcal{D}_1^-, \mathcal{D}_2^-\}$, transforms E into two positive and *one negative* example using 'pads' of fresh atoms B, C. We show



that E is $\mathcal{Q}_p[\mathsf{U}]$-separable iff $(\{\mathcal{D}_1''^+, \mathcal{D}_2'^+\}, \{\mathcal{D}'^-\})$ is.

$(ii.2)$ The first reduction is established by modifying every $\mathcal{D}$ in the given E as illustrated below using fresh atoms A_i and B_j that encode $\bigcirc^i A$ and $\bigcirc^j B$, respectively:



Then E is $\mathcal{Q}[\bigcirc,\Diamond]$-separable iff E' is $\mathcal{Q}[\Diamond]$-separable. The converse and the second reduction are similar to $(ii.1)$. $\dashv$

4 QBE without Ontologies

We start investigating the complexity of the QBE problems for *LTL* by considering queries without mediating ontologies.

Theorem 5. *The* QBE*-problems for the classes of queries defined above* (*with the empty ontology*) *belong to the complexity classes shown in Table 1.*

We comment on the proof in the remainder of this section.

$\bigcirc\Diamond$-queries. NP-hardness is established by reduction of the consistent subsequence existence problems [Fraser, 1996, Theorems 2.1, 2.2] in tandem with Theorem 4; membership in NP follows from the fact that separating queries, if any, can always be taken of polynomial size.

Tractability is shown using dynamic programming. We explain the idea for $\mathsf{QBE}^{\mathsf{b+}}_{\mathsf{b-}}(\mathcal{Q}_p[\bigcirc,\Diamond])$, $E^+ = \{\mathcal{D}_1^+, \mathcal{D}_2^+\}$ and $E^- = \{\mathcal{D}_1^-, \mathcal{D}_2^-\}$. Suppose $\varkappa$ takes the form (1) with $\rho_n \neq \top$. Then $\mathcal{D} \models \varkappa(0)$ iff there is a strictly monotone map $f\colon [0,n] \to [0, \max\mathcal{D}]$ with $f(0) = 0$, $f(i+1) = f(i)+1$ if $\boldsymbol{o}_i = \bigcirc$, and $\rho_i \subseteq t_{\mathcal{D}}(f(i)) = \{A \mid A(f(i)) \in \mathcal{D}\}$. We call such an f a *satisfying assignment* for $\varkappa$ in $\mathcal{D}$. Let $S_{i,j}$ be the set of tuples $(k, \ell_1, \ell_2, n_1, n_2)$ such that $\ell_1 \leq i \leq \max\mathcal{D}_1^+$, $\ell_2 \leq j \leq \max\mathcal{D}_2^+$, and there is $\varkappa = \rho_0 \wedge \boldsymbol{o}_1(\rho_1 \wedge \cdots \wedge \boldsymbol{o}_k\rho_k)$ for which (i) there are satisfying assignments f_1, f_2 in $\mathcal{D}_1^+$ and $\mathcal{D}_2^+$ with $f_1(k) = \ell_1$ and $f_2(k) = \ell_2$, respectively, and (ii) n_1 is minimal with a satisfying assignment f for $\varkappa$ in $\mathcal{D}_1^-$ having $f(k) = n_1$, and $n_1 = \infty$ if there is no such f; and similarly for n_2, $\mathcal{D}_2^-$. It suffices to compute $S_{\max\mathcal{D}_1^+, \max\mathcal{D}_2^+}$ in polytime. This can be done incrementally by initially observing that $S_{0,j}$ can only contain $(0,0,0,0,0)$, which is the case if there is $\rho_0 \subseteq t_{\mathcal{D}_1^+}(0)$, $\rho_0 \subseteq t_{\mathcal{D}_2^+}(0)$ and $\rho_0 \not\subseteq t_{\mathcal{D}_1^-}(0)$, $\rho_0 \not\subseteq t_{\mathcal{D}_2^-}(0)$ (and similarly for $S_{i,0}$).

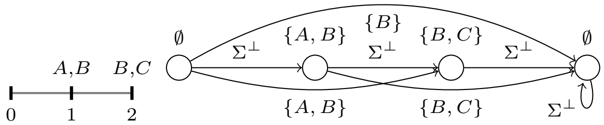
U-queries. NP-hardness for $\mathcal{Q}_p[\mathsf{U}]$, $\mathcal{Q}[\mathsf{U}_s]$ follows from Theorem 4 $(ii.1)$, $(ii.2)$ and NP-hardness for $\bigcirc\Diamond$-queries.

The upper bounds are shown by reduction of $\mathcal{Q}_p[\mathsf{U}]$- and $\mathcal{Q}[\mathsf{U}_s]$-separability to the simulation and containment problems for transition systems [Kupferman and Vardi, 1996]. A *transition system*, S, is a digraph each of whose nodes and edges is labelled by some set of symbols from a node or, respectively, edge alphabet; S also has a designated set S_0 of *start nodes*. A *run* of S is a path in digraph S, starting in S_0, together with all of its labels. The *computation tree* of S is the tree unravelling $\mathfrak{T}_S$ of S. For systems S and S' over the same alphabets, we say that S is *contained* in S' if, for every run r of S, there is a run r' of S' such that r and r' have the same length and the labels on the states and edges in r are subsumed by the corresponding labels in r'. S is *simulated* by S' if $\mathfrak{T}_S$ is *finitely embeddable* into $\mathfrak{T}_{S'}$ in the sense that every finite subtree[2] of $\mathfrak{T}_S$ can be homomorphically mapped into $\mathfrak{T}_{S'}$ preserving (subsumption of) node and edge labels.

Now, let $E = (E^+, E^-)$ with $E^\sigma = \{\mathcal{D}_i \mid i \in I^\sigma\}$, for $\sigma \in \{+,-\}$ and disjoint I^+ and I^-, and let Σ be the signature of E. For each $i \in I^+ \cup I^-$, we take a transition system S^i with states $0^i, \dots, (\max\mathcal{D}_i + 1)^i$, where $(\max\mathcal{D}_i + 1)^i$ is labelled with $\emptyset$ and the remaining j^i by $\{A \mid A(j) \in \mathcal{D}_i\}$. Transitions are $j^i \to k^i$, for $0 \leq j < k \leq \max\mathcal{D}_i + 1$, that

[2] A *subtree* is a convex subset of $\mathfrak{T}_S$'s nodes with some start node.

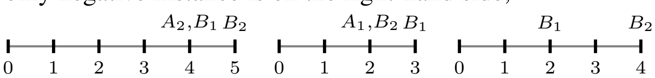
are labelled by $\{A \in \Sigma \cup \{\bot\} \mid A(n) \in \mathcal{D}_i, n \in (j,k)\}$ and $(\max \mathcal{D}_i + 1)^i \to (\max \mathcal{D}_i + 1)^i$ with label $\Sigma^\bot = \Sigma \cup \{\bot\}$. Thus, $\mathcal{D}_i$ shown on the left below gives rise to S^i on the right:



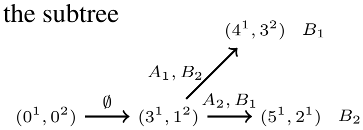
We form the direct product (synchronous composition) $\mathfrak{P}$ of $\{S^i \mid i \in I^+\}$, for $I^+ = \{1,\dots,l\}$, whose states are vectors $(s_1,\dots,s_l)$ of states $s_i \in S^i$, which are labelled by the intersection of the labels of s_i in S^i, with transitions $(s_1,\dots,s_l) \to (p_1,\dots,p_l)$, if $s_i \to p_i$ in S^i for all i, also labelled by the intersection of the component transition labels. On the other hand, we take the disjoint union $\mathfrak{N}$ of S^i, for $i \in I^-$, and establish the following separability criterion:

Theorem 6. *(i) E is not $\mathcal{Q}[\mathsf{U}_s]$-separable iff $\mathfrak{P}$ is simulated by $\mathfrak{N}$. (ii) E is not $\mathcal{Q}_p[\mathsf{U}]$-separable iff $\mathfrak{P}$ is contained in $\mathfrak{N}$.*

Example 7. For the example set depicted below, in which the only negative instance is on the right-hand side,



$\mathfrak{T}_{\mathfrak{P}}$ contains the subtree

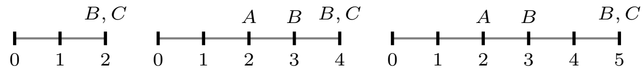


where only the last $\mathfrak{P}$-node of a $\mathfrak{T}_{\mathfrak{P}}$-node (a sequence) is indicated together with the atoms that are true at nodes and on edges. Intuitively, $\mathfrak{T}_{\mathfrak{P}}$ 'represents' all possible $\mathcal{Q}[\mathsf{U}_s]$-queries and its paths represent $\mathcal{Q}_p[\mathsf{U}]$-queries $\varkappa$ such that $\mathcal{O}, \mathcal{D} \models \varkappa(0)$ for all $\mathcal{D} \in E^+$. The $\mathcal{Q}[\mathsf{U}_s]$-query given by the subtree above is $\varkappa = \Diamond\big(((A_1 \land B_2) \mathsf{U} B_1) \land ((A_2 \land B_1) \mathsf{U} B_2)\big)$. The subtree is not embeddable into $\mathfrak{T}_{\mathfrak{N}}$ (obtained for the negative instance), so $\varkappa$ separates E. Observe that every path in $\mathfrak{T}_{\mathfrak{P}}$ (and in the subtree above) is embeddable into $\mathfrak{T}_{\mathfrak{N}}$.

By inspecting the structure of $\mathfrak{P}$ and $\mathfrak{N}$ we observe that if $\mathfrak{P}$ has a run that is not embeddable into any run of $\mathfrak{N}$, then we can find such a run of length $\le M = \min\{\max \mathcal{D}_i \mid i \in I^+\}$ (any longer run has $\emptyset$-labels on its states after the Mth one). Thus, we can guess the required run and check in P if it is correct, establishing the NP upper bound for $\mathcal{Q}_p[\mathsf{U}]$. To show the PSPACE upper bound for $\mathcal{Q}[\mathsf{U}_s]$, we notice that if there is a finite subtree of $\mathfrak{T}_{\mathfrak{P}}$ that is not embeddable into $\mathfrak{T}_{\mathfrak{N}}$, then the full subtree $\mathfrak{T}^M_{\mathfrak{P}}$ of depth M is not embeddable into $\mathfrak{T}_{\mathfrak{N}}$, which can be checked by constructing $\mathfrak{T}^M_{\mathfrak{P}}$ branch-by-branch while checking all possible embeddings of these branches into $\mathfrak{T}_{\mathfrak{N}}$. Finally, we have the P upper bound for $\mathcal{Q}[\mathsf{U}_s]$ with a bounded number of positive examples because $\mathfrak{P}$ is constructible in polytime and checking simulation between transition systems is P-complete [Kupferman and Vardi, 1996]. Interestingly, the smallest separating query we can construct in this case is of the same size as $\mathfrak{T}^M_{\mathfrak{P}}$, i.e., exponential in $|E^+|$; however, we can check its existence in polytime.

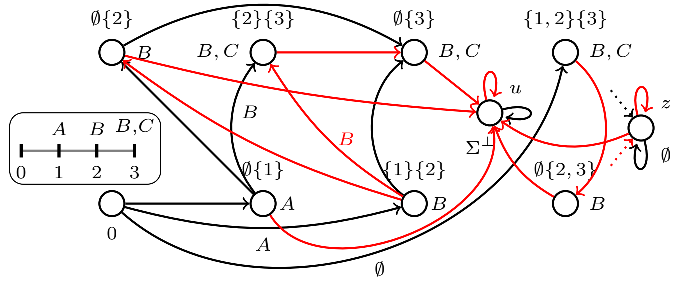
The PSPACE upper bound for $\mathcal{Q}[\mathsf{U}]$ requires a more sophisticated notion of simulation between transition systems.

Example 8. The example set below, where only the rightmost instance is negative, is separated by the $\mathcal{Q}[\mathsf{U}]$-query



$(A \mathsf{U} B) \mathsf{U} C$ but is not $\mathcal{Q}[\mathsf{U}_s]$-separable by Theorem 6. ⊣

We prove a $\mathcal{Q}[\mathsf{U}]$-inseparability criterion using transition systems whose non-initial/sink states are *pairs* of sets of numbers, and transitions are of *two* types. The picture below shows a data instance and the induced transition system (where z has incoming arrows labelled by $\Sigma^\bot$ from all states



but u, which are all omitted). Each arrow from 0 leads to a state $\{1,\dots,n-1\}\{n\}$; it represents a formula $\varphi \mathsf{U} \psi$ that is true at 0, with the arrow label indicating the non-nested atoms of φ and the state label indicating the atoms of ψ. Each black (resp., red) arrow from $\boldsymbol{s}_1\boldsymbol{s}_2$ to $\boldsymbol{s}'_1\boldsymbol{s}'_2$ represents a U-formula $\alpha_{\boldsymbol{s}_2 \to \boldsymbol{s}'_1\boldsymbol{s}'_2}$ (resp., $\alpha_{\boldsymbol{s}_1 \to \boldsymbol{s}'_1\boldsymbol{s}'_2}$) that is true at all points in $\boldsymbol{s}_2$ (resp., $\boldsymbol{s}_1$). The black and red transitions are arranged in such a way that a transition from $\boldsymbol{s}''_1\boldsymbol{s}''_2$ to $\boldsymbol{s}_1\boldsymbol{s}_2$ with an arrow label λ and $\boldsymbol{s}_1\boldsymbol{s}_2$-label μ represents the U-formula $(\lambda \land \bigwedge \alpha_{\boldsymbol{s}_1 \to \boldsymbol{s}'_1\boldsymbol{s}'_2}) \mathsf{U} (\mu \land \bigwedge \alpha_{\boldsymbol{s}_2 \to \boldsymbol{s}'_1\boldsymbol{s}'_2})$ and similarly for the transitions from 0. A version of Theorem 6 for $\mathcal{Q}[\mathsf{U}]$ and a PSPACE-algorithm are given in the full paper.

5 QBE with $\boldsymbol{LTL}^{\Box\bigcirc}_{\boldsymbol{horn}}$-Ontologies

Recall from [Artale *et al.*, 2021] that, for any $LTL^{\Box\bigcirc}_{horn}$-ontology $\mathcal{O}$ and data 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 any $k \in \mathbb{N}$, we have $\mathcal{O}, \mathcal{D} \models \varkappa(k)$ iff $\mathcal{C}_{\mathcal{O},\mathcal{D}} \models \varkappa(k)$.

Let $sub_{\mathcal{O}}$ be the set of subformulas of the C_i in the axioms (3) of $\mathcal{O}$ and their negations. A *type for* $\mathcal{O}$ is any maximal subset $tp \subseteq sub_{\mathcal{O}}$ 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 at $n \in \mathbb{N}$ in $\mathcal{I}$. For $\mathcal{O}$ consistent with $\mathcal{D}$, we abbreviate $tp_{\mathcal{C}_{\mathcal{O},\mathcal{D}}}$ to $tp_{\mathcal{O},\mathcal{D}}$. The canonical models have a periodic structure in the following sense:

Proposition 9. *For any $LTL^{\Box\bigcirc}_{horn}$ ontology $\mathcal{O}$ and any data instance $\mathcal{D}$ consistent with $\mathcal{O}$, there are $s_{\mathcal{O},\mathcal{D}} \le 2^{|\mathcal{O}|}$ and $p_{\mathcal{O},\mathcal{D}} \le 2^{2|\mathcal{O}|}$ such that $tp_{\mathcal{O},\mathcal{D}}(n) = tp_{\mathcal{O},\mathcal{D}}(n + p_{\mathcal{O},\mathcal{D}})$, for all $n \ge \max \mathcal{D} + s_{\mathcal{O},\mathcal{D}}$. Deciding $\mathcal{C}_{\mathcal{O},\mathcal{D}} \models \xi(\ell)$, for a binary ℓ and a conjunction of atoms ξ, is in* PSPACE/P *for combined/data complexity.*

We now show that the combined complexity of QBE with $\Diamond$- and $\bigcirc, \Diamond$-queries is PSPACE-complete in both bounded and unbounded cases, i.e., as complex as $LTL^{\Box\bigcirc}_{horn}$ reasoning.

Theorem 10. *Let $\mathcal{Q} \in \{\mathcal{Q}[\bigcirc, \Diamond], \mathcal{Q}[\Diamond], \mathcal{Q}_p[\bigcirc, \Diamond], \mathcal{Q}_p[\Diamond]\}$. Then $\mathsf{QBE}(LTL^{\Box\bigcirc}_{horn}, \mathcal{Q})$ and $\mathsf{QBE}^{\mathsf{b+}}_{\mathsf{b-}}(LTL^{\Box\bigcirc}_{horn}, \mathcal{Q})$ are both* PSPACE-*complete for combined complexity.*

Proof. PSPACE-hardness is inherited from that of $LTL^{\Box\bigcirc}_{horn}$. We briefly sketch the proof of the matching upper bound for $\mathcal{Q}[\bigcirc, \Diamond]$ using the reduction of Theorem 4 $(i.3)$. We can assume that $\mathcal{O}$ and $\mathcal{D}$ are consistent for any $\mathcal{D} \in E^+ \cup E^-$. For if $\mathcal{O}$ and $\mathcal{D} \in E^-$ are inconsistent, then E is not $\mathcal{Q}$-separable under $\mathcal{O}$ as $\mathcal{O}, \mathcal{D} \models \varkappa(0)$ for all $\varkappa \in \mathcal{Q}$; if $\mathcal{O}$ and $\mathcal{D} \in E^+$ are inconsistent, then E is separable iff $(E^+ \setminus \{\mathcal{D}\}, E^-)$ is. Checking consistency is known to be PSPACE-complete.

Given an $LTL^{\Box\bigcirc}_{horn}$-ontology $\mathcal{O}$ and an example set E, let

$$k = \max_{\mathcal{D} \in E^+ \cup E^-} (\max \mathcal{D} + s_{\mathcal{O},\mathcal{D}}), \quad m = \prod_{\mathcal{D} \in E^+ \cup E^-} p_{\mathcal{O},\mathcal{D}},$$

where $s_{\mathcal{O},\mathcal{D}}$ and $p_{\mathcal{O},\mathcal{D}}$ in $\mathcal{C}_{\mathcal{O},\mathcal{D}}$ are from Proposition 9. We show that if E is $\mathcal{Q}[\bigcirc, \Diamond]$-separable under $\mathcal{O}$, then it is separated by a conjunction of $|E^-|$-many $\varkappa \in \mathcal{Q}^{\circ}_p[\Diamond]$ of $\Diamond$-depth $\leq k+1$ and $\bigcirc$-depth $\leq k+m$ in (4). Indeed, in this case any $(E^+, \{\mathcal{D}\})$, for $\mathcal{D} \in E^-$, is separated under $\mathcal{O}$ by some $\varkappa$ of the form (4) with the ρ_l of $\bigcirc$-depth $\leq k+m$ because $\rho_l = \bigwedge_{i=0}^{\ell} \bigcirc^i \lambda_i$ with $\ell > k+m$ can be replaced by

$$\bigwedge_{i=0}^{k} \bigcirc^i \lambda_i \wedge \bigwedge_{j=1}^{m} \bigcirc^{k+j} \bigwedge_{i \leq \ell, j=(i-k) \bmod m} \lambda_i.$$

In addition, if $n > k$ in (4), then $(E^+, \{\mathcal{D}^-\})$ is separated by

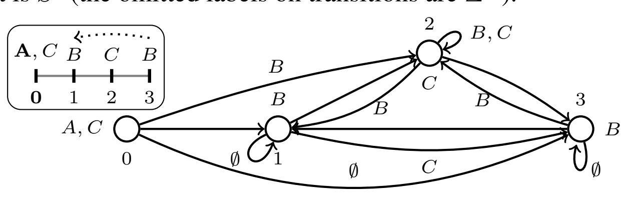
$$\rho_0 \wedge \Diamond(\rho_1 \wedge \Diamond(\rho_2 \wedge \cdots \wedge \Diamond\rho_k)) \wedge \bigwedge_{i=k+1}^{n} \Diamond^{k+1}\rho_i,$$

and so by some $\rho_0 \wedge \Diamond(\rho_1 \wedge \cdots \wedge \Diamond(\rho_k \wedge \Diamond\rho_j))$ with $k < j \leq n$. Our nondeterministic PSPACE-algorithm incrementally guesses the ρ_l and checks if they are satisfiable in the relevant part of the relevant $\mathcal{C}_{\mathcal{O},\mathcal{D}}$ bounded by $k+m$. $\dashv$

The situation is quite different for queries with U:

Theorem 11. $\mathsf{QBE}(LTL^{\Box\bigcirc}_{horn}, \mathcal{Q}[\mathsf{U}_s])$ *is in* EXPTIME *for combined complexity,* $\mathsf{QBE}(LTL^{\Box\bigcirc}_{horn}, \mathcal{Q}_p[\mathsf{U}])$ *is in* EXPSPACE*, and* $\mathsf{QBE}^{\mathsf{b+}}_{\mathsf{b-}}(LTL^{\Box\bigcirc}_{horn}, \mathcal{Q}_p[\mathsf{U}])$ *is* NEXPTIME*-hard.*

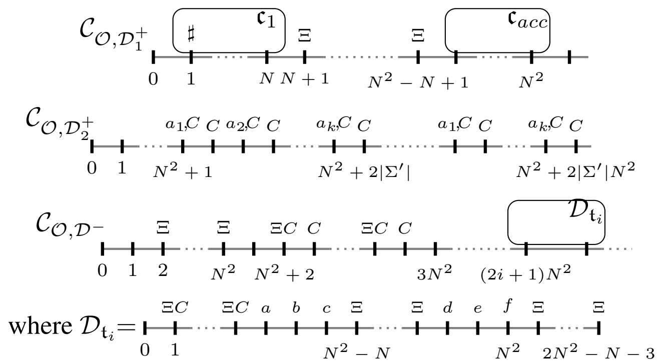
Proof. For the upper bounds, we again assume that $\mathcal{O}$ and $\mathcal{D}$ are consistent for all $\mathcal{D} \in E^+ \cup E^-$. Observe that Theorem 6 continues to hold in the presence of $LTL^{\Box\bigcirc}_{horn}$ ontologies $\mathcal{O}$ but we need a different construction of transition systems S^i that represent all $\mathcal{Q}[\mathsf{U}_s]$-queries mediated by $\mathcal{O}$ over $\mathcal{D}_i$. We illustrate it for $\mathcal{O} = \{A \to C \wedge \bigcirc B,\ B \to \bigcirc^2 B,\ B \to \bigcirc C\}$ and $\mathcal{D}_i = \{A(0)\}$ below, where the picture on the left shows the canonical model of $\mathcal{O}, \mathcal{D}_i$ (see Proposition 9) and next to it is S^i (the omitted labels on transitions are $\Sigma^{\perp}$).



In general, the size of S^i is $|\mathcal{D}_i| + O(2^{|\mathcal{O}|})$ and the product of S^i, $\mathcal{D}_i \in E^+$ is of size $O(2^{|\mathcal{O}|+|E^+|})$. The upper bounds now follow from P and PSPACE completeness of checking simulation and containment for transition systems.

Now we sketch the proof of the lower bound. Let $\boldsymbol{M}$ be a non-deterministic Turing machine that accepts Σ-words $\boldsymbol{x} = x_1 \dots x_n$ in $N = 2^{poly(|\boldsymbol{x}|)}$ steps and erases the tape after a successful computation. We represent configurations $\mathfrak{c}$ of a computation of $\boldsymbol{M}$ on $\boldsymbol{x}$ by an $N-1$-long word (with sufficiently many blanks at the end), in which y in the active cell is replaced by (q, y) with the current state $q \in Q$. An accepting computation of $\boldsymbol{M}$ on $\boldsymbol{x}$ is encoded by the N^2-long word $w = \sharp\mathfrak{c}_1 \sharp \mathfrak{c}_2 \sharp \dots \sharp \mathfrak{c}_{N-1} \sharp \mathfrak{c}_N$ over $\Xi = \Sigma \cup (Q \times \Sigma) \cup \{\sharp\}$. Thus, a word w of length N^2 encodes an accepting computation iff it starts with the initial configuration $\mathfrak{c}_1$ preceded by $\sharp$, ends with the accepting configuration $\mathfrak{c}_{acc}$, and every two length 3 subwords at distance N apart form a *legal tuple* [Sipser, 1997, Theorem 7.37].

We define $\mathcal{O}$ and $E = (\{\mathcal{D}_1^+, \mathcal{D}_2^+\}, \{\mathcal{D}^-\})$ so that their canonical models look as follows, for $\Xi = \{a_1, \dots, a_k\}$:



and $\mathfrak{t}_i = (a, b, c, d, e, f)$ is the lexicographically i-th illegal tuple. The parts of the canonical models shown above are of exponential size; however, due to their repetitive nature, they can be described by a polynomial-size $LTL^{\Box\bigcirc}_{horn}$ ontology $\mathcal{O}$ as in [Ryzhikov *et al.*, 2021]. We show that the $\mathcal{Q}_p[\mathsf{U}]$-query

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

where $\rho_1 \dots \rho_{N^2}$ encodes an accepting computation of $\boldsymbol{M}$ on $\boldsymbol{x}$, is the only type of query that can separate E under $\mathcal{O}$. $\dashv$

As for *data* complexity, we show that $LTL^{\Box\bigcirc}_{horn}$-ontologies come essentially for free:

Theorem 12. *The results of Theorem 5 continue to hold for queries mediated by a fixed* $LTL^{\Box\bigcirc}_{horn}$*-ontology.*

Intuitively, the reason is that, given a fixed $LTL^{\Box\bigcirc}_{horn}$-ontology $\mathcal{O}$, we can compute the types of the canonical model $\mathcal{C}_{\mathcal{O},\mathcal{D}}$, for consistent $\mathcal{O}$ and $\mathcal{D}$, in polynomial time in $\mathcal{D}$ by Proposition 9, with the length M from Section 4 being polynomial in E. Checking consistency of $\mathcal{D}$ and fixed $\mathcal{O}$ is known to be in P [Artale *et al.*, 2021].

6 QBE with $LTL^{\Box\Diamond}$-Ontologies

In this section, we investigate separability by $\Diamond$-queries under $LTL^{\Box\Diamond}$-ontologies. Remarkably, we show that, for data complexity, $LTL^{\Box\Diamond}$-ontologies also come for free despite admitting arbitrary Boolean operators; cf., [Schaerf, 1993].

Theorem 13. *Let* $\mathcal{Q} \in \{\mathcal{Q}_p[\Diamond], \mathcal{Q}[\Diamond]\}$*. If* E *is* $\mathcal{Q}$*-separable under an* $LTL^{\Box\Diamond}$*-ontology* $\mathcal{O}$*, then* E *can be separated under* $\mathcal{O}$ *by a* $\mathcal{Q}$*-query of polysize in* E *and* $\mathcal{O}$*.* $\mathsf{QBE}(LTL^{\Box\Diamond}, \mathcal{Q})$ *and* $\mathsf{QBE}^{\mathsf{b+}}_{\mathsf{b-}}(LTL^{\Box\Diamond}, \mathcal{Q})$ *are* Σ^p_2*-complete for combined complexity. The presence of* $LTL^{\Box\Diamond}$*-ontologies has no effect on the data complexity, which remains the same as in Theorem 5.*

We comment on the proof of this theorem for $\mathcal{Q}_p[\Diamond]$. Taking into account NP-completeness of checking if $\mathcal{O}$ is consistent with $\mathcal{D}$ and tractability of this problem for a fixed $\mathcal{O}$ [Artale *et al.*, 2021], we can assume, as in Theorem 10, that $\mathcal{O}$ and $\mathcal{D}$ are consistent for each $\mathcal{D} \in E^+ \cup E^-$. Observe first that if E is separated by $\varkappa \in \mathcal{Q}_p[\Diamond]$ of the form (1) under an $LTL^{\Box\Diamond}$-ontology $\mathcal{O}$, then, as follows from [Ono and Nakamura, 1980], for any $\mathcal{D} \in E^-$, there is a model $\mathcal{J}_\mathcal{D} \not\models \varkappa(0)$ of $\mathcal{O}$ and $\mathcal{D}$ whose types form a sequence

$$tp_0, \dots, tp_k, tp_{k+1}, \dots, tp_{k+l}, \dots, tp_{k+1}, \dots, tp_{k+l}, \dots \quad (5)$$

with $\max \mathcal{D} \leq k \leq \max \mathcal{D} + |\mathcal{O}|$ and $l \leq |\mathcal{O}|$. This allows us to find a separating $\varkappa$ of polysize in E, $\mathcal{O}$. Indeed, let K be the maximal k in (5) over all $\mathcal{D} \in E^-$. If the depth n of $\varkappa$ is $\leq K$, we are done. If $n > K$, we shorten $\varkappa$ as follows. Consider the prefix $\varkappa'$ of $\varkappa$ formed by $\rho_0, \dots, \rho_K$. If $\mathcal{J}_\mathcal{D} \not\models \varkappa'(0)$ for all $\mathcal{D} \in E^-$, we are done. Otherwise, for each $\mathcal{D} \in E^-$, we pick a ρ_i, $i > K$, with $\rho_i \not\subseteq tp_{k+j}$ for any $j \leq l$; it must exist as $\mathcal{J}_\mathcal{D} \not\models \varkappa(0)$. 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, $\varkappa''$ is as required.

A Σ_2^p-algorithm guesses $\varkappa$ and $\mathcal{J}_\mathcal{D}$, for $\mathcal{D} \in E^-$, and checks in polytime that $\mathcal{J}_\mathcal{D} \models \mathcal{O}, \mathcal{D}$ and $\mathcal{J}_\mathcal{D} \not\models \varkappa(0)$ and in CONP [Ono and Nakamura, 1980] that $\mathcal{O}, \mathcal{D} \models \varkappa(0)$ for all $\mathcal{D} \in E^+$. The lower bound is shown by reduction of the validity problem for fully quantified Boolean formulas $\exists \boldsymbol{p} \forall \boldsymbol{q}\, \psi$, where $\boldsymbol{p} = p_1, \dots, p_k$ and $\boldsymbol{q} = q_1, \dots, q_m$ are all propositional variables in ψ. We can assume that ψ is not a tautology and $\neg\psi \not\models x$ for $x \in \{p_i, \neg p_i, q_j, \neg q_j \mid i \leq k, j \leq m\}$. Let $E = (E^+, E^-)$ with $E^+ = \{\mathcal{D}_1, \mathcal{D}_2\}$, $E^- = \{\mathcal{D}_3\}$, where

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

and let $\mathcal{O}$ contain the following axioms with fresh atoms $B_1, B_2, A_i, \bar{A}_i$, for $i = 1, \dots, k$:

$$B_1 \vee B_2 \to \neg\psi, \quad p_i \to \Diamond\big(\bar{A}_i \wedge \textstyle\bigwedge_{j \neq i}(A_j \wedge \bar{A}_j)\big),$$
$$\neg p_i \to \Diamond\big(A_i \wedge \textstyle\bigwedge_{j \neq i}(A_j \wedge \bar{A}_j)\big).$$

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

We obtain the NP upper bounds in data complexity using the same argument as for the Σ_2^p-upper bound and observing that checking $\mathcal{O}, \mathcal{D} \models \varkappa(0)$ is in P in data complexity. The NP lower bounds are inherited from the ontology-free case. The proof of the P upper bounds is more involved. We illustrate the idea for $\mathcal{O}$ with arbitrary Boolean but without temporal operators. In this case, one can show (which is non-trivial) that $\mathcal{O}, \mathcal{D} \models \varkappa(0)$ iff $\mathcal{I}_{\mathcal{O},\mathcal{D}} \models \varkappa(0)$, where $\mathcal{I}_{\mathcal{O},\mathcal{D}}$ is the *completion* of $\mathcal{D}$: it contains $A(\ell)$ iff $\mathcal{O} \cup \{B \mid B(\ell) \in \mathcal{D}\} \models A$. For example, if $\mathcal{O} = \{A \vee B\}$ and $\mathcal{D} = \{A(1), B(1), A(3), B(3)\}$, the completion $\mathcal{I}_{\mathcal{O},\mathcal{D}}$ is just $\mathcal{D}$ regarded as an interpretation (so $\mathcal{I}_{\mathcal{O},\mathcal{D}}$ does not have to be a model of $\mathcal{O}$). It can be constructed in polytime in $\mathcal{D}$ and, due to the equivalence above, used to prove the P upper bounds using dynamic programming. That equivalence does not hold for $LTL^{\Box\Diamond}$, but the technique can be extended by applying it to data sets enriched by certain types.

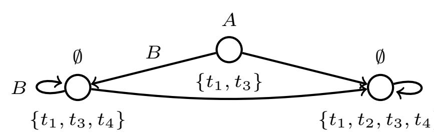
Note that the completion technique does not work for $\bigcirc$,$\Diamond$-queries. For example, $\mathcal{O}, \mathcal{D} \models \Diamond(A \wedge \bigcirc B)$ for $\mathcal{D}$ and $\mathcal{O}$ defined above, and so the equivalence does not hold. In fact, the complexity of separability by $\bigcirc\Diamond$-queries remains open.

7 QBE with *LTL*-Ontologies

For ontologies with arbitrary *LTL*-axioms, we obtain:

Theorem 14. (i) $\mathsf{QBE}(LTL, \mathcal{Q})$ *is in* 2EXPTIME*, for any* $\mathcal{Q} \in \{\, \mathcal{Q}[\Diamond], \mathcal{Q}[\bigcirc, \Diamond], \mathcal{Q}[\mathsf{U}_s] \,\}$. (ii) $\mathsf{QBE}(LTL, \mathcal{Q})$ *is in* 2EXPSPACE*, for any* $\mathcal{Q} \in \{\, \mathcal{Q}_p[\Diamond], \mathcal{Q}_p[\bigcirc, \Diamond], \mathcal{Q}_p[\mathsf{U}] \,\}$.

The proof requires a further modification of the transition systems S^i in Theorem 6. We illustrate it by an example. Let $\mathcal{O} = \{A \to \Diamond B,\ \top \to A \vee B,\ A \wedge B \to \bot\}$ with the set of $\mathcal{O}$-types $\boldsymbol{T}_\mathcal{O} = \{tp_1, tp_2, tp_3\}$, where $tp_1 = \{A, \neg B, \Diamond B\}$, $tp_2 = \{\neg A, B, \neg\Diamond B\}$, $tp_3 = \{A, \neg B, \neg\Diamond B\}$, and $tp_4 = \{\neg A, B, \Diamond B\}$, from which we omitted subformulas such as $A \vee B$ that are true or false in all types. For non-empty sets $\boldsymbol{T}_1, \boldsymbol{T}_2 \subseteq \boldsymbol{T}_\mathcal{O}$ and $\Gamma \subseteq \Sigma^\perp$, we take the relation $\boldsymbol{T}_1 \to_\Gamma \boldsymbol{T}_2$, which, intuitively, says that if there are instants $n_\mathcal{I}$ in all models $\mathcal{I}$ of $\mathcal{O}, \mathcal{D}$ such that $\{tp_\mathcal{I}(n_\mathcal{I}) \mid \mathcal{I} \models \mathcal{O}, \mathcal{D}\} = \boldsymbol{T}_1$, then there exist $m_\mathcal{I} > n_\mathcal{I}$ with $\{tp_\mathcal{I}(m_\mathcal{I}) \mid \mathcal{I} \models \mathcal{O}, \mathcal{D}\} = \boldsymbol{T}_2$ and $\Gamma = \{A \in \Sigma^\perp \mid \mathcal{I}, m \models A$ for all $\mathcal{I}$ and $n_\mathcal{I} < m < m_\mathcal{I}\}$. In our example, we have $\{tp_1, tp_3\} \to_{\Sigma^\perp} \{tp_1, tp_2, tp_3, tp_4\}$ and $\{tp_1, tp_3\} \to_{\{B\}} \{tp_1, tp_3, tp_4\}$ (among others). Then we construct the following transition system S^i for, say, $\mathcal{D}_i = \{A(0)\}$, which reflects all $\mathcal{Q}[\mathsf{U}_s]$-queries over $\mathcal{O}, \mathcal{D}_i$ using $\boldsymbol{T}' \subseteq \boldsymbol{T}_\mathcal{O}$ as states (the initial state is $\{tp_1, tp_3\}$ since $A(0) \in \mathcal{D}_i$):



The S^i can be constructed in 2EXPTIME in $|\mathcal{D}_i| + |\mathcal{O}|$ (checking $\boldsymbol{T}_1 \to_\Gamma \boldsymbol{T}_2$, for given $\boldsymbol{T}_1, \boldsymbol{T}_2$ and Γ, can be done in EXPSPACE). Also, the product of the S^i, for $\mathcal{D}_i \in E^+$, can be constructed in 2EXPTIME in $|\mathcal{D}_i| + |E^+|$.

8 Conclusions

We have started an investigation of the computational complexity of query-by-example for principal classes of *LTL*-queries, both with and without mediating ontologies. Our results are encouraging as we exhibit important cases that are tractable for data complexity and not harder than satisfiability for combined complexity. Many interesting and technically challenging problems remain open. Especially intriguing are queries with U. For example, we still need to pinpoint the size of minimal separating $\mathcal{Q}[\mathsf{U}_s]$- and $\mathcal{Q}_p[\mathsf{U}]$-queries under a Horn ontology. The tight complexity of QBE for unrestricted U-queries is also open. In general, such queries could be too perplexing for applications; however, they can express useful disjunctive patterns such as 'in at most n moments of time'.

Our results and techniques provide a good starting point for studying QBE with (ontology-mediated) queries over temporal databases with a full relational component [Chomicki *et al.*, 2001; Chomicki and Toman, 2018; Artale *et al.*, 2022] and also for the construction of separating queries satisfying additional conditions such as being a longest/shortest separator [Blum *et al.*, 2021; Fijalkow and Lagarde, 2021] or a most specific/general one [ten Cate *et al.*, 2022].

Acknowledgements

This work was supported by EPSRC UK grants EP/S032207, EP/S032282, and EP/W025868.

References

[Arenas *et al.*, 2016] M. Arenas, G. I. Diaz, and E. Kostylev. Reverse engineering SPARQL queries. In *Proc. of WWW*, pages 239–249, 2016.

[Artale *et al.*, 2021] A. Artale, R. Kontchakov, A. Kovtunova, V. Ryzhikov, F. Wolter, and M. Zakharyaschev. First-order rewritability of ontology-mediated queries in linear temporal logic. *Artif. Intell.*, 299:103536, 2021.

[Artale *et al.*, 2022] A. Artale, R. Kontchakov, A. Kovtunova, V. Ryzhikov, F. Wolter, and M. Zakharyaschev. First-order rewritability and complexity of two-dimensional temporal ontology-mediated queries. *J. Artif. Intell. Res.*, 75:1223–1291, 2022.

[Badea and Nienhuys-Cheng, 2000] L. Badea and S. Nienhuys-Cheng. A refinement operator for description logics. In *Proc. of ILP*, pages 40–59, 2000.

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

[Bergroth *et al.*, 2000] L. Bergroth, H. Hakonen, and T. Raita. A survey of longest common subsequence algorithms. In *Proc. of SPIRE*, pages 39–48, 2000.

[Blum *et al.*, 2021] C. Blum, M. Djukanovic, A. Santini, H. Jiang, C. Li, F. Manyà, and G. Raidl. Solving longest common subsequence problems via a transformation to the maximum clique problem. *Comput. Oper. Res.*, 125:105089, 2021.

[Bühmann *et al.*, 2016] L. Bühmann, J. Lehmann, and P. Westphal. DL-learner - A framework for inductive learning on the semantic web. *J. Web Sem.*, 39:15–24, 2016.

[Camacho and McIlraith, 2019] A. Camacho and S. McIlraith. Learning interpretable models expressed in linear temporal logic. In *Proc. of ICAPS*, pages 621–630, 2019.

[Chen and Lin, 1994] C.-C. Chen and I-Peng Lin. The computational complexity of the satisfiability of modal Horn clauses for modal propositional logics. *Theor. Comp. Sci.*, 129(1):95–121, 1994.

[Chomicki and Toman, 2018] J. Chomicki and D. Toman. Temporal logic in database query languages. In *Encyclopedia of Database Systems*. Springer, 2018.

[Chomicki *et al.*, 2001] J. Chomicki, D. Toman, and M. Böhlen. Querying ATSQL databases with temporal logic. *ACM Trans. Database Syst.*, 26(2):145–178, 2001.

[Chowdhury *et al.*, 2010] R. Chowdhury, H.-S. Le, and V. Ramachandran. Cache-oblivious dynamic programming for bioinformatics. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 7(3):495–510, 2010.

[Cima *et al.*, 2021] G. Cima, F. Croce, and M. Lenzerini. Query definability and its approximations in ontology-based data management. In *Proc. of CIKM*, pages 271–280. ACM, 2021.

[Cohen and Weiss, 2016] S. Cohen and Y. Weiss. The complexity of learning tree patterns from example graphs. *ACM Trans. Database Syst.*, 41(2):14:1–14:44, 2016.

[Demri *et al.*, 2016] S. Demri, V. Goranko, and M. Lange. *Temporal Logics in Computer Science*. Cambridge University Press, 2016.

[Deutch and Gilad, 2019] D. Deutch and A. Gilad. Reverse-engineering conjunctive queries from provenance examples. In *Proc. of EDBT*, pages 277–288, 2019.

[Fanizzi *et al.*, 2018] N. Fanizzi, G. Rizzo, C. d'Amato, and F. Esposito. DLFoil: Class expression learning revisited. In *Proc. of EKAW*, pages 98–113, 2018.

[Fijalkow and Lagarde, 2021] N. Fijalkow and G. Lagarde. The complexity of learning linear temporal formulas from examples. In *Proc. of 15th Int. Conf. on Grammatical Inference*, volume 153 of *PMLR*, pages 237–250, 2021.

[Fortin *et al.*, 2022] M. Fortin, B. Konev, V. Ryzhikov, Y. Savateev, F. Wolter, and M. Zakharyaschev. Unique characterisability and learnability of temporal instance queries. In *Proc. of KR*, 2022.

[Fortin *et al.*, 2023] M. Fortin, B. Konev, V. Ryzhikov, Y. Savateev, F. Wolter, and M. Zakharyaschev. Reverse engineering of temporal queries mediated by LTL ontologies. CoRR, abs/2303.04598, 2023.

[Fraser, 1996] C. Fraser. Consistent subsequences and supersequences. *Theor. Comput. Sci.*, 165(2):233–246, 1996.

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

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

[Gutiérrez-Basulto *et al.*, 2018] V. Gutiérrez-Basulto, J. Ch. Jung, and L. Sabellek. Reverse engineering queries in ontology-enriched systems: The case of expressive Horn description logic ontologies. In *Proc. of IJCAI*, 2018.

[Heindorf *et al.*, 2022] S. Heindorf, L. Blübaum, N. Düsterhus, T. Werner, V. N. Golani, C. Demir, and A.-C. Ngonga Ngomo. Evolearner: Learning description logics with evolutionary algorithms. In *Proc. of WWW*, pages 818–828. ACM, 2022.

[Hofman and Martens, 2015] P. Hofman and W. Martens. Separability by short subsequences and subwords. In *Proc. of ICDT*, volume 31 of *LIPIcs*, pages 230–246, 2015.

[Iannone *et al.*, 2007] L. Iannone, I. Palmisano, and N. Fanizzi. An algorithm based on counterfactuals for concept learning in the Semantic Web. *Appl. Intell.*, 26(2):139–159, 2007.

[Jung *et al.*, 2020] J. Ch. Jung, C. Lutz, and F. Wolter. Least General Generalizations in Description Logic: Verification and Existence. In *Proc. of AAAI*, 2020.

[Jung *et al.*, 2021] J. Ch. Jung, C. Lutz, H. Pulcini, and F. Wolter. Separating data examples by description logic concepts with restricted signatures. In *Proc. of KR*, 2021.

[Jung *et al.*, 2022] J. Ch. Jung, C. Lutz, H. Pulcini, and F. Wolter. Logical separability of labeled data examples under ontologies. *Artif. Intell.*, 313 (2022).

[Kalashnikov *et al.*, 2018] D. Kalashnikov, L. Lakshmanan, and D. Srivastava. Fastqre: Fast query reverse engineering. In *Proc. of SIGMOD*, pages 337–350, 2018.

[Kupferman and Vardi, 1996] O. Kupferman and M. Vardi. Verification of fair transisiton systems. In *Proc. of CAV, volume 1102 of* LNCS, pages 372–382. Springer, 1996.

[Kurtonina and de Rijke, 1997] N. Kurtonina and M. de Rijke. Bisimulations for temporal logic. *J. Log. Lang. Inf.*, 6(4):403–425, 1997.

[Lehmann and Haase, 2009] J. Lehmann and Ch. Haase. Ideal downward refinement in the $\mathcal{EL}$ description logic. In *Proc. of ILP*, pages 73–87, 2009.

[Lehmann and Hitzler, 2010] J. Lehmann and P. Hitzler. Concept learning in description logics using refinement operators. *Machine Learning*, 78:203–250, 2010.

[Lemieux *et al.*, 2015] C. Lemieux, D. Park, and I. Beschastnikh. General LTL specification mining (T). In *Proc. of ASE*, pages 81–92. IEEE, 2015.

[Lisi and Straccia, 2015] F. Lisi and U. Straccia. Learning in description logics with fuzzy concrete domains. *Fundamenta Informaticae*, 140(3-4):373–391, 2015.

[Lisi, 2012] F. Lisi. A formal characterization of concept learning in description logics. In *Proc. of DL*, 2012.

[Maier, 1978] D. Maier. The complexity of some problems on subsequences and supersequences. *J. ACM*, 25(2):322–336, 1978.

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

[Neider and Gavran, 2018] D. Neider and I. Gavran. Learning linear temporal properties. In *Proc. of FMCAD*, pages 1–10. IEEE, 2018.

[Ono and Nakamura, 1980] H. Ono and A. Nakamura. On the size of refutation Kripke models for some linear modal and tense logics. *Studia Logica*, pages 325–333, 1980.

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

[Place and Zeitoun, 2016] T. Place and M. Zeitoun. Separating regular languages with first-order logic. *Log. Methods Comput. Sci.*, 12(1), 2016.

[Place and Zeitoun, 2022] T. Place and M. Zeitoun. A generic polynomial time approach to separation by first-order logic without quantifier alternation. In *Proc. of FSTTCS*, volume 250 of *LIPIcs*, pages 43:1–43:22, 2022.

[Prior, 1956] A. Prior. *Time and Modality*. OUP, 1956.

[Raha *et al.*, 2022] R. Raha, R. Roy, N. Fijalkow, and D. Neider. Scalable anytime algorithms for learning fragments of linear temporal logic. In *Proc. of TACAS*, volume 13243 of *LNCS*, pages 263–280. Springer, 2022.

[Rizzo *et al.*, 2018] G. Rizzo, N. Fanizzi, C. d'Amato, and F. Esposito. A framework for tackling myopia in concept learning on the web of data. In *Proc. of EKAW*, pages 338–354. Springer, 2018.

[Rizzo *et al.*, 2020] G. Rizzo, N. Fanizzi, and C. d'Amato. Class expression induction as concept space exploration: From DL-FOIL to DL-FOCL. *Future Gener. Comput. Syst.*, 108:256–272, 2020.

[Ryzhikov *et al.*, 2021] V. Ryzhikov, Y. Savateev, and M. Zakharyaschev. Deciding FO-rewritability of ontology-mediated queries in linear temporal logic. In *Proc. of TIME*, volume 206 of *LIPIcs*, pages 10:1–10:15, 2021.

[Sarker and Hitzler, 2019] Md. K. Sarker and P. Hitzler. Efficient concept induction for description logics. In *Proc. of AAAI*, pages 3036–3043, 2019.

[Schaerf, 1993] A. Schaerf. On the complexity of the instance checking problem in concept languages with existential quantification. *J. Intel. Inf. Sys.*, 2:265–278, 1993.

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

[Staworko and Wieczorek, 2012] S. Staworko and P. Wieczorek. Learning twig and path queries. In *Proc. of ICDT*, pages 140–154, 2012.

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

[ten Cate and Dalmau, 2015] B. ten Cate and V. Dalmau. The product homomorphism problem and applications. In *Proc. of ICDT*, pages 161–176, 2015.

[ten Cate *et al.*, 2022] B. ten Cate, V. Dalmau, M. Funk, and C. Lutz. Extremal fitting problems for conjunctive queries. *CoRR*, abs/2206.05080, 2022.

[Tran *et al.*, 2017] An C. Tran, J. Dietrich, H. W. Guesgen, and S. Marsland. Parallel symmetric class expression learning. *J. Mach. Learn. Res.*, 18:64:1–64:34, 2017.

[Vardi, 2007] M. Vardi. Automata-theoretic techniques for temporal reasoning. In *Handbook of Modal Logic*, pages 971–989. North-Holland, 2007.

[Weiss and Cohen, 2017] Y. Weiss and S. Cohen. Reverse engineering SPJ-queries from examples. In *Proc. of PODS*, pages 151–166, 2017.

[Willard, 2010] R. Willard. Testing expressibility is hard. In *Proc. of CP*, vol. 6308 of *LNCS*, pp. 9–23. Springer, 2010.

[Zhang *et al.*, 2013] M. Zhang, H. Elmeleegy, C. Procopiuc, and D. Srivastava. Reverse engineering complex join queries. In *Proc. of SIGMOD*, pages 809–820, 2013.

[Zloof, 1977] M. Zloof. Query-by-example: A data base language. *IBM Syst. J.*, 16(4):324–343, 1977.