Unique Characterisability and Learnability of Temporal Instance Queries

M. Fortin[1] , **B. Konev**[1] , **V. Ryzhikov**[2] , **Y. Savateev**[2,3] , **F. Wolter**[1] , **M. Zakharyaschev**[2]

[1]Department of Computer Science, University of Liverpool, UK
[2]Department of Computer Science, Birkbeck, University of London, UK
[3]HSE University, Moscow, Russia

{marie.fortin,boris.konev,wolter@liverpool.ac.uk}, {vlad,yury,michael}@dcs.bbk.ac.uk

Abstract

We aim to determine which temporal instance queries can be uniquely characterised by a (polynomial-size) set of positive and negative temporal data examples. We start by considering queries formulated in fragments of propositional linear temporal logic *LTL* that correspond to conjunctive queries (CQs) or extensions thereof induced by the until operator. Not all of these queries admit polynomial characterisations, but by imposing a further restriction to path-shaped queries we identify natural classes that do. We then investigate how far the obtained characterisations can be lifted to temporal knowledge graphs queried by 2D languages combining *LTL* with concepts in description logics $\mathcal{EL}$ or $\mathcal{ELI}$ (i.e., tree-shaped CQs). While temporal operators in the scope of description logic constructors can destroy polynomial characterisability, we obtain general transfer results for the case when description logic constructors are within the scope of temporal operators. We finally apply our characterisations to establish (polynomial) learnability of temporal instance queries using membership queries in the active learning framework.

1 Introduction

Constructing queries or, more generally, logical concepts describing individuals of interest, can be difficult. Supporting a user addressing this problem has been a major research topic in databases, logic, and knowledge representation. For instance, in reverse engineering of database queries and concept descriptions (Martins 2019; Lehmann and Hitzler 2010; Jung et al. 2020), one aims to identify a query using a set of positively and negatively labelled examples of answers and non-answers, respectively; and in active learning approaches, one aims to identify a query by asking an oracle (e.g., domain specialist) whether an example is an answer or a non-answer to the query (Angluin, Frazier, and Pitt 1992; Funk, Jung, and Lutz 2021; ten Cate and Dalmau 2021).

Recently, the *unique characterisation* of a query by a finite (ideally, polynomial-size) set of positive and negative example answers has been identified as a fundamental link between queries and data (ten Cate and Dalmau 2021). Namely, we say that a query $\boldsymbol{q}$ *fits* a pair $E = (E^+, E^-)$ of sets E^+ and E^- of pointed databases $(\mathcal{D}, a)$ if $\mathcal{D} \models \boldsymbol{q}(a)$ for $(\mathcal{D}, a) \in E^+$, and $\mathcal{D} \not\models \boldsymbol{q}(a)$ for $(\mathcal{D}, a) \in E^-$. Then E *uniquely characterises* $\boldsymbol{q}$ within a class $\mathcal{Q}$ of queries if $\boldsymbol{q}$ is the only (up to equivalence) query in $\mathcal{Q}$ that fits E.

Unique (polynomial) characterisations can be used to illustrate, explain, and construct queries. They are also a 'non-procedural' necessary condition for (polynomial) learnability using membership queries in Angluin's (1987b) framework of active learning, where membership queries to the oracle take the form 'does $\mathcal{D} \models \boldsymbol{q}(a)$ hold?'. It is shown by ten Cate and Dalmau (2021) that, for classes of conjunctive queries (CQs), it is often a sufficient condition as well.

In many applications, queries are required to capture the temporal evolution of individuals, making their formulation even harder. The aim of this paper is to start an investigation of the (polynomial) characterisability of temporal instance queries. We first consider one-dimensional data instances of the form $(\delta_0, \dots, \delta_n)$, where δ_i is the set of atomic propositions that are true at timestamp i, describing the temporal behaviour of a single individual, and queries formulated in fragments of propositional linear temporal logic *LTL*. Although rather basic as a temporal data model, this restriction allows us to focus on the purely temporal aspect of unique characterisability. We then generalise our results, where possible, to standard two-dimensional temporal data instances, in which the δ_i are replaced by non-temporal data instances and queries are obtained by combining fragments of *LTL* with $\mathcal{ELI}$-concept queries (or tree-shaped CQs), thereby combining a well established formalism for accessing temporal data (Chomicki and Toman 2018) with the basic concept descriptions for tractable data access from description logic (Baader et al. 2017).

Our initial observation is that already very primitive temporal queries are not uniquely characterisable. For example, consider the query $\boldsymbol{q} = \Diamond_r(A \wedge B)$ with the operator $\Diamond_r$ 'now or later' (interpreted by $\leq$ over linearly ordered timestamps). By the pigeonhole principle, no finite example set E can distinguish $\boldsymbol{q}$ from a query $\boldsymbol{q}' = \Diamond_r(A \wedge (\Diamond_r B \wedge \Diamond_r(A \wedge \dots)))$ with sufficiently many alternating A and B. Similarly, the query $\boldsymbol{q} = \bigcirc A$ with the 'next time' operator $\bigcirc$ is not distinguishable by a finite example set from $\boldsymbol{q}' = (\bigcirc \dots \bigcirc A) \,\mathsf{U}\, A$ with the strict 'until' operator U and sufficiently many $\bigcirc$ on its left-hand side.

Aiming to identify natural and useful classes of temporal queries enjoying (polynomial) characterisability, in this paper we consider the conjunctive fragment of *LTL*. To begin with, we focus on two classes of path CQs: the class

$\mathcal{Q}_p[\bigcirc, \Diamond_r]$ of queries of the form

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

where $\boldsymbol{o}_i \in \{\bigcirc, \Diamond_r\}$ and ρ_i is a conjunction of atomic propositions, and also the class $\mathcal{Q}_p^\sigma[\mathsf{U}]$ of U-queries of the form

$$\boldsymbol{q} = \rho_0 \wedge (\lambda_1 \,\mathsf{U}\, (\rho_1 \wedge (\lambda_2 \,\mathsf{U}\, (\ldots (\lambda_n \,\mathsf{U}\, \rho_n) \ldots)))), \quad (2)$$

where λ_i is a conjunction of atoms or $\bot$. The superscript σ in $\mathcal{Q}_p^\sigma[\mathsf{U}]$ indicates that queries are formulated in a finite signature σ, a condition required because of the universal quantification implicit in U. Our first main result is a syntactic criterion of (polynomial) characterisability of $\mathcal{Q}_p[\bigcirc, \Diamond_r]$-queries. In fact, it turns out that the query $\Diamond_r(A \wedge B)$ mentioned above epitomises the cause of non-characterisability in $\mathcal{Q}_p[\bigcirc, \Diamond_r]$. It follows, in particular, that the restriction $\mathcal{Q}_p[\bigcirc, \Diamond]$ of $\mathcal{Q}_p[\bigcirc, \Diamond_r]$ to queries with $\bigcirc$ and strict eventuality $\Diamond = \bigcirc\Diamond_r$ is polynomially characterisable. Our second main result is that all $\mathcal{Q}_p^\sigma[\mathsf{U}]$-queries with $\subseteq$-incomparable λ_i and ρ_i, for each i, are polynomially characterisable within $\mathcal{Q}_p^\sigma[\mathsf{U}]$. Although we show that all $\mathcal{Q}_p^\sigma[\mathsf{U}]$-queries are exponentially characterisable, it remains open whether they are polynomially characterisable in $\mathcal{Q}_p^\sigma[\mathsf{U}]$.

The essential property that distinguishes $\mathcal{Q}_p[\bigcirc, \Diamond_r]$ and $\mathcal{Q}_p^\sigma[\mathsf{U}]$ from other queries is that they do not admit temporal branching as, for instance, in $\Diamond A \wedge \Diamond B$. In fact, we show that even within the class of queries using only $\wedge$ and $\Diamond$ and with a bound on the number of branches, not all queries are polynomially characterisable. A first step towards positive results covering non-path queries is made for the case of queries in which all branches are of equal length.

Our next aim is to generalise the obtained results to 2D temporal queries combining *LTL* and the descriptions logic constructor $\exists P$ of $\mathcal{ELI}$. Our first main result is negative: even queries of the form $\exists P.\boldsymbol{q}_1 \wedge \cdots \wedge \exists P.\boldsymbol{q}_n$ with $\boldsymbol{q}_i \in \mathcal{Q}_p[\bigcirc, \Diamond]$ are not polynomially characterisable. The situation changes drastically, however, if we consider queries of the form (1) or (2), in which ρ_i and λ_i are $\mathcal{ELI}$-queries. Indeed, we generalise our polynomial characterisability results for $\mathcal{Q}_p[\bigcirc, \Diamond_r]$ and $\mathcal{Q}_p^\sigma[\mathsf{U}]$ to such queries using recent results on the computation of frontiers in the lattice of $\mathcal{ELI}$-queries (ten Cate and Dalmau 2021) and proving a new result on split partners in the lattice of $\mathcal{EL}$-queries (where $\mathcal{EL}$ denotes $\mathcal{ELI}$ without inverse roles).

Finally, we discuss applications of our results to learning temporal instance queries using membership queries of the form 'does $\mathcal{D} \models \boldsymbol{q}$ hold?'. As we always construct example sets effectively, our unique (exponential) characterisability results imply (exponential-time) learnability with membership queries. Obtaining polynomial-time learnability from polynomial charcterisations is more challenging and, in fact, not always possible. A main result here is that $\mathcal{Q}_p[\bigcirc, \Diamond_r]$ with $\mathcal{ELI}$-queries is polynomial-time learnable with membership queries.

Omitted proofs can be found in the appendix to the full version of the paper.

2 Related Work

Our contribution is closely related to work on active learnability of formal languages and on learning temporal logic formulas interpreted over finite and infinite traces. It is also related to learning database queries and other formal expressions for accessing data. In the former area, the seminal paper by Angluin (1987a) has given rise to a large body of work on active learning of regular languages or variations, for example, (Shahbaz and Groz 2009; Aarts and Vaandrager 2010; Cassel et al. 2016; Howar and Steffen 2018). This work has mainly focused on learning various types of finite state machines or automata using a combination of membership queries with other powerful types of queries such as equivalence queries. The use of two or more types of queries is motivated by the fact that otherwise one cannot efficiently learn a wide variety of important languages, including regular languages. In fact, the main difference between this work and our contribution is that we focus on queries for which the corresponding formal languages form only a small subset of the regular languages and it is this restriction that enables us to focus on characterisability and learnability with membership queries.

Rather surprisingly, there has hardly been any work on active learning of temporal formulas over finite or infinite traces; we refer the reader to (Camacho and McIlraith 2019), also for a discussion of the relationship between learning automata and *LTL*-formulas. In contrast, passive learning of *LTL*-formulas has recently received significant attention; see (Lemieux, Park, and Beschastnikh 2015; Neider and Gavran 2018; Fijalkow and Lagarde 2021) and, in the context of explainable AI, also (Camacho and McIlraith 2019) for an overview.

In the database and KR communities, there has been extensive work on identifying queries and concept descriptions from data examples. For instance, in reverse engineering of queries, the goal is typically to decide whether there is a query that fits (or separates) a set of positive and negative examples. Relevant work under the closed world assumption include (Arenas and Diaz 2016; Barceló and Romero 2017) and under the open world assumption (Gutiérrez-Basulto, Jung, and Sabellek 2018; Funk et al. 2019). Related work on active learning not yet discussed include the identification of $\mathcal{EL}$-queries (Funk, Jung, and Lutz 2021) and ontologies (Konev, Ozaki, and Wolter 2016; Konev et al. 2017), and of schema-mappings (ten Cate, Dalmau, and Kolaitis 2013; ten Cate et al. 2018). Again this work differs from our contribution as it focusses on learning using membership and equivalence queries rather than only the former. The use of unique characterisations to explain and construct schema mappings has been promoted and investigated in (Kolaitis 2011; Alexe et al. 2011).

Combining *LTL* and description logics for temporal conceptual modeling and data access has a long tradition (Lutz, Wolter, and Zakharyaschev 2008; Artale et al. 2017). For querying purposes sometimes description logic concepts have been replaced by general CQs. Our restriction to $\mathcal{ELI}$-concepts instead of general CQs is motivated by results of (ten Cate and Dalmau 2021) showing that only CQs that are acyclic modulo cycles through the answer variables are polynomially characterisable within the class of CQs. Hence very strong acyclicity conditions are needed to ensure polynomial characterisability. We conjecture that our results can

be extended to this class.

The class of queries in which no $\exists P$ is within the scope of temporal operators was first introduced by (Baader, Borgwardt, and Lippmann 2015; Borgwardt and Thost 2015) in the context of monitoring applications.

3 Preliminaries

By a *signature* we mean any finite set $\sigma \neq \emptyset$ of *atomic concepts* $A, B, C, \dots$ representing observations, measurements, events, etc. A *σ-data instance* is any finite sequence $\mathcal{D} = (\delta_0, \dots, \delta_n)$ with $\delta_i \subseteq \sigma$, saying that $A \in \delta_i$ happened at moment i. The *length* of $\mathcal{D}$ is $\max(\mathcal{D}) = n$ and the *size* of $\mathcal{D}$ is $|\mathcal{D}| = \Sigma_{i \leq n} |\delta_i|$. We do not distinguish between $\mathcal{D}$ and its *variants* of the form $(\delta_0, \dots, \delta_n, \emptyset, \dots, \emptyset)$.

We access data by means of *queries*, $\boldsymbol{q}$, constructed from atoms, $\bot$ and $\top$ using $\wedge$ and the temporal operators $\bigcirc$, $\Diamond$, $\Diamond_r$ and U. The set of atomic concepts occurring in $\boldsymbol{q}$ is denoted by $sig(\boldsymbol{q})$. The set of queries that only use the operators from $\Phi \subseteq \{\bigcirc, \Diamond, \Diamond_r, \mathsf{U}\}$ is denoted by $\mathcal{Q}[\Phi]$; $\mathcal{Q}^\sigma[\Phi]$ is the restriction of $\mathcal{Q}[\Phi]$ to a signature σ. The *size* $|\boldsymbol{q}|$ of $\boldsymbol{q}$ is the number of symbols in $\boldsymbol{q}$, and the *temporal depth* $tdp(\boldsymbol{q})$ of $\boldsymbol{q}$ is the maximum number of nested temporal operators in $\boldsymbol{q}$.

$\mathcal{Q}[\bigcirc, \Diamond, \Diamond_r]$-queries can be equivalently defined as *tree-shaped* conjunctive queries (CQs) with the binary predicates *suc*, $<$, $\leq$ over $\mathbb{N}$, and atomic concepts as unary predicates. Each such CQ is a set $Q(t_0)$ of assertions of the form $A(t)$, $suc(t, t')$, $t < t'$, and $t \leq t'$, 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*, $<$, $\leq$.

The set of $\mathcal{Q}[\bigcirc, \Diamond, \Diamond_r]$-queries with *path-shaped* CQ counterparts is denoted by $\mathcal{Q}_p[\bigcirc, \Diamond, \Diamond_r]$. Such queries $\boldsymbol{q}$ take the form (1), where $\boldsymbol{o}_i \in \{\bigcirc, \Diamond, \Diamond_r\}$ and ρ_i is a conjunction of atoms (the empty conjunction is $\top$). Similarly, $\mathcal{Q}_p[\mathsf{U}]$-queries take the form (2).

Given a data instance $\mathcal{D} = (\delta_0, \dots, \delta_n)$, the *truth-relation* $\mathcal{D}, \ell \models \boldsymbol{q}$, for $\ell < \omega$, is defined as follows:

$$
\begin{aligned}
&\mathcal{D}, \ell \models A \text{ iff } A \in \delta_\ell, \quad \mathcal{D}, \ell \models \top, \quad \mathcal{D}, \ell \not\models \bot,\\
&\mathcal{D}, \ell \models \boldsymbol{q}_1 \wedge \boldsymbol{q}_2 \text{ iff } \mathcal{D}, \ell \models \boldsymbol{q}_1 \text{ and } \mathcal{D}, \ell \models \boldsymbol{q}_2,\\
&\mathcal{D}, \ell \models \bigcirc \boldsymbol{q} \text{ iff } \mathcal{D}, \ell + 1 \models \boldsymbol{q},\\
&\mathcal{D}, \ell \models \Diamond \boldsymbol{q} \text{ iff } \mathcal{D}, m \models \boldsymbol{q}, \text{ for some } m > \ell,\\
&\mathcal{D}, \ell \models \Diamond_r \boldsymbol{q} \text{ iff } \mathcal{D}, m \models \boldsymbol{q}, \text{ for some } m \geq \ell,\\
&\mathcal{D}, \ell \models \boldsymbol{q}_1 \,\mathsf{U}\, \boldsymbol{q}_2 \text{ iff there is } m > \ell \text{ such that } \mathcal{D}, m \models \boldsymbol{q}_2\\
&\qquad\qquad \text{and } \mathcal{D}, k \models \boldsymbol{q}_1, \text{ for all } k \text{ with } \ell < k < m.
\end{aligned}
$$

Note that $\mathcal{D}, n \models \Diamond\top \wedge \bigcirc\top \wedge (\boldsymbol{q}\,\mathsf{U}\,\top)$ as $(\delta_0, \dots, \delta_n, \emptyset)$ is a variant of $\mathcal{D}$. We write $\boldsymbol{q} \models \boldsymbol{q}'$ if $\mathcal{D}, \ell \models \boldsymbol{q}$ implies $\mathcal{D}, \ell \models \boldsymbol{q}'$, for any $\mathcal{D}$ and ℓ. If $\boldsymbol{q} \models \boldsymbol{q}'$ and $\boldsymbol{q}' \models \boldsymbol{q}$, we call $\boldsymbol{q}$ and $\boldsymbol{q}'$ *equivalent* and write $\boldsymbol{q} \equiv \boldsymbol{q}'$. Since $\bigcirc\boldsymbol{q} \equiv \bot\mathsf{U}\boldsymbol{q}$, $\Diamond\boldsymbol{q} \equiv \top\mathsf{U}\boldsymbol{q}$ and $\Diamond\boldsymbol{q} \equiv \bigcirc\Diamond_r\boldsymbol{q}$, one can assume that $\mathcal{Q}[\bigcirc, \Diamond] \subseteq \mathcal{Q}[\mathsf{U}]$, $\mathcal{Q}[\Diamond] \subseteq \mathcal{Q}[\bigcirc, \Diamond_r]$ and $\mathcal{Q}[\bigcirc, \Diamond_r] = \mathcal{Q}[\bigcirc, \Diamond_r, \Diamond]$.

4 Unique Characaterisability

An *example set* is a pair $E = (E^+, E^-)$ with finite sets E^+ and E^- of data instances that are called *positive* and *negative examples*, respectively. A query $\boldsymbol{q}$ *fits* E if $\mathcal{D}^+, 0 \models \boldsymbol{q}$ and $\mathcal{D}^-, 0 \not\models \boldsymbol{q}$, for all $\mathcal{D}^+ \in E^+$ and $\mathcal{D}^- \in E^-$. E *uniquely characterises* $\boldsymbol{q} \in \mathcal{Q}$ within a class $\mathcal{Q}$ of queries if $\boldsymbol{q}$ fits E and $\boldsymbol{q} \equiv \boldsymbol{q}'$ for any $\boldsymbol{q}' \in \mathcal{Q}$ that fits E. If all $\boldsymbol{q} \in \mathcal{Q}$ are characterised by some E within $\mathcal{Q}' \supseteq \mathcal{Q}$, we call $\mathcal{Q}$ *uniquely characterisable* within $\mathcal{Q}'$. Further, $\mathcal{Q}$ is *polynomially characterisable* within $\mathcal{Q}' \supseteq \mathcal{Q}$ if there is a polynomial f such that every $\boldsymbol{q} \in \mathcal{Q}$ is characterised within $\mathcal{Q}'$ by some E of size $|E| \leq f(|\boldsymbol{q}|)$, where $|E| = \Sigma_{\mathcal{D} \in E^+ \cup E^-} |\mathcal{D}|$. Let $\mathcal{Q}^n$ be the set of queries in $\mathcal{Q}$ of size at most n. We say that $\mathcal{Q}$ is *polynomially characterisable for bounded query size* if there is a polynomial f such that every $\boldsymbol{q} \in \mathcal{Q}^n$ is characterised by some E of size $\leq f(n)$ within $\mathcal{Q}^n$.

Observe that (polynomial) characterisability is anti-monotone: if a query $\boldsymbol{q}$ is (polynomially) characterisable within $\mathcal{Q}$ and $\mathcal{Q}' \subseteq \mathcal{Q}$, then $\boldsymbol{q}$ is (polynomially) characterisable within $\mathcal{Q}'$. In counterexamples to characterisability, we therefore only provide the smallest natural class of queries within which non-characterisability holds. The following examples illustrate (non-)characterisability within the classes $\mathcal{Q}_p[\Diamond_r]$ and $\mathcal{Q}_p[\bigcirc, \Diamond_r]$.

Example 1. (i) Recall from Section 1 that $\Diamond_r(A \wedge B)$ is not uniquely characterisable within $\mathcal{Q}_p[\Diamond_r]$. The same argument shows non-characterisability of $\Diamond(A \wedge B)$ within $\mathcal{Q}_p[\Diamond_r, \Diamond]$. On the other hand, the query $\Diamond(A \wedge B)$ is characterised within $\mathcal{Q}_p[\Diamond, \bigcirc]$ by the example set (E^+, E^-) with positive examples $E^+ = \{(\emptyset, \{A, B\}), (\emptyset, \emptyset, \{A, B\})\}$ and negative $E^- = \{(\emptyset, \{A\}), (\emptyset, \{B\})\}$.

(ii) The conjunction of atoms does not always lead to non-characterisability within classes of queries with $\Diamond_r$. For example, $\boldsymbol{q} = \Diamond_r(A \wedge \bigcirc(A \wedge B))$ is characterised within $\mathcal{Q}_p[\bigcirc, \Diamond_r]$ by $E = (E^+, E^-)$ with E^+ containing two data instances $(\{A\}, \{A, B\})$ and $(\emptyset, \{A\}, \{A, B\})$ and E^- also two instances:

$$(\emptyset, \emptyset, \{A, B\}), \quad (\emptyset, \{A\}, \{A\}, \{B\}, \{A, B\}).$$

The intuition here is that some instances from E^- have to satisfy the query $\Diamond_r(A \wedge \bigcirc(B \wedge \Diamond_r(A \wedge B)))$ as well as the query $\Diamond_r(A \wedge \bigcirc(A \wedge \Diamond_r(A \wedge B)))$.

(iii) While the query $\Diamond_r(A \wedge B)$ from (i) is not characterisable, there is a polynomial f such that, for all $n \in \mathbb{N}$, it is characterisable within $\mathcal{Q}_p^n[\bigcirc, \Diamond_r]$ by some E_n of size $\leq f(n)$. Namely, we take $E^+ = \{(\{A, B\}), (\emptyset, \{A, B\})\}$ and $E^- = \{(\underbrace{\{A\}, \{B\}, \dots, \{A\}, \{B\}}_{n \text{ times}})\}$.

Observe that one can always separate $\boldsymbol{q} \in \mathcal{Q}[\bigcirc, \Diamond_r]$ from any other $\boldsymbol{q}' \in \mathcal{Q}[\bigcirc, \Diamond_r]$ with $sig(\boldsymbol{q}') \supsetneq \text{sig}(\boldsymbol{q}) = \sigma$ using the positive example $(\sigma, \dots, \sigma)$ with $tdp(\boldsymbol{q})+1$-many copies of σ. One can therefore focus on characterisability within the relevant class of queries over the same signature as the input query. However, this not the case for $\mathcal{Q}[\mathsf{U}]$:

Example 2. The query $\boldsymbol{q} = \bot\,\mathsf{U}\,A \equiv \bigcirc A$ is not uniquely characterisable within $\mathcal{Q}_p[\mathsf{U}]$. Indeed, suppose $\boldsymbol{q}$ fits E and σ comprises all atoms occurring in E. Then $\mathcal{D}, 0 \models C\,\mathsf{U}\,A$ iff $\mathcal{D}, 0 \models \bigcirc A$, for all $\mathcal{D}$ in E and $C \notin \sigma$, and so E does not characterise $\boldsymbol{q}$. On the other hand, for the signature $\sigma = \{A, B\}$, the query $\boldsymbol{q}$ is characterised within $\mathcal{Q}_p^\sigma[\mathsf{U}]$ by the example set (E^+, E^-) in which $E^+ = \{(\emptyset, \{A\})\}$ and $E^- = \{(\sigma, \{B\}, \{A\})\}$ as $A\,\mathsf{U}\,A \equiv (A \wedge B)\,\mathsf{U}\,A \equiv \bigcirc A$.

As noted in Section 1, $\bot \,\mathsf{U}\, A$ is not uniquely characterisable within $\mathcal{Q}^{\{A\}}[\mathsf{U}]$ because of nested U-operators on the left-hand side of U. This observation prompts us to consider the subclass $\mathcal{Q}^\sigma_-[\mathsf{U}]$ of $\mathcal{Q}^\sigma[\mathsf{U}]$-queries $\boldsymbol{q}$ in which any subquery $\boldsymbol{q}' \,\mathsf{U}\, \boldsymbol{q}''$ does not contain occurrences of U in $\boldsymbol{q}'$. Note that $\mathcal{Q}^\sigma_p[\mathsf{U}] \subseteq \mathcal{Q}^\sigma_-[\mathsf{U}]$. We show that $\mathcal{Q}^\sigma_-[\mathsf{U}]$ is uniquely characterisable. To simplify notation, we give σ-data instances as *words* over the alphabet 2^σ using the standard notation of regular languages. Instead of $\mathcal{D}, 0 \models \boldsymbol{q}$ we simply write $\mathcal{D} \models \boldsymbol{q}$. By the semantics of U, for any $\boldsymbol{q} \in \mathcal{Q}^\sigma_-[\mathsf{U}]$, we have

$$\sigma^d \not\models \boldsymbol{q} \;\text{ for } d \leq \mathit{tdp}(\boldsymbol{q}), \quad \sigma^d \models \boldsymbol{q} \;\text{ for } d > \mathit{tdp}(\boldsymbol{q}) \qquad (3)$$

where σ^d is a word with d-many σ. Note also that there are finitely-many, say $N_d < \omega$, pairwise non-equivalent queries of any depth $d < \omega$ in $\mathcal{Q}^\sigma_-(\mathsf{U})$.

Lemma 3. *If $\boldsymbol{q}, \boldsymbol{q}' \in \mathcal{Q}^\sigma_-[\mathsf{U}]$ are of depth d and $\boldsymbol{q} \not\models \boldsymbol{q}'$, then there is $\mathcal{D}$ such that $\max(\mathcal{D}) \leq N_d$, $\mathcal{D} \models \boldsymbol{q}$ and $\mathcal{D} \not\models \boldsymbol{q}'$.*

Proof. Consider $\mathcal{D}$ of minimal length such that $\mathcal{D} \models \boldsymbol{q}$ and $\mathcal{D} \not\models \boldsymbol{q}'$. Let $\mathit{tp}(i)$ comprise all of the subqueries $\boldsymbol{s}$ of $\boldsymbol{q}$ and $\boldsymbol{q}'$ with $\mathcal{D}, i \models \boldsymbol{s}$. By the choice of $\mathcal{D}$, we have $\mathit{tp}(i) \neq \mathit{tp}(j)$ for any distinct $i, j \in [0, \max(\mathcal{D})]$ (otherwise we could cut the interval $[i, j)$ out of $\mathcal{D}$ to obtain a shorter instance separating $\boldsymbol{q}$ from $\boldsymbol{q}'$). It follows that $\max(\mathcal{D}) \leq N_d$. ❑

Theorem 4. *For any σ, $\mathcal{Q}^\sigma_-[\mathsf{U}]$ is uniquely characterisable.*

Proof. Any $\boldsymbol{q} \in \mathcal{Q}^\sigma_-(\mathsf{U})$ is uniquely characterised by E with

$$E^+ = \{\mathcal{D} \models \boldsymbol{q} \mid \max(\mathcal{D}) \leq N_{\mathit{tdp}(\boldsymbol{q})}\},$$
$$E^- = \{\mathcal{D} \not\models \boldsymbol{q} \mid \max(\mathcal{D}) \leq N_{\mathit{tdp}(\boldsymbol{q})}\}.$$

Indeed, let $\boldsymbol{q}' \in \mathcal{Q}^\sigma_-(\mathsf{U})$ fit E. Then $\mathit{tdp}(\boldsymbol{q}') = \mathit{tdp}(\boldsymbol{q})$ by (3), and so $\boldsymbol{q} \equiv \boldsymbol{q}'$ by Lemma 3. ❑

5 Characterisability in $\mathcal{Q}_p[\bigcirc, \Diamond_r]$

In this section, we prove a criterion of (polynomial) unique characterisability of queries within $\mathcal{Q}_p[\bigcirc, \Diamond_r]$. The criterion is applicable to $\mathcal{Q}_p[\bigcirc, \Diamond, \Diamond_r]$-queries in a normal form, which is defined and illustrated below.

Example 5. It is readily checked that the $\mathcal{Q}_p[\bigcirc, \Diamond_r]$-query $\boldsymbol{q} = \bigcirc\Diamond_r\bigcirc\Diamond_r(A \wedge B \wedge C \wedge \Diamond_r(B \wedge \Diamond_r(B \wedge C)))$ is equivalent to the $\mathcal{Q}_p[\bigcirc, \Diamond, \Diamond_r]$-query $\boldsymbol{q}^{nf} = \Diamond\Diamond(A \wedge B \wedge C)$.

We define the normal form for $\mathcal{Q}_p[\bigcirc, \Diamond, \Diamond_r]$-queries represented as a first-order CQ by a list of atoms. For example, the query $\boldsymbol{q}^{nf}$ above is given by the CQ

$$\boldsymbol{q}^{nf}(t_0) = t_0 < t_1, t_1 < t_2, A(t_2), B(t_2), C(t_2)$$

with one free (answer) variable t_0 and existentially quantified t_1 and t_2. In general, any $\boldsymbol{q} \in \mathcal{Q}_p[\bigcirc, \Diamond, \Diamond_r]$ is represented as a CQ

$$\rho_0(t_0), R_1(t_0, t_1), \dots, \rho_{n-1}(t_{n-1}), R_n(t_{n-1}, t_n), \rho_n(t_n),$$

where ρ_i is a set of atoms, $\rho_i(t_i) = \{A(t_i) \mid A \in \rho_i\}$ and $R_i \in \{\mathit{suc}, <, \leq\}$. We divide $\boldsymbol{q}$ into *blocks* $\boldsymbol{q}_i$ such that

$$\boldsymbol{q} = \boldsymbol{q}_0 \mathcal{R}_1 \boldsymbol{q}_1 \dots \mathcal{R}_n \boldsymbol{q}_n \qquad (4)$$

with $\mathcal{R}_i = R^i_1(t^i_0, t^i_1) \dots R^i_{n_i}(t^i_{n_i-1}, t^i_{n_i})$, for $R^i_j \in \{<, \leq\}$,

$$\boldsymbol{q}_i = \rho^i_0(s^i_0)\mathit{suc}(s^i_0, s^i_1)\rho^i_1(s^i_1) \dots \mathit{suc}(s^i_{k_i-1}, s^i_{k_i})\rho^i_{k_i}(s^i_{k_i})$$

and $s^i_{k_i} = t^{i+1}_0$, $t^i_{n_i} = s^i_0$. If $k_i = 0$, the block $\boldsymbol{q}_i$ is *primitive*. A primitive block $\boldsymbol{q}_i = \rho^i_0(s^i_0)$ with $i > 0$ and $|\rho^i_0| \geq 2$ is called a *lone conjunct* of $\boldsymbol{q}$.

Example 6. The query $\Diamond_r(A \wedge B)$ in Example $1(i)$, whose CQ representation is $t_0 \leq t_1, \rho_1(t_1)$, for $\rho_1 = \{A, B\}$, has a lone conjunct $\rho_1(t_1)$. In $\Diamond_r(A \wedge \bigcirc(A \wedge B))$ from Example $1(ii)$, represented as $t_0 \leq t_1$, $A(t_1), \mathit{suc}(t_1, t_2), \rho_1(t_2)$, the conjunct $\rho_1(t_2)$ is not lone.

Now, we say that $\boldsymbol{q}$ given by (4) is in *normal form* if the following conditions hold:

(n1) $\rho^i_0 \neq \emptyset$ if $i > 0$, and $\rho^i_{k_i} \neq \emptyset$ if $i > 0$ or $k_i > 0$ (thus, of all the first/last ρ in a block only ρ^0_0 can be empty);

(n2) each $\mathcal{R}_i$ is either a single $t^i_0 \leq t^i_1$ or a sequence of $<$;

(n3) $\rho^i_{k_i} \not\supseteq \rho^{i+1}_0$ if $\boldsymbol{q}_{i+1}$ is primitive and R_{i+1} is $\leq$;

(n4) $\rho^i_{k_i} \not\subseteq \rho^{i+1}_0$ if $i > 0$, $\boldsymbol{q}_i$ is primitive and R_{i+1} is $\leq$.

The queries in Example 6 are in normal form with two blocks each; the query $\boldsymbol{q}^{nf}$ above is in normal form with two blocks $\boldsymbol{q}_0 = \top(t_0)$ and $\boldsymbol{q}_1 = A(t_2) \wedge B(t_2) \wedge C(t_2)$.

Lemma 7. *Every query in $\mathcal{Q}_p[\bigcirc, \Diamond_r]$ is equivalent to a query in normal form that can be computed in linear time.*

A query $\boldsymbol{q} \in \mathcal{Q}_p[\bigcirc, \Diamond]$ is *safe* if it is equivalent to a query $\boldsymbol{q}' \in \mathcal{Q}_p[\bigcirc, \Diamond]$ in normal form not containing lone conjuncts. We are now in the position to formulate the criterion.

Theorem 8. *(i) A query $\boldsymbol{q} \in \mathcal{Q}_p[\bigcirc, \Diamond_r]$ is uniquely characterisable within $\mathcal{Q}_p[\bigcirc, \Diamond_r]$ iff $\boldsymbol{q}$ is safe.*

(ii) Those queries that are uniquely characterisable within $\mathcal{Q}_p[\bigcirc, \Diamond_r]$ are actually polynomially characterisable within $\mathcal{Q}_p[\bigcirc, \Diamond_r]$.

(iii) The class $\mathcal{Q}_p[\bigcirc, \Diamond_r]$ is polynomially characterisable for bounded query size.

(iv) The class $\mathcal{Q}_p[\bigcirc, \Diamond]$ is polynomially characterisable.

Proof. A detailed proof is given in the full version. Here, we define a polynomial-size example set $E = (E^+, E^-)$ characterising a query $\boldsymbol{q}$ in normal form (4), which does not contain lone conjuncts. Let b be the number of $\bigcirc$ and $\Diamond$ in $\boldsymbol{q}$ plus 1. For each block $\boldsymbol{q}_i$ in (4), we take two words

$$\bar{\boldsymbol{q}}_i = \rho^i_0 \dots \rho^i_{k_i}, \quad \bar{\boldsymbol{q}}_i \bowtie \bar{\boldsymbol{q}}_{i+1} = \rho^i_0 \dots (\rho^i_{k_i} \cup \rho^{i+1}_0) \dots \rho^{i+1}_{k_{i+1}}.$$

The set E^+ contains the data instances given by the words

- $\mathcal{D}_b = \bar{\boldsymbol{q}}_0 \emptyset^b \dots \bar{\boldsymbol{q}}_i \emptyset^b \bar{\boldsymbol{q}}_{i+1} \dots \emptyset^b \bar{\boldsymbol{q}}_n$,
- $\mathcal{D}_i = \bar{\boldsymbol{q}}_0 \emptyset^b \dots \bar{\boldsymbol{q}}_i \bowtie \bar{\boldsymbol{q}}_{i+1} \dots \emptyset^b \bar{\boldsymbol{q}}_n$ if $\mathcal{R}_{i+1}$ is $\leq$,
- $\mathcal{D}_i = \bar{\boldsymbol{q}}_0 \emptyset^b \dots \bar{\boldsymbol{q}}_i \emptyset^{n_{i+1}} \bar{\boldsymbol{q}}_{i+1} \dots \emptyset^b \bar{\boldsymbol{q}}_n$ otherwise.

Here, $\emptyset^b$ is a sequence of b-many $\emptyset$ and similarly for $\emptyset^{n_{i+1}}$. The set E^- contains all data instances of the form

- $\mathcal{D}^-_i = \bar{\boldsymbol{q}}_0 \emptyset^b \dots \bar{\boldsymbol{q}}_i \emptyset^{n_{i+1}-1} \bar{\boldsymbol{q}}_{i+1} \dots \emptyset^b \bar{\boldsymbol{q}}_n$ if $n_{i+1} > 1$;
- $\mathcal{D}^-_i = \bar{\boldsymbol{q}}_0 \emptyset^b \dots \bar{\boldsymbol{q}}_i \bowtie \bar{\boldsymbol{q}}_{i+1} \dots \emptyset^b \bar{\boldsymbol{q}}_n$ if $\mathcal{R}_{i+1}$ is a single $<$,

and also the data instances obtained from $\mathcal{D}_b$ by

(a) removing a single atom from some $\rho^i_j \neq \emptyset$ or removing the whole $\rho^i_j = \emptyset$, for $i \neq 0$ and $j \neq 0$, from some $\bar{\boldsymbol{q}}_i$;

(b) replacing $\bar{\boldsymbol{q}}_i = \rho^i_0 \dots \rho^i_l \rho^i_{l+1} \dots \rho^i_{k_i}$ $(k_i > 0)$ by $\bar{\boldsymbol{q}}'_i \emptyset^b \bar{\boldsymbol{q}}''_i$, where $\bar{\boldsymbol{q}}'_i = \rho^i_0 \dots \rho^i_l$, $\bar{\boldsymbol{q}}''_i = \rho^i_{l+1} \dots \rho^i_{k_i}$ and $l \geq 0$;

(c) replacing some $\rho^i_l \neq \emptyset$, $0 < l < k_i$, by $\rho^i_l \emptyset^b \rho^i_l$;

(d) replacing $\rho^i_{k_i}$ $(k_i > 0, |\rho^i_{k_i}| \geq 2)$ with $\rho^i_{k_i} \setminus \{A\} \emptyset^b \rho^i_{k_i}$, for some $A \in \rho^i_{k_i}$, or replacing ρ^i_0 $(k_i > 0, |\rho^i_0| \geq 2)$ with $\rho^i_0 \emptyset^b \rho^i_0 \setminus \{A\}$, for some $A \in \rho^i_0$;

(e) replacing $\rho^0_0 \neq \emptyset$ with $\rho^0_0 \setminus \{A\} \emptyset^b \rho^0_0$, for some $A \in \rho^0_0$, if $k_0 = 0$, and with $\rho^0_0 \emptyset^b \rho^0_0$ if $k_0 > 0$.

The size of E is clearly polynomial in $|\boldsymbol{q}|$. It is readily seen that $\mathcal{D} \models \boldsymbol{q}$ for all $\mathcal{D} \in E^+$. To continue the proof sketch, note that $\mathcal{D} \models \boldsymbol{q}$ iff there is a *homomorphism* h from the set $\mathit{var}(\boldsymbol{q})$ of variables in $\boldsymbol{q}$ to $[0, \max(\mathcal{D})]$, i.e., $h(t_0) = 0$, $A(h(t)) \in \mathcal{D}$ if $A(t) \in \boldsymbol{q}$, $h(t') = h(t) + 1$ if $\mathit{suc}(t, t') \in \boldsymbol{q}$, and $h(t)\, R\, h(t')$ if $R(t, t') \in \boldsymbol{q}$ for $R \in \{<, \leq\}$. Using the assumption that $\boldsymbol{q}$ is in normal form, one can show that there is no homomorphism witnessing $\mathcal{D} \models \boldsymbol{q}$, for any $\mathcal{D} \in E^-$.

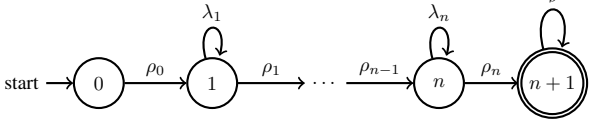
Suppose now that $\boldsymbol{q}' \in \mathcal{Q}_p[\bigcirc, \Diamond_r]$ in normal form is given and $\boldsymbol{q}' \not\equiv \boldsymbol{q}$. If $\mathcal{D}_b \not\models \boldsymbol{q}'$, we are done as $\mathcal{D}_b \in E^+$. Otherwise, let h be a homomorphism witnessing $\mathcal{D}_b \models \boldsymbol{q}'$. Then one can show that either the restriction of h to the blocks of $\boldsymbol{q}'$ is an isomorphism onto the blocks of $\boldsymbol{q}$ or there exists a data instance $\mathcal{D}$ obtained using one of the rules (a)–(e) such that a suitably modified h is a homomorphism from $\boldsymbol{q}'$ to $\mathcal{D}$. In the latter case, we are done as $\mathcal{D} \in E^-$ and $\mathcal{D} \models \boldsymbol{q}'$. In the former case, $\boldsymbol{q}$ and $\boldsymbol{q}'$ coincide with the exception of the sequences of $\Diamond$ and $\Diamond_r$ between blocks. Then $\boldsymbol{q}$ can be separated from $\boldsymbol{q}'$ using the examples $\mathcal{D}_i$ and $\mathcal{D}^-_i$. ❑

6 Characterisability in $\mathcal{Q}^\sigma_p[\mathsf{U}]$

LTL-queries with U do not correspond to CQs (because of the universal quantification in its semantics), and so require a different approach. We view them as defining regular languages. With each $\mathcal{Q}^\sigma_p[\mathsf{U}]$-query of the form (2) we associate the following regular expression over the alphabet 2^σ:

$$\boldsymbol{q} = \rho_0 \lambda^*_1 \rho_1 \lambda^*_2 \dots \lambda^*_n \rho_n \lambda^*_{n+1} \tag{5}$$

where $\lambda_{n+1} = \emptyset$ and $\bot^* = \varepsilon$. We regard the words of the language $\boldsymbol{L}(\boldsymbol{q})$ over 2^σ as data instances. Clearly, $\mathcal{D}' \models \boldsymbol{q}$ iff there is $\mathcal{D} \in \boldsymbol{L}(\boldsymbol{q})$ such that $\mathcal{D} \Subset \mathcal{D}'$, i.e., $\mathcal{D} = (\delta_0, \dots, \delta_k)$ and $\mathcal{D}' = (\delta'_0, \dots, \delta'_k)$, for some $k < \omega$, and $\delta_i \subseteq \delta'_i$, for all $i \leq k$. The language $\boldsymbol{L}_{\boldsymbol{q}}$ of all σ-data instances $\mathcal{D} \models \boldsymbol{q}$ (regarded as words over 2^σ) can be given by the NFA $\mathfrak{A}_{\boldsymbol{q}}$ below, where each $\to_\alpha$, for $\alpha \neq \bot$, stands for all transitions $\to_\beta$ with $\alpha \subseteq \beta \subseteq \sigma$ (note that $\bot \notin \sigma$):



Without loss of generality we assume that all our $\boldsymbol{q}$ are *minimal* in the sense that by replacing any $\lambda_i \neq \bot$ with $\bot$ in $\boldsymbol{q}$ we obtain a query that is *not equivalent* to $\boldsymbol{q}$. For example, in minimal $\boldsymbol{q}$, $\rho_j \supseteq \dots \supseteq \rho_i \supseteq \lambda_i$ and $\lambda_l = \bot$ for all $l \in (j, i)$ imply $\rho_j \not\subseteq \lambda_j$ as otherwise $\lambda_j \mathsf{U} (\rho_j \wedge (\bot \mathsf{U} \dots (\lambda_i \mathsf{U} \varphi) \dots))$ is equivalent to $\bot \mathsf{U} (\rho_j \wedge (\bot \mathsf{U} \dots (\lambda_i \mathsf{U} \varphi) \dots))$. Using standard automata-theoretic techniques, one can show:

Theorem 9. *Any $\mathcal{Q}^\sigma_p[\mathsf{U}]$-queries $\boldsymbol{q} \not\equiv \boldsymbol{q}'$ can be separated by some $\mathcal{D}$ with $\max(\mathcal{D}) \leq O((\min\{\mathit{tdp}(\boldsymbol{q}), \mathit{tdp}(\boldsymbol{q}')\})^2)$.*

Using Theorem 9 in the proof of Theorem 4 we obtain:

Corollary 10. *The class $\mathcal{Q}^\sigma_p[\mathsf{U}]$ is exponentially characterisable within $\mathcal{Q}^\sigma_p[\mathsf{U}]$.*

The following examples illustrate difficulties in finding short unique characterisations of $\mathcal{Q}^\sigma_p[\mathsf{U}]$-queries, namely, that in general, data instances of different shapes and forms are needed to separate $\mathcal{Q}^\sigma_p[\mathsf{U}]$-queries. To unclutter notation we omit $\{\}$ in singletons like $\{A\}$.

Example 11. (a) The shortest data instance separating

$$\boldsymbol{q} = X\emptyset^* A \bot^* B \bot^* A B^* A A^* B \emptyset^*,$$
$$\boldsymbol{q}' = X\emptyset^* A \bot^* B A^* A B^* A \bot^* B \emptyset^*$$

is $\mathcal{D} = XABABBAAB$ with $\mathcal{D} \models \boldsymbol{q}$ and $\mathcal{D} \not\models \boldsymbol{q}'$ (e.g., $XABABAAB$ satisfies both $\boldsymbol{q}$ and $\boldsymbol{q}'$).

(b) For $l > 0$, let $\boldsymbol{q}_l = (AB^*)^l A A^* B B^*$. Then

$$X A^* \boldsymbol{q}_{l_1} \boldsymbol{q}_{l_2} \dots \boldsymbol{q}_{l_k} X \emptyset^* \;\not\equiv\; X \bot^* \boldsymbol{q}_{l_1} \boldsymbol{q}_{l_2} \dots \boldsymbol{q}_{l_k} X \emptyset^*,$$
$$X A^* \boldsymbol{q}_{l_1} \boldsymbol{q}_{l_2} \dots \boldsymbol{q}_{l_k} A \emptyset^* \;\equiv\; X \bot^* \boldsymbol{q}_{l_1} \boldsymbol{q}_{l_2} \dots \boldsymbol{q}_{l_k} A \emptyset^*.$$

If $l_1 \leq \dots \leq l_k$, the former inequivalence is witnessed by the instance $X A^{l_1} B A^{l_2} B \dots A^{l_k} B A^{l_k} B X$. Less generally, $X A^* \boldsymbol{q}_1 \boldsymbol{q}_2 X \emptyset^* \not\equiv X \bot^* \boldsymbol{q}_1 \boldsymbol{q}_2 X \emptyset^*$ can be shown by $XAABAAABAAABX$ or by $XAABABAABABX$ (spot the difference and see $(\mathfrak{n}_2)$ below).

Here, we consider the class $\mathcal{P}^\sigma[\mathsf{U}]$ of *peerless* queries given by (5), in which, for any i, either $\lambda_i = \bot$ or the sets λ_i and ρ_i are *incomparable* with respect to $\subseteq$. Our main result is that $\mathcal{P}^\sigma[\mathsf{U}]$ is polynomially characterisable within $\mathcal{Q}^\sigma_p[\mathsf{U}]$.

We start with a general observation. Consider two queries $\boldsymbol{q} = \rho_0 \lambda^*_1 \dots \lambda^*_n \rho_n \emptyset^*$ and $\boldsymbol{q}' = \rho_0 \mu^*_1 \dots \mu^*_n \rho_n \emptyset^*$. We say that $\lambda_i \neq \bot$ *subsumes* $\mu_j \neq \bot$ if either $i = j$ and $\mu_j \subseteq \lambda_i$, or $j < i$ and $\mu_j \rho_j \dots \rho_{i-1} \Subset \rho_j \dots \rho_{i-1} \lambda_i$, or $j > i$ and $\rho_i \dots \rho_{j-1} \mu_j \Subset \lambda_i \rho_i \dots \rho_{j-1}$. In the last two cases,

$$\mu_j \subseteq \rho_j \subseteq \dots \subseteq \rho_{i-1} \subseteq \lambda_i, \;\; \mu_j \subseteq \rho_{j-1} \subseteq \dots \subseteq \rho_i \subseteq \lambda_i,$$

respectively. Note that, for peerless $\boldsymbol{q}$, the last inclusion is impossible. If λ_i and μ_j subsume each other, in which case $\lambda_i = \mu_j$, we call (λ_i, μ_j) a *matching pair*. Observe also that, for $\mathcal{D}^i_{\boldsymbol{q}} = \rho_0 \dots \rho_{i-1} \lambda_i \rho_i \dots \rho_n$, if $\mathcal{D}^i_{\boldsymbol{q}} \models \boldsymbol{q}'$, then λ_i subsumes some μ_j: $\rho_0 \dots \rho_n \emptyset \Subset \mathcal{D}^i_{\boldsymbol{q}}$ means that λ_i subsumes $\mu_{n+1} = \emptyset$, and $\rho_0 \dots \mu_j \dots \rho_n \Subset \mathcal{D}^i_{\boldsymbol{q}}$ that λ_i subsumes μ_j. The proof of the next lemma is given in the full version:

Lemma 12. *For any queries $\boldsymbol{q}$ and $\boldsymbol{q}'$ as above, either (i) each $\lambda_i \neq \bot$ subsumes μ_j occurring in some matching pair (λ_k, μ_j) or (ii) $\boldsymbol{q}$ and $\boldsymbol{q}'$ are separated by a data instance of the form $\mathcal{D}^i_{\boldsymbol{q}}$ or $\mathcal{D}^j_{\boldsymbol{q}'}$. Also, if $\boldsymbol{q}$ is peerless, λ_i can only subsume μ_j in the matching pair (λ_i, μ_j) with $i \geq j$, in which case $\mu_j = \rho_j = \dots = \rho_{i-1} = \lambda_i$.*

Note that the number of data instances of the form $\mathcal{D}^i_{\boldsymbol{q}'}$ for all possible $\mathcal{Q}^\sigma_p[\mathsf{U}]$-queries $\boldsymbol{q}'$ can be exponential in $|\sigma|$. The following example indicates how to overcome this issue.

Example 13. Let $\sigma = \{A, B, C, D, X\}$. To separate the query $X\{C, D\}^* A\emptyset^*$ from any $X\lambda^* A\emptyset^*$ with $A, D \notin \lambda$, we can use $\mathcal{D} = X\sigma \setminus \{A, D\}A$.

Theorem 14. *The class $\mathcal{P}^\sigma[\mathsf{U}]$ is polynomially characterisable within $\mathcal{Q}_p^\sigma[\mathsf{U}]$.*

Proof. We show that any $\boldsymbol{q} = \rho_0\lambda_1^*\rho_1\lambda_2^* \dots \lambda_n^*\rho_n\emptyset^*$ in $\mathcal{P}^\sigma[\mathsf{U}]$ is characterised by the example set $E = (E^+, E^-)$, where E^+ contains all data instances of the following forms:

- $(\mathfrak{p}_0)$ $\rho_0 \dots \rho_n$,
- $(\mathfrak{p}_1)$ $\rho_0 \dots \rho_{i-1}\lambda_i\rho_i \dots \rho_n = \mathcal{D}^i_{\boldsymbol{q}}$,
- $(\mathfrak{p}_2)$ $\rho_0 \dots \rho_{i-1}\lambda_i^k\rho_i \dots \rho_{j-1}\lambda_j\rho_j \dots \rho_n = \mathcal{D}^j_{i,k}$, for $i < j$ and $k = 1, 2$;

and E^- has all instances that are *not* in $\boldsymbol{L}(\boldsymbol{q})$ of the forms:

- $(\mathfrak{n}_0)$ σ^n and $\sigma^{n-i}\sigma \setminus \{A\}\sigma^i$, for $A \in \rho_i$,
- $(\mathfrak{n}_1)$ $\rho_0 \dots \rho_{i-1}\sigma \setminus \{A, B\}\rho_i \dots \rho_n$, for $A \in \lambda_i \cup \{\bot\}$ and $B \in \rho_i \cup \{\bot\}$,
- $(\mathfrak{n}_2)$ for all i and $A \in \lambda_i \cup \{\bot\}$, *some* data instance

$$\mathcal{D}^i_A = \rho_0 \dots \rho_{i-1}(\sigma \setminus \{A\})\rho_i\lambda_{i+1}^{k_{i+1}} \dots \lambda_n^{k_n}\rho_n, \qquad (6)$$

if any, such that $\max(\mathcal{D}^i_A) \leq (n+1)^2$ and $\mathcal{D}^i_A \not\models \boldsymbol{q}^\dagger$ for $\boldsymbol{q}^\dagger$ obtained from $\boldsymbol{q}$ by replacing λ_j, for all $j \leq i$, with $\bot$. Note that $\mathcal{D}^i_A \not\models \boldsymbol{q}$ for peerless $\boldsymbol{q}$.

By definition, $\boldsymbol{q}$ fits E and $|E|$ is polynomial in $|\boldsymbol{q}|$. We prove in the full version that E uniquely characterises $\boldsymbol{q}$. ❑

One reason why this construction does not generalise to the whole $\mathcal{Q}_p^\sigma[\mathsf{U}]$ is that $\mathcal{D}^i_A \not\models \boldsymbol{q}^\dagger$ does not imply $\mathcal{D}^i_A \not\models \boldsymbol{q}$ for non-peerless $\boldsymbol{q}$, as shown by the following example:

Example 15. Let $\boldsymbol{q} = XA^*AB^*A\bot^*AB^*AA^*BB^*X\emptyset^*$. For any data instance $\mathcal{D}^3_\bot$ satisfying (6)—for example, $\mathcal{D}^3_\bot = XAA\sigma ABABX$—we have $\mathcal{D}^3_\bot \models \boldsymbol{q}$.

7 Characterisability in $\mathcal{Q}[\Diamond]$

In the previous two sections, we have investigated characterisability of path-shaped queries. Here, we first justify that restriction by exhibiting two examples that show how temporal branching can destroy polynomial characterisability in $\mathcal{Q}[\Diamond]$. Both examples make use of *unbalanced* queries, in which different branches have different length. We then show that this is no accident: one can at least partially restore polynomial characterisability for classes without unbalanced queries.

We start by observing that, without loss of generality, it is enough to consider conjunctions of path queries only:

Lemma 16. *For every $\boldsymbol{q} \in \mathcal{Q}[\Diamond]$, one can compute in polynomial time an equivalent query of the form $\boldsymbol{q}_1 \wedge \dots \wedge \boldsymbol{q}_n$ with $\boldsymbol{q}_i \in \mathcal{Q}_p[\Diamond]$, for $1 \leq i \leq n$.*

The first example showing non-polynomial characterisability is rather straightforward but requires unbounded branching and an unbounded number of atoms. We write queries $\boldsymbol{q} \in \mathcal{Q}_p^\sigma[\Diamond]$ of the form

$$\boldsymbol{q} = \rho_0 \wedge \Diamond(\rho_1 \wedge \Diamond(\rho_2 \wedge \dots \wedge \Diamond\rho_n)) \qquad (7)$$

as words $\rho_0\rho_1 \dots \rho_n$ over 2^σ (omitting but not forgetting $\lambda_i^* = \emptyset^*$ from (5)) and also use $\rho_0\rho_1 \dots \rho_n$ to denote the data instance defined by $\boldsymbol{q}$.

Example 17. Consider $\boldsymbol{q}_n = \boldsymbol{s}_1 \wedge \dots \wedge \boldsymbol{s}_n$, where $n \geq 2$ and each $\boldsymbol{s}_i$ is a word repeating n times the sequence $A_1 \dots A_n$ (of singletons) with omitted A_i. Now, consider the queries $\boldsymbol{q}_n^{\boldsymbol{p}} = \boldsymbol{q}_n \wedge \boldsymbol{p}$, where $\boldsymbol{p} = \Diamond(A_{i_1} \wedge \Diamond(A_{i_2} \wedge \dots \wedge \Diamond A_{i_n}))$ and $A_{i_1} \dots A_{i_n}$ is a permutation of $A_1 \dots A_n$. Then $\boldsymbol{q}_n^{\boldsymbol{p}} \models \boldsymbol{q}_n$ and $\boldsymbol{q}_n \not\models \boldsymbol{q}_n^{\boldsymbol{p}}$ as shown by the data instance $\boldsymbol{s}_{i_1}\boldsymbol{s}_{i_2} \dots \boldsymbol{s}_{i_n}$. Moreover, if $\mathcal{D} \models \boldsymbol{q}_n$, $\mathcal{D} \not\models \boldsymbol{p}$ and $\boldsymbol{p}' \neq \boldsymbol{p}$, then $\mathcal{D} \models \boldsymbol{p}'$. It follows that, in any $E = (E^+, E^-)$ uniquely characterising $\boldsymbol{q}_n$, the set E^+ contains at least $n!$ data instances.

The class $\mathcal{Q}_{\leq n}[\Diamond]$ of queries of *branching factor* at most n contains all queries in $\mathcal{Q}[\Diamond]$ that are equivalent to a query of the form $\boldsymbol{q}_1 \wedge \dots \wedge \boldsymbol{q}_m$ with $m \leq n$ and $\boldsymbol{q}_i \in \mathcal{Q}_p[\Diamond]$. We next provide an example of non-polynomial characterisability that requires four atoms and bounded branching only.

Example 18. Let $\sigma = \{A_1, A_2, B_1, B_2\}$, $\boldsymbol{q}_1 = \emptyset(\boldsymbol{s}\sigma)^n\boldsymbol{s}$, and $\boldsymbol{q}_2 = \emptyset\sigma^{2n+1}$, where $\boldsymbol{s} = \{A_1, A_2\}\{B_1, B_2\}$. Consider the set P of 2^{n+1}-many queries of the form $\emptyset\boldsymbol{s}_1 \dots \boldsymbol{s}_{n+1}$ with $\boldsymbol{s}_i$ either $\{A_1\}\{A_2\}$ or $\{B_1\}\{B_2\}$. Then $\boldsymbol{q}_1 \wedge \boldsymbol{q}_2 \not\models \boldsymbol{q}$ for any $\boldsymbol{q} \in P$ and, for any $\mathcal{D}$ with $\mathcal{D} \models \boldsymbol{q}_1 \wedge \boldsymbol{q}_2$, there is at most one $\boldsymbol{q} \in P$ with $\mathcal{D} \not\models \boldsymbol{q}$ (the proof is rather involved). It follows that $\boldsymbol{q}_1 \wedge \boldsymbol{q}_2 \not\models \boldsymbol{q}_1 \wedge \boldsymbol{q}_2 \wedge \boldsymbol{q}$ for all $\boldsymbol{q} \in P$, but 2^{n+1} positive examples are needed to separate $\boldsymbol{q}_1 \wedge \boldsymbol{q}_2$ from all $\boldsymbol{q}_1 \wedge \boldsymbol{q}_2 \wedge \boldsymbol{q}$ with $\boldsymbol{q} \in P$.

We next identify polynomially characterisable classes of $\mathcal{Q}[\Diamond]$-queries, assuming as before that $\rho_n \neq \emptyset$ in any $\boldsymbol{q}$ of the form (1). We call a query $\boldsymbol{q}_1 \wedge \dots \wedge \boldsymbol{q}_n \in \mathcal{Q}[\Diamond]$ with $\boldsymbol{q}_1, \dots, \boldsymbol{q}_n \in \mathcal{Q}_p[\Diamond]$ *balanced* if all $\boldsymbol{q}_i$ have the same depth; further, we call it *simple* if, in each $\boldsymbol{q}_i$ given by (1), $|\rho_j| = 1$ for all j. Let $\mathcal{Q}_b[\Diamond]$ denote the class of queries in $\mathcal{Q}[\Diamond]$ that are equivalent to a balanced query.

Theorem 19. (i) *The class of simple queries in $\mathcal{Q}_b[\Diamond]$ is polynomially characterisable within $\mathcal{Q}_b[\Diamond]$.*

(ii) *For any n, the class $\mathcal{Q}_b[\Diamond] \cap \mathcal{Q}_{\leq n}[\Diamond]$ is polynomially characterisable.*

Proof. Let $\boldsymbol{q} \in \mathcal{Q}_p^\sigma[\Diamond]$. We start with a lemma on the existence of polynomial-size σ-data instances $\mathcal{D}_{\boldsymbol{q},k}$ such that $\mathcal{D}_{\boldsymbol{q},k} \not\models \boldsymbol{q}$ and $\mathcal{D}_{\boldsymbol{q},k} \models \boldsymbol{q}'$ for all $\boldsymbol{q}' \in \mathcal{Q}_p^\sigma[\Diamond]$ with $\boldsymbol{q}' \not\models \boldsymbol{q}$ and $tdp(\boldsymbol{q}') \leq k$. Note that such $\mathcal{D}_{\boldsymbol{q},k}$ do not exist in general.

Example 20. Let $\boldsymbol{q} = A \wedge B$. Then $A \not\models \boldsymbol{q}$ and $B \not\models \boldsymbol{q}$ but there does not exist any $\mathcal{D}_{\boldsymbol{q},0}$ such that $\mathcal{D}_{\boldsymbol{q},0} \not\models \boldsymbol{q}$ and $\mathcal{D}_{\boldsymbol{q},0} \models A$ and $\mathcal{D}_{\boldsymbol{q},0} \models B$.

In the following lemma we therefore assume that $\boldsymbol{q}$ does not speak about the initial timepoint.

Lemma 21. *Let $\boldsymbol{q} \in \mathcal{Q}_p^\sigma[\Diamond]$ be of the form $\Diamond\boldsymbol{q}'$ and let $k > 0$. Then one can construct in polynomial time a σ-data instance $\mathcal{D}_{\boldsymbol{q},k}$ such that $\mathcal{D}_{\boldsymbol{q},k} \not\models \boldsymbol{q}$ and $\mathcal{D}_{\boldsymbol{q},k} \models \boldsymbol{q}'$ for all $\boldsymbol{q}' \in \mathcal{Q}_p^\sigma[\Diamond]$ with $\boldsymbol{q}' \not\models \boldsymbol{q}$ and $tdp(\boldsymbol{q}') \leq k$.*

Proof. Assuming that $\boldsymbol{q} = \Diamond(\rho_1 \wedge \Diamond(\rho_2 \wedge \dots \wedge \Diamond\rho_n))$ with $\rho_i = \{A_1^i, \dots, A_{n_i}^i\}$ for $i \geq 1$, we set

$$\mathcal{D}_{\boldsymbol{q},k} = \sigma\boldsymbol{s}_1^k\sigma \dots \boldsymbol{s}_{n-1}^k\sigma\boldsymbol{s}_n^k,$$

where $\boldsymbol{s}_i = \sigma \setminus \{A_1^i\} \dots \sigma \setminus \{A_{n_i}^i\}$. One can show by induction that $\mathcal{D}_{\boldsymbol{q},k}$ is as required. ❑

Using Lemma 21, for any $\boldsymbol{q} \in \mathcal{Q}^\sigma[\Diamond]$, one can construct a polynomial-size set of negative examples as follows. Suppose $\boldsymbol{q} = \boldsymbol{q}_1 \wedge \dots \wedge \boldsymbol{q}_n \in \mathcal{Q}^\sigma[\Diamond]$ with

$$\boldsymbol{q}_i = \rho_0^i \wedge \Diamond(\rho_1^i \wedge \Diamond(\rho_2^i \wedge \dots \wedge \Diamond\rho_{n_i^i})).$$

Let $\rho = \bigwedge_{i=1}^n \rho_0^i$ and let $\boldsymbol{q}_i^-$ be $\boldsymbol{q}_i$ without the conjunct ρ_0^i, so Lemma 21 is applicable to $\boldsymbol{q}_i^-$. Now let $E^-_{\boldsymbol{q},m}$ contain the σ-data instances $\mathcal{D}_{\boldsymbol{q}_i^-,m}$ and $\sigma \setminus \{A\}\sigma^m$ for all $A \in \rho$.

Lemma 22. *(i) For any $\mathcal{D} \in E^-_{\boldsymbol{q},m}$, we have $\mathcal{D} \not\models \boldsymbol{q}$.*

(ii) For any $\boldsymbol{q}' \in \mathcal{Q}^\sigma[\Diamond]$ with $\boldsymbol{q}' \not\models \boldsymbol{q}$ and $tdp(\boldsymbol{q}') \leq m$, there exists $\mathcal{D} \in E^-_{\boldsymbol{q},m}$ with $\mathcal{D} \models \boldsymbol{q}'$.

It follows from Lemma 22 that non-polynomial characterisability of $\mathcal{Q}[\Diamond]$-queries can only be caused by the need for super-polynomially-many positive examples. We now discuss the construction of positive examples in the proof of Theorem 19 (ii); part (i) is dealt with in the full version. Let $\boldsymbol{q} = \boldsymbol{q}_1 \wedge \dots \wedge \boldsymbol{q}_m \in \mathcal{Q}_b^\sigma[\Diamond] \cap \mathcal{Q}^\sigma_{\leq n}[\Diamond]$ with $m \leq n$ and

$$\boldsymbol{q}_i = \rho_0^i \wedge \Diamond(\rho_1^i \wedge \Diamond(\rho_2^i \wedge \dots \wedge \Diamond\rho_N^i)).$$

For any map $f\colon \{1,\dots,m\} \to \{1,\dots,N\}$, construct a σ-data instance $\mathcal{D}_f$ by inserting $\rho^i_{f(i)}$ into the data instance σ^N in position $f(i)$. Let E^+ contain the data instance $\rho\sigma^N$ for $\rho = \bigcup_{i=1}^m \rho_0^i$ and all the data instances $\mathcal{D}_f$. One can show that (E^+, E^-) characterises $\boldsymbol{q}$ in $\mathcal{Q}_b[\Diamond] \cap \mathcal{Q}_{\leq n}[\Diamond]$. ❑

8 2D Temporal Instance Queries

Now we consider 'two-dimensional' query languages that combine instance queries (over the object domain) in the standard description logics $\mathcal{EL}$ and $\mathcal{ELI}$ (Baader et al. 2017) with the *LTL*-queries (over the temporal domain) considered above. Our aim is to understand how far the characterisability results of the previous sections can be generalised to the 2D case. A *relational signature* is a finite set $\Sigma \neq \emptyset$ of unary and binary predicate symbols. A *Σ-data instance* $\mathcal{A}$ is a finite set of *atoms* $A(a)$ and $P(a,b)$ with $A, P \in \Sigma$ and *individual names* a, b. Let $ind(\mathcal{A})$ be the set of individual names in $\mathcal{A}$. We assume that $P^-(a,b) \in \mathcal{A}$ iff $P(b,a) \in \mathcal{A}$, calling P^- the *inverse* of P (with $P^{--} = P$). Let $S := P \mid P^-$. *Temporal instance queries* are defined by the grammar

$$\boldsymbol{q} \;:=\; \top \mid \bot \mid A \mid \exists S.\boldsymbol{q} \mid \boldsymbol{q}_1 \wedge \boldsymbol{q}_2 \mid \boldsymbol{op}\,\boldsymbol{q} \mid \boldsymbol{q}_1 \mathsf{U}\, \boldsymbol{q}_2,$$

where $\boldsymbol{op} \in \{\bigcirc, \Diamond, \Diamond_r\}$. Such queries without temporal operators are called *$\mathcal{ELI}$-queries*; those of them without inverses P^- are *$\mathcal{EL}$-queries*. A *temporal Σ-data instance* $\mathcal{D}$ is a finite sequence $\mathcal{A}_0, \dots, \mathcal{A}_n$ of Σ-data instances. We set $ind(\mathcal{D}) = \bigcup_{i=1}^n ind(\mathcal{A}_i)$. For any $\ell \in \mathbb{N}$ and $a \in ind(\mathcal{D})$, the *truth-relation* $\mathcal{D}, a, \ell \models \boldsymbol{q}$ is defined by induction:

$\mathcal{D}, a, \ell \models A$ iff $A(a) \in \mathcal{A}_i$,

$\mathcal{D}, a, \ell \models \exists S.\boldsymbol{q}$ iff there is $b \in ind(\mathcal{A}_i)$ such that $S(a,b) \in \mathcal{A}_i$ and $\mathcal{D}, b, \ell \models \boldsymbol{q}$,

with the remaining clauses being obvious generalisations of the *LTL* ones. An *example set* is a pair $E = (E^+, E^-)$ with finite sets E^+ and E^- of pointed temporal data instances $\mathcal{D}, a$ such that $a \in ind(\mathcal{D})$. We say that $\boldsymbol{q}$ *fits* E if $\mathcal{D}^+, a^+, 0 \models \boldsymbol{q}$ and $\mathcal{D}^-, a^-, 0 \not\models \boldsymbol{q}$, for all $\mathcal{D}^+, a^+ \in E^+$ and $\mathcal{D}^-, a^- \in E^-$. As before, E *uniquely characterises* $\boldsymbol{q}$ if $\boldsymbol{q}$ fits it and every $\boldsymbol{q}'$ fitting E is logically equivalent to $\boldsymbol{q}$.

We need the following result on the unique characterisability of $\mathcal{ELI}$-queries.

Theorem 23 (ten Cate and Dalmau 2021)**.** *The class of $\mathcal{ELI}$-queries is polynomially characterisable.*

Theorem 24 is shown by constructing frontiers in the set of $\mathcal{ELI}$-queries partially ordered by entailment, where a set $\mathcal{F}$ of $\mathcal{ELI}$-queries is called a *frontier* of an $\mathcal{ELI}$-query $\boldsymbol{q}$ if the following hold:

- $\boldsymbol{q} \models \boldsymbol{q}'$ and $\boldsymbol{q}' \not\models \boldsymbol{q}$, for all $\boldsymbol{q}' \in \mathcal{F}$;
- if $\boldsymbol{q} \models \boldsymbol{q}''$ for some $\mathcal{ELI}$-query $\boldsymbol{q}''$, then $\boldsymbol{q}'' \models \boldsymbol{q}$ or there exists $\boldsymbol{q}' \in \mathcal{F}$ with $\boldsymbol{q}' \models \boldsymbol{q}''$.

Theorem 24 (ten Cate and Dalmau 2021)**.** *A frontier $\mathcal{F}(\boldsymbol{q})$ of any $\mathcal{ELI}$-query $\boldsymbol{q}$ can be computed in polynomial time.*

Theorem 23 follows from Theorem 24. For any $\mathcal{ELI}$-query $\boldsymbol{q}$ we denote by $\hat{\boldsymbol{q}}$ the tree-shaped data instance defined by $\boldsymbol{q}$ with designated root a. Then $\boldsymbol{q}$ is characterised by E with $E^+ = \{\hat{\boldsymbol{q}}\}$ and $E^- = \{\hat{\boldsymbol{r}} \mid \boldsymbol{r} \in \mathcal{F}(\boldsymbol{q})\}$.

For any unrestricted temporal query language $\mathcal{Q}[\Phi]$ and $\mathcal{L} \in \{\mathcal{EL}, \mathcal{ELI}\}$, we denote by $\mathcal{Q}[\Phi] \otimes \mathcal{L}$ the set of all temporal instance queries with operators in Φ with (for $\mathcal{ELI}$) or without (for $\mathcal{EL}$) inverse predicates. We generalise the path-shaped queries $\mathcal{Q}_p[\Phi]$ as follows: $\mathcal{Q}_p[\Phi] \otimes \mathcal{L}$ denotes the class of queries $\boldsymbol{q}$ in $\mathcal{Q}[\Phi] \otimes \mathcal{L}$ such that, for any subquery $\boldsymbol{q}_1 \wedge \boldsymbol{q}_2$ of $\boldsymbol{q}$, either $\boldsymbol{q}_1$ or $\boldsymbol{q}_2$ do not have an occurrence of any operator in Φ that is not in the scope of $\exists S$. To illustrate, $\exists S.\Diamond A \wedge \Diamond\exists S.A$ is in $\mathcal{Q}_p[\Phi] \otimes \mathcal{L}$, but $\Diamond A \wedge \Diamond\exists S.A$ is not. We make two observations about unique characterisability in these 'full' combinations.

Theorem 25. *(i) $\mathcal{Q}[\bigcirc, \Diamond] \otimes \mathcal{EL}$ is uniquely characterisable.*

(ii) $\mathcal{Q}_p[\bigcirc] \otimes \mathcal{ELI}$ and $\mathcal{Q}_p[\Diamond] \otimes \mathcal{ELI}$ are polynomially characterisable.

Here, (i) is shown similarly to Theorem 4 (it remains open whether it can be extended to $\mathcal{Q}[\bigcirc, \Diamond] \otimes \mathcal{ELI}$); (ii) is proved by generalising Theorem 24 to temporal data instances.

We now show that the application of the DL constructor $\exists P$ to temporal queries with both $\bigcirc$ and $\Diamond$ destroys polynomial characterisability. Denote by $\mathcal{EL}(\mathcal{Q}_p[\bigcirc, \Diamond])$ the class of queries in $\mathcal{Q}_p[\bigcirc, \Diamond] \otimes \mathcal{EL}$ that contain no $\exists P$ in the scope of a temporal operator.

Theorem 26. *$\mathcal{EL}(\mathcal{Q}_p[\bigcirc, \Diamond])$ is not polynomially characterisable.*

Proof. Consider the queries $\boldsymbol{q}_n = \exists P.\boldsymbol{q}_1^n \wedge \dots \wedge \exists P.\boldsymbol{q}_n^n$, in which each $\boldsymbol{q}_i^n$ corresponds to the regular expression

$$\underbrace{BB\emptyset^*A}_{1} \dots \underbrace{BB\emptyset^*A}_{i-1}\underbrace{\emptyset^*B\emptyset^*A}_{i}\underbrace{BB\emptyset^*A}_{i+1} \dots \underbrace{BB\emptyset^*A}_{n}\emptyset^*$$

(with omitted $\bot^* = \varepsilon$ in BB). One can show that any unique characterisation of $\boldsymbol{q}_n$ contains at least 2^n positive examples to separate it from all queries $\boldsymbol{q}_n \wedge \exists P.\boldsymbol{s}$ with

$$\boldsymbol{s} = \boldsymbol{o}_1(B \wedge \Diamond(A \wedge \boldsymbol{o}_2(B \wedge \Diamond(A \wedge \dots \wedge \boldsymbol{o}_n(B \wedge \Diamond A) \dots)))),$$

where $\boldsymbol{o}_i$ is $\bigcirc$ or $\Diamond\bigcirc$ if $i > 1$, and blank or $\Diamond$ if $i = 1$. ❑

The situation changes drastically if we do not admit temporal operators in the scope of $\exists P$. We start by investigating the class $\mathcal{Q}_p[\bigcirc, \Diamond_r](\mathcal{ELI})$ of queries of the form

$$\boldsymbol{q} = \boldsymbol{r}_0 \wedge \boldsymbol{o}_1(\boldsymbol{r}_1 \wedge \boldsymbol{o}_2(\boldsymbol{r}_2 \wedge \cdots \wedge \boldsymbol{o}_n \boldsymbol{r}_n)),$$

where the $\boldsymbol{r}_i$ are $\mathcal{ELI}$-queries and $\boldsymbol{o}_i \in \{\bigcirc, \Diamond_r\}$. We can generalise the CQ-representation, the normal form, and the notion of lone conjunct from $\mathcal{Q}_p[\bigcirc, \Diamond_r]$ to $\mathcal{Q}_p[\bigcirc, \Diamond_r](\mathcal{ELI})$ in a straightforward way. To formulate conditions $(\mathbf{n1})$–$(\mathbf{n4})$, we replace the set inclusions '$\rho_i \subseteq \rho_j$' by entailment '$\boldsymbol{r}_i \models \boldsymbol{r}_j$'. For example, $(\mathbf{n4})$ becomes

(n4′) $\boldsymbol{r}_0^{i+1} \not\models \boldsymbol{r}_{k_i}^i$ if $i > 0$, $\boldsymbol{q}_i$ is primitive and R_{i+1} is $\leq$.

The condition for lone conjuncts now requires that $\boldsymbol{r}$ is not equivalent to any $\boldsymbol{q}_1 \wedge \boldsymbol{q}_2$ with $\mathcal{ELI}$-queries $\boldsymbol{q}_1, \boldsymbol{q}_2$ such that $\boldsymbol{q}_i \not\models \boldsymbol{r}$ for $i = 1, 2$. Then one can show again that every $\mathcal{Q}_p[\bigcirc, \Diamond_r](\mathcal{ELI})$-query is equivalent to a query in normal form, which can be computed in polynomial time.

Theorem 27. *The statements of Theorem 8 (i)–(iv) also hold if one replaces $\mathcal{Q}_p[\bigcirc, \Diamond_r]$ by $\mathcal{Q}_p[\bigcirc, \Diamond_r](\mathcal{ELI})$.*

The proof generalises the example set defined in Theorem 8 using the frontiers provided by Theorem 24 as a *black box*. Indeed, in the definition of examples replace ρ_i by $\hat{\boldsymbol{r}}_i$, the data instance corresponding to the $\mathcal{ELI}$-query $\boldsymbol{r}_i$, and replace '$\rho \backslash \{A\}$ for $A \in \rho$' by 'the data instance corresponding to a query in $\mathcal{F}(\boldsymbol{r})$'. We choose a single individual, a, as the root of these data instances. For example, item (a) becomes:

(a′) replacing some $\boldsymbol{r}_j^i$ by the data instance corresponding to a query in $\mathcal{F}(\boldsymbol{r}_j^i)$ or removing the whole $\boldsymbol{r}_j^i = \emptyset$ for $i \neq 0$ and $j \neq 0$ from some $\boldsymbol{q}_i$.

Next consider the class $\mathcal{Q}_p[\mathsf{U}](\mathcal{L})$ of queries of the form

$$\boldsymbol{q} = \boldsymbol{r}_0 \wedge (\boldsymbol{l}_1 \,\mathsf{U}\, (\boldsymbol{r}_1 \wedge (\boldsymbol{l}_2 \,\mathsf{U}\, (\ldots (\boldsymbol{l}_n \,\mathsf{U}\, \boldsymbol{r}_n) \ldots)))), \quad (8)$$

where $\boldsymbol{r}_i$ is an $\mathcal{L}$-query and $\boldsymbol{l}_i$ is either an $\mathcal{L}$-query or $\bot$, for $\mathcal{L} \in \{\mathcal{EL}, \mathcal{ELI}\}$. For the same reason as in the 1D case, we fix a finite signature Σ of predicate symbols. Denote by $\mathcal{L}(\Sigma)$ and $\mathcal{Q}_p[\mathsf{U}](\mathcal{L})$ the set of queries in $\mathcal{L}$ and $\mathcal{Q}_p^{\Sigma}[\mathsf{U}](\mathcal{L})$, respectively, with predicate symbols in Σ. Aiming to generalise Theorem 14, we again translate set-inclusion to entailment, so the *peerless queries* $\mathcal{P}^{\Sigma}[\mathsf{U}](\mathcal{L})$ take the form (8) such that either $\boldsymbol{l}_i = \bot$ or $\boldsymbol{l}_i \not\models \boldsymbol{r}_i$ and $\boldsymbol{r}_i \not\models \boldsymbol{l}_i$.

Theorem 28. *Let Σ be a finite relational signature. Then $\mathcal{P}^{\Sigma}[\mathsf{U}](\mathcal{EL})$ is polynomially characterisable within $\mathcal{Q}_p^{\Sigma}[\mathsf{U}](\mathcal{EL})$, while $\mathcal{P}^{\Sigma}[\mathsf{U}](\mathcal{ELI})$ is exponentially characterisable within $\mathcal{Q}_p^{\Sigma}[\mathsf{U}](\mathcal{ELI})$.*

To prove Theorem 28, we generalise the example set from the proof of Theorem 14. The positive examples are straightforward: simply replace ρ_i and λ_i by the data instances corresponding to $\boldsymbol{r}_i$ and $\boldsymbol{l}_i$ (and choose a single root individual). For the negative examples, we have to generalise the construction of σ, $\sigma \setminus \{A\}$, and $\sigma \setminus \{A, B\}$. For σ this is straightforward as its role can now be played by the Σ-data instance $\mathcal{A}_\Sigma = \{A(a), R(a, a) \mid A, R \in \Sigma\}$ for which $\mathcal{A}_\Sigma \models \boldsymbol{q}(a)$ for all $\boldsymbol{q} \in \mathcal{ELI}(\Sigma)$. For $\sigma \backslash \{A\}$ and $\sigma \backslash \{A, B\}$, we require *split partners* defined as follows. Let Q be a finite set of $\mathcal{L}(\Sigma)$-queries. A set $\mathcal{S}(Q)$ of pointed Σ-data instances $(\mathcal{A}, a)$ is called a *split partner* of Q in $\mathcal{L}(\Sigma)$ if the following conditions are equivalent for all $\mathcal{L}(\Sigma)$-queries $\boldsymbol{q}'$:

- $\mathcal{A} \models \boldsymbol{q}'(a)$ for some $(\mathcal{A}, a) \in \mathcal{S}(Q)$;
- $\boldsymbol{q}' \not\models \boldsymbol{q}$ for all $\boldsymbol{q} \in Q$.

Example 29. The split partner $\mathcal{S}(\{A\})$ of $\{A\}$ in $\mathcal{EL}(\Sigma)$ is the singleton set containing $\mathcal{A}_\Sigma^{-A}$ defined as

$$\{B(a), R(a, b), R(b, b), B'(b) \mid B \in \Sigma \setminus \{A\}, R, B' \in \Sigma\}.$$

Theorem 30. *Let $n > 0$ be fixed. For every set Q of $\mathcal{EL}(\Sigma)$-queries with $|Q| \leq n$, one can compute in polynomial time a split partner $\mathcal{S}(Q)$ of Q in $\mathcal{EL}(\Sigma)$. For $\mathcal{ELI}$, one can compute a split partner in exponential time.*

The proof, given in the full version, requires (as does $\mathcal{A}_\Sigma$) the construction of non-tree-shaped data instances. The exponential upper bound for $\mathcal{ELI}$ follows from results on generalised dualities for homomorphisms between relational structures (Foniok, Nesetril, and Tardif 2008; Nesetril and Tardif 2005). It remains open whether a polynomial construction is possible for our particular case.

We obtain the negative examples for $\boldsymbol{q}$ of the form (8) by taking the following data instances $\mathcal{D}$:

($\mathfrak{n}_0'$) $\mathcal{A}_\Sigma^n$ and $\mathcal{A}_\Sigma^{n-i} \mathcal{A} \mathcal{A}_\Sigma^i$, for $\mathcal{A} \in \mathcal{S}(\{\boldsymbol{r}_i\})$;

($\mathfrak{n}_1'$) $\mathcal{D} = \hat{\boldsymbol{r}}_0 \ldots \hat{\boldsymbol{r}}_{i-1} \mathcal{A} \hat{\boldsymbol{r}}_i \ldots \hat{\boldsymbol{r}}_n$ such that $\mathcal{D} \not\models \boldsymbol{q}$, where $\mathcal{A} \in \mathcal{S}(\{\boldsymbol{l}_i, \boldsymbol{r}_i\}) \cup \mathcal{S}(\{\boldsymbol{l}_i\}) \cup \mathcal{S}(\{\boldsymbol{r}_i\}) \cup \{\mathcal{A}_\Sigma\}$;

($\mathfrak{n}_2'$) for all i and $\mathcal{A} \in \mathcal{S}(\{\boldsymbol{l}_i\}) \cup \{\mathcal{A}_\Sigma\}$, *some* data instance

$$\mathcal{D}_{\mathcal{A}}^i = \hat{\boldsymbol{r}}_0 \ldots \hat{\boldsymbol{r}}_{i-1} \mathcal{A} \hat{\boldsymbol{r}}_i \hat{\boldsymbol{l}}_{i+1}^{k_{i+1}} \hat{\boldsymbol{r}}_{i+1} \ldots \hat{\boldsymbol{l}}_n^{k_n} \boldsymbol{r}_n,$$

if any, such that $\max(\mathcal{D}_{\mathcal{A}}^i) \leq (n+1)^2$ and $\mathcal{D}_{\mathcal{A}}^i \not\models \boldsymbol{q}^\dagger$ for $\boldsymbol{q}^\dagger$ obtained from $\boldsymbol{q}$ by replacing all $\boldsymbol{l}_j$, for $j \leq i$, with $\bot$.

We illustrate the construction by generalising Example 2.

Example 31. For $\boldsymbol{q} = \bigcirc A$ and any relational signature Σ containing A, we obtain, after removing redundant data instances, E^+ with $\emptyset\{A(a)\}$ and E^- with $\mathcal{A}_\Sigma \mathcal{A}_\Sigma^{-A}\{A(a)\}$.

We finally generalise Theorem 19 (ii) (part (i) is not interesting since simple queries do not generalise to any new class of $\mathcal{ELI}$-queries). Query classes such as $\mathcal{Q}[\Diamond](\mathcal{EL})$ are defined in the obvious way by replacing in $\mathcal{Q}[\Diamond]$-queries conjunctions of atoms by $\mathcal{EL}$-queries.

Theorem 32. *The class $\mathcal{Q}_b[\Diamond](\mathcal{EL}) \cap \mathcal{Q}_{\leq n}[\Diamond](\mathcal{EL})$ is polynomially characterisable for any $n < \omega$.*

Again the positive and negative examples are obtained from the 1D case by replacing σ by $\mathcal{A}_\Sigma$ and $\sigma \setminus \{A\}$ by appropriate split partners.

9 Applications to Learning

We apply our results on unique characterisability to exact learnability of temporal instance queries. Given a class $\mathcal{Q}$ of such queries, we aim to identify a *target query* $\boldsymbol{q} \in \mathcal{Q}$ using queries to an oracle. The learner knows $\mathcal{Q}$ and the signature σ (Σ in the 2D case) of $\boldsymbol{q}$. We allow only one type of queries, called *membership queries*, in which the learner picks a σ-data instance $\mathcal{D}$ and asks the oracle whether $\mathcal{D} \models \boldsymbol{q}$ holds. (In the 2D case, the learner picks a pointed Σ-data instance $(\mathcal{D}, a)$ and asks whether $\mathcal{D}, a, 0 \models \boldsymbol{q}$ holds.) The oracle answers 'yes' or 'no' truthfully. The class $\mathcal{Q}$ is (*polynomial*

time) *learnable with membership queries* if there exists an algorithm that halts for any $\boldsymbol{q} \in \mathcal{Q}$ and computes (in polynomial time in the size of $\boldsymbol{q}$ and σ/Σ), using membership queries, a query $\boldsymbol{q}' \in \mathcal{Q}$ that is equivalent to $\boldsymbol{q}$. By default, the learner does not know $|\boldsymbol{q}|$ in advance but reflecting Theorem 8 (iii), we also consider the case when $|\boldsymbol{q}|$ is known (which is common in active learning).

Obviously, unique characterisability is a necessary condition for learnability with membership queries. Conversely, if there is an algorithm that computes, for every $\boldsymbol{q} \in \mathcal{Q}$, an example set that uniquely characterises $\boldsymbol{q}$ within $\mathcal{Q}^{sig(\boldsymbol{q})}$, then $\mathcal{Q}$ is learnable with membership queries: enumerate $\mathcal{Q}^{sig(\boldsymbol{q})}$ starting with the smallest query $\boldsymbol{q}$, compute a characterising set E for $\boldsymbol{q}$ and check using membership queries whether $\boldsymbol{q}$ is equivalent to the target query. Eventually the algorithm will terminate with a query that is equivalent to the target query. As all our positive results on unique characterisability provide algorithms computing example sets, we directly obtain learnability with membership queries. Moreover, if the examples sets are computed in exponential time, then we obtain an exponential-time learning algorithm: in the enumeration above only $|sig(\boldsymbol{q})|^{|\boldsymbol{q}|}$ queries are checked before the target query is found. Unfortunately, we cannot infer polynomial-time learnability from polynomial characterisability in this way. In fact, monotone DNFs (disjunctions of conjunctions of propositional atoms) are a class of queries, which is polynomially characterisabile (simply take the models corresponding to the disjuncts as positive examples and the models obtained from them by removing an atom as negative examples) but not polynomial-time learnable with membership queries (Angluin 1987b).

A detailed analysis of polynomial-time learnability using membership queries is beyond the scope of this paper. Instead, we focus on one main result, the polynomial-time learnability of $\mathcal{Q}_p[\bigcirc, \Diamond](\mathcal{ELI})$.

Theorem 33. (i) *The class of safe queries in* $\mathcal{Q}_p[\bigcirc, \Diamond_r](\mathcal{ELI})$ *is polynomial-time learnable with membership queries.*

(ii) *The class* $\mathcal{Q}_p[\bigcirc, \Diamond_r](\mathcal{ELI})$ *is polynomial-time learnable with membership queries if the learner knows the size of the target query in advance.*

(iii) *The class* $\mathcal{Q}_p[\bigcirc, \Diamond](\mathcal{ELI})$ *is polynomially-time learnable with membership queries.*

Proof. We consider the 1D case without $\mathcal{ELI}$-queries first. (i) Our proof strategy is to construct a query $\boldsymbol{q}'$ that agrees with $\boldsymbol{q}$ on the positive and negative examples for $\boldsymbol{q}'$ from Theorem 8. The algorithm proceeds by computing a data instance $\mathcal{D}$. Our aim is to arrive at $\mathcal{D}_b$ through iterations of steps, from which the required query can be 'read off'.

Step 1. First, identify the number of $\bigcirc$ and $\Diamond$ in $\boldsymbol{q}$ by asking membership queries of the form σ^k incrementally, starting from $k = 1$, and then set $b = \min\{k \mid \sigma^k \models \boldsymbol{q}\} + 1$ and $\mathcal{D}_0 = \sigma^b$. Initialise $\mathcal{D} = \mathcal{D}_0$.

Step 2. Suppose that a data instance $\mathcal{D}'$ is obtained from $\mathcal{D}$ by applying one of the rules (a)–(e) of Theorem 8. If $\mathcal{D}' \models \boldsymbol{q}$ then replace $\mathcal{D}$ with $\mathcal{D}'$. Repeat as long as possible. One can show that the number of applications of each rule is bounded by a polynomial in $|\sigma|$ and the size of $\boldsymbol{q}$, and so **step 2** finishes in polynomial time.

Step 3. Suppose $\mathcal{D}$ contains $\emptyset^b \rho_0^i \emptyset^b$ and $|\rho_0^i| \geq 2$. Since rule (a) does not apply, every homomorphism $h\colon \boldsymbol{q} \to \mathcal{D}$ sends some $t_1, \dots, t_l$ to ρ_0^i, for $l \geq 1$. As $\boldsymbol{q}$ does not contain lone conjuncts, $\boldsymbol{q}$ contains singleton primitive blocks at positions $t_1, \dots, t_l$. Suppose $\rho_0^i = \{A_1, \dots, A_{|\rho_0^i|}\}$ and let $w = \{A_1\}\emptyset^b\{A_2\}\emptyset^b \dots \{A_{|\rho_0^i|}\}\emptyset^b$ (the order in which $A_1, \dots, A_{|\rho_0^i|}$, the elements of ρ_0^i, are enumerated does not matter, we fix any one). Let $\mathcal{D}_k^i$ be obtained from $\mathcal{D}$ by replacing $\emptyset^b \rho_0^i \emptyset^b$ with $\emptyset^b (w)^k$. Notice that, for $k = |\boldsymbol{q}|$, we have $\mathcal{D}_k^i \models \boldsymbol{q}$; however, the algorithm is not given this k. Instead, the algorithm incrementally iterates starting from $k = 1$ until $\mathcal{D}_k^i \models \boldsymbol{q}$. Since $k \leq |\boldsymbol{q}|$, this takes polynomially-many iterations. Let $\mathcal{D}'$ be obtained from $\mathcal{D}_k^i$ by removing primitive blocks as long as $\mathcal{D}' \models \boldsymbol{q}$. Notice that rules (a)–(e) do not apply to $\mathcal{D}'$. Replace $\mathcal{D}$ with $\mathcal{D}'$. Repeat **step 3** as long as possible. Since no new lone conjuncts are introduced, the process finishes after polynomially-many steps.

Step 4. At this point of computation, the algorithm has identified all blocks of $\boldsymbol{q}$ but not the sequences of $\Diamond$ and $\Diamond_r$ between them. They can be easily determined based on the positive and negative examples $\mathcal{D}_i$ and $\mathcal{D}_i^-$.

The proof of (ii) is similar, with a modified **step 3**. Finally, (iii) is a consequence of (ii) as the size of the query $\boldsymbol{q}$ does not exceed $n = |\sigma|b$.

We obtain a learning algorithm for $\mathcal{Q}_p[\bigcirc, \Diamond_r](\mathcal{ELI})$ by combining the learning algorithm above with the learning algorithm for $\mathcal{ELI}$-queries by ten Cate and Dalmau (2021) using the positive and negative examples given in Theorem 27. Note that the data instance $\mathcal{A}_\Sigma$ is now used instead of σ and that one has to 'unfold' such non tree-shaped data instances into tree-shaped ones. ❑

10 Conclusions

In this paper, we have considered temporal instance queries with *LTL* operators and started investigating their unique (polynomial) characterisability and exact learnability using membership queries. We have obtained both positive and negative results, depending on the available temporal operators and the allowed interaction between the temporal and object dimensions in queries. The results indicate that finding complete classifications of 1D and 2D temporal queries according to (polynomial) characterisability and learnability could be a very difficult task. In particular, interesting open problems include the polynomial characterisability of full $\mathcal{Q}_p^\sigma[\mathsf{U}]$, more general criteria of polynomial characterisability for temporal branching queries and other temporal operators, the existence of polynomial split partners for $\mathcal{ELI}$-queries, and the polynomial-time learnability of $\mathcal{Q}_p^\sigma[\mathsf{U}]$ and 2D extensions. From a conceptual viewpoint, it would be of interest to develop a framework that spells out explicitly the conditions that non-temporal queries should satisfy so that their combination with *LTL*-queries preserves polynomial characterisability and polynomial-time learnability.

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

Aarts, F., and Vaandrager, F. 2010. Learning i/o automata. In *International Conference on Concurrency Theory*, 71–85. Springer.

Alexe, B.; ten Cate, B.; Kolaitis, P. G.; and Tan, W. C. 2011. Characterizing schema mappings via data examples. *ACM Trans. Database Syst.* 36(4):23.

Angluin, D.; Frazier, M.; and Pitt, L. 1992. Learning conjunctions of Horn clauses. *Mach. Learn.* 9:147–164.

Angluin, D. 1987a. Learning regular sets from queries and counterexamples. *Inf. Comput.* 75(2):87–106.

Angluin, D. 1987b. Queries and concept learning. *Mach. Learn.* 2(4):319–342.

Arenas, M., and Diaz, G. I. 2016. The exact complexity of the first-order logic definability problem. *ACM Trans. Database Syst.* 41(2):13:1–13:14.

Artale, A.; Kontchakov, R.; Kovtunova, A.; Ryzhikov, V.; Wolter, F.; and Zakharyaschev, M. 2017. Ontology-mediated query answering over temporal data: A survey (invited talk). In *Proc. of TIME 2017*, volume 90 of *LIPIcs*, 1:1–1:37. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Baader, F.; Horrocks, I.; Lutz, C.; and Sattler, U. 2017. *An Introduction to Description Logics*. Cambride University Press.

Baader, F.; Borgwardt, S.; and Lippmann, M. 2015. Temporal query entailment in the description logic SHQ. *J. Web Semant.* 33:71–93.

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

Borgwardt, S., and Thost, V. 2015. Temporal query answering in the description logic EL. In *Proc. of IJCAI 2015*, 2819–2825. AAAI Press.

Camacho, A., and McIlraith, S. A. 2019. Learning interpretable models expressed in linear temporal logic. In *Proc. of ICAPS 2018*, 621–630. AAAI Press.

Cassel, S.; Howar, F.; Jonsson, B.; and Steffen, B. 2016. Active learning for extended finite state machines. *Formal Aspects Comput.* 28(2):233–263.

Chomicki, J., and Toman, D. 2018. *Temporal Logic in Database Query Languages*. New York, NY: Springer New York. 3992–3998.

Fijalkow, N., and Lagarde, G. 2021. The complexity of learning linear temporal formulas from examples. *CoRR* abs/2102.00876.

Foniok, J.; Nesetril, J.; and Tardif, C. 2008. Generalised dualities and maximal finite antichains in the homomorphism order of relational structures. *Eur. J. Comb.* 29(4):881–899.

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

Funk, M.; Jung, J. C.; and Lutz, C. 2021. Actively learning concepts and conjunctive queries under ELr-ontologies. In *Proc. of IJCAI 2021*, 1887–1893. ijcai.org.

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

Hodkinson, I. M.; Wolter, F.; and Zakharyaschev, M. 2000. Decidable fragment of first-order temporal logics. *Ann. Pure Appl. Log.* 106(1-3):85–134.

Howar, F., and Steffen, B. 2018. Active automata learning in practice - an annotated bibliography of the years 2011 to 2016. In *Machine Learning for Dynamic Software Analysis: Potentials and Limits, International Dagstuhl Seminar 16172*, volume 11026 of *Lecture Notes in Computer Science*, 123–148. Springer.

Jung, J. C.; Lutz, C.; Pulcini, H.; and Wolter, F. 2020. Logical separability of incomplete data under ontologies. In *Proc. of KR 2020*, 517–528.

Kolaitis, P. G. 2011. Schema Mappings and Data Examples: Deriving Syntax from Semantics (Invited Talk). In *Proc. of FSTTCS 2011*, volume 13 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 25–25. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Konev, B.; Lutz, C.; Ozaki, A.; and Wolter, F. 2017. Exact learning of lightweight description logic ontologies. *J. Mach. Learn. Res.* 18:201:1–201:63.

Konev, B.; Ozaki, A.; and Wolter, F. 2016. A model for learning description logic ontologies based on exact learning. In *Proc. of AAAI*, 1008–1015. AAAI Press.

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

Lemieux, C.; Park, D.; and Beschastnikh, I. 2015. General ltl specification mining (t). In *Proc. of ASE*, 81–92. IEEE.

Lutz, C.; Wolter, F.; and Zakharyaschev, M. 2008. Temporal description logics: A survey. In *Proc. of TIME 2008*, 3–14. IEEE Computer Society.

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

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

Nesetril, J., and Tardif, C. 2005. Short answers to exponentially long questions: Extremal aspects of homomorphism duality. *SIAM J. Discret. Math.* 19(4):914–920.

Schild, K. 1993. Combining terminological logics with tense logic. In *Proc. of the 6th Portuguese Conf. on Progress in Artificial Intelligence, EPIA'93*, volume 727 of *Lecture Notes in Computer Science*, 105–120. Springer.

Shahbaz, M., and Groz, R. 2009. Inferring mealy machines. In *International Symposium on Formal Methods*, 207–222. Springer.

ten Cate, B., and Dalmau, V. 2021. Conjunctive queries: Unique characterizations and exact learnability. In *Proc. of ICDT 2021*, volume 186 of *LIPIcs*, 9:1–9:24. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

ten Cate, B.; Kolaitis, P. G.; Qian, K.; and Tan, W. 2018. Active learning of GAV schema mappings. In *Proc. of PODS 2018*, 355–368. ACM.

ten Cate, B.; Dalmau, V.; and Kolaitis, P. G. 2013. Learning schema mappings. *ACM Trans. Database Syst.* 38(4):28:1–28:31.

A Proofs for Section 4

We prove the claims made in the introduction and Example 1.

(1) The query $\boldsymbol{q} = \Diamond_r(A \wedge B)$ is not uniquely characterisable within $\mathcal{Q}_p[\Diamond_r]$. Indeed, consider the queries $\boldsymbol{q}_1 = \Diamond_r(A \wedge \Diamond_r B)$ and $\boldsymbol{q}_i = \Diamond_r(A \wedge \Diamond_r(B \wedge \Diamond_r \boldsymbol{q}_{i-1}))$. Clearly, $\boldsymbol{q} \models \boldsymbol{q}_i$ and $\boldsymbol{q}_i \not\models \boldsymbol{q}$ for all $i \geq 1$. Suppose $\boldsymbol{q}$ fits (E^+, E^-) and n is the length of the longest example in E^-. Then $\boldsymbol{q}_{n+1}$ also fits (E^+, E^-) as $\mathcal{D}, 0 \not\models \Diamond_r(A \wedge B)$, and so $\mathcal{D}, 0 \not\models \boldsymbol{q}_{n+1}$, for any $\mathcal{D} \in E^-$.

(2) The query $\boldsymbol{q} = \bot \mathsf{U} A$ (i.e., $\bigcirc A$) is not uniquely characterisable within $\mathcal{Q}^{\{A\}}[\mathsf{U}]$. For suppose $\boldsymbol{q}$ fits (E^+, E^-) and n is the length of the longest example in E^-. Consider $\boldsymbol{q}' = (\bigcirc^{n+1} A) \mathsf{U} A$. Clearly, $\boldsymbol{q}' \not\models \boldsymbol{q}$ and $E \models \boldsymbol{q}'$ ($\mathcal{D} \not\models \boldsymbol{q}'$, for any $\mathcal{D} \in E^-$, because $\mathcal{D}, 1 \not\models A$ and $\mathcal{D}, 1 \not\models \bigcirc^{n+1} A$).

(3) While the query $\Diamond_r(A \wedge B)$ is not characterisable, there is a polynomial f such that for all $n \in \mathbb{N}$, it is characterisable within $\mathcal{Q}_p^n[\bigcirc, \Diamond_r]$ by some E_n of size $\leq f(n)$. Take $E_n = (E^+, E^-)$ with $E^+ = \{(\{A, B\}), (\emptyset, \{A, B\})\}$ and $E^- = \{(\underbrace{\{A\}, \{B\}, \ldots, \{A\}, \{B\}}_{n \text{ times}})\}$. If $\boldsymbol{q}' \in \mathcal{Q}_p^n[\bigcirc, \Diamond_r]$ fits E then we can assume without loss of generality that it does not use $\bigcirc$, as $(\{A, B\}) \models \boldsymbol{q}'$. This means that $\boldsymbol{q}'$ is of the form $\rho_o \wedge \Diamond_r(\rho_1 \wedge \Diamond_r(\rho_2 \wedge \cdots \wedge (\Diamond_r \rho_m) \cdots))$, with $m < n$. Moreover, $\rho_0 = \emptyset$ as $(\emptyset, \{A, B\}) \models \boldsymbol{q}'$. And since $(\{A, B\}) \models \boldsymbol{q}'$, we must have $\rho_i \subseteq \{A, B\}$ for all i. Finally, as $(\underbrace{\{A\}, \{B\}, \ldots, \{A\}, \{B\}}_{n \text{ times}}) \not\models \boldsymbol{q}'$, there must be i such that $\rho_i = \{A, B\}$. Thus, $\boldsymbol{q}' \equiv \Diamond_r(A \wedge B)$.

B Proofs for Section 5

We show Theorem 8. We use the fact that $\mathcal{D} \models \boldsymbol{q}$ iff there is a *homomorphism* h from the set $\mathit{var}(\boldsymbol{q})$ of variables in $\boldsymbol{q}$ to $[0, \max(\mathcal{D})]$, i.e., $h(t_0) = 0$, $A(h(t)) \in \mathcal{D}$ if $A(t) \in \boldsymbol{q}$, $h(t') = h(t) + 1$ if $\mathit{suc}(t, t') \in \boldsymbol{q}$, and $h(t)\, R\, h(t')$ if $R(t, t') \in \boldsymbol{q}$ for $R \in \{<, \leq\}$.

Theorem 8. *(i) A query $\boldsymbol{q} \in \mathcal{Q}_p[\bigcirc, \Diamond_r]$ is uniquely characterisable within $\mathcal{Q}_p[\bigcirc, \Diamond_r]$ iff $\boldsymbol{q}$ is safe.*

(ii) Those queries that are uniquely characterisable within $\mathcal{Q}_p[\bigcirc, \Diamond_r]$ are actually polynomially characterisable within $\mathcal{Q}_p[\bigcirc, \Diamond_r]$.

(iii) The class $\mathcal{Q}_p[\bigcirc, \Diamond_r]$ is polynomially characterisable for bounded query size.

(iv) The class $\mathcal{Q}_p[\bigcirc, \Diamond]$ is polynomially characterisable.

Proof. *(i)* Assume that $\boldsymbol{q}$ is given. To show ($\Leftarrow$), we may assume $\boldsymbol{q}$ in normal form (4) does not contain lone conjuncts. Let b be the number of $\bigcirc$ and $\Diamond$ in $\boldsymbol{q}$ plus 1. Consider the example set $E = (E^+, E^-)$ constructed in the main paper. Then E is polynomial in $|\boldsymbol{q}|$ and $\mathcal{D} \models \boldsymbol{q}$ for all $\mathcal{D} \in E^+$. Using the condition that $\boldsymbol{q}$ is in normal form, we show the following:

Claim 1. *(i) There is only one homomorphism $h\colon \boldsymbol{q} \to \mathcal{D}_b$, and it maps isomorphically each $\boldsymbol{q}_i$ onto $\bar{\boldsymbol{q}}_i$.*

(ii) $\mathcal{D}_i^- \not\models \boldsymbol{q}$, for any $\mathcal{R}_i$ different from $\leq$.

(iii) If $\mathcal{D}_b'$ is obtained from $\mathcal{D}_b$ by replacing some $\bar{\boldsymbol{q}}_i$ with $\bar{\boldsymbol{q}}_i'$ such that $\bar{\boldsymbol{q}}_i', \ell \not\models \boldsymbol{q}_i$ for any $\ell \leq \max(\bar{\boldsymbol{q}}_i')$, then $\mathcal{D}_b' \not\models \boldsymbol{q}$, and so $\mathcal{D} \not\models \boldsymbol{q}$, for all $\mathcal{D} \in E^-$.

Proof of claim. *(i)* As $\boldsymbol{q}$ is in normal form and the gaps between $\bar{\boldsymbol{q}}_i$ and $\bar{\boldsymbol{q}}_{i+1}$ are not shorter than any block in $\boldsymbol{q}$, each block $\boldsymbol{q}_i$ in $\boldsymbol{q}$ is mapped by h to a single block $\bar{\boldsymbol{q}}_j$ of D_b. The function $f\colon [0, n] \to [0, n]$ defined by taking $f(i) = j$ is such that $f(0) = 0$, $i < j$ implies $f(i) \leq f(j)$, and block $\boldsymbol{q}_i$ is satisfied in $\bar{\boldsymbol{q}}_{f(i)}$. It also follows from the definition of normal form that if $f(i) = i$, then h isomorphically maps $\boldsymbol{q}_i$ onto $\bar{\boldsymbol{q}}_i$ and $f(i-1) < i$ and $f(i+1) > i$. To show that $f(i) = i$ for all i, we first observe that $f(1) \geq 1$ and $f(j) = j$, for $j = \max\{i \mid f(i) \geq i\}$, from which $f(j-1) < j$ and $f(j+1) > j$. Then we can proceed in the same way inductively by considering f restricted to the smaller intervals $[j, n]$ and $[0, j]$.

(ii) Suppose $\mathcal{R}_i$ is not $\leq$ but there is a homomorphism $h\colon \boldsymbol{q} \to \mathcal{D}_i^-$. Consider the location of $h(s_0^i) = \ell$. Suppose ℓ is in $\bar{\boldsymbol{q}}_i$. Since $\rho_{k_{i+1}}^{i+1} \neq \emptyset$ and by the construction of $\mathcal{D}_i^-$, $h(s_0^{i+1})$ lies in some $\bar{\boldsymbol{q}}_j$ with $j > i+1$. But then there is a homomorphism $h'\colon \boldsymbol{q} \to \mathcal{D}_b$ different from the one in *(i)*, which is impossible. We arrive to the same contradiction if we assume that ℓ lies in $\bar{\boldsymbol{q}}_j$ with $j < i$ or $j > i$.

(iii) is proved analogously. ❑

Thus, $\boldsymbol{q}$ fits E. Suppose now $\boldsymbol{q}'$ is any $\mathcal{Q}_p[\bigcirc, \Diamond, \Diamond_r]$-query in normal form. For $t \in \mathit{var}(\boldsymbol{q}')$, denote by τ_t the set of atoms A with $A(t) \in \boldsymbol{q}'$ and call it the *type* of t in $\boldsymbol{q}'$. Similarly, for $\ell \in [0, \max(\mathcal{D}_b)]$, denote by ρ_ℓ the set of atoms A with $A(\ell) \in \mathcal{D}_b$ and call it the *type* of ℓ in $\mathcal{D}_b$. A homomorphism $h\colon \boldsymbol{q}' \to \mathcal{D}_b$ is *block surjective* if every point in every block $\bar{\boldsymbol{q}}_i$ of $\mathcal{D}_b$ is in the range $\mathit{ran}(h)$ of h; it is *type surjective* if $\rho_\ell = \bigcup_{h(t)=\ell} \tau_t$ for all $\ell \in \mathit{ran}(h)$. The following claim follows immediately from the definitions:

Claim 2. *(i) If there is a homomorphism $h\colon \boldsymbol{q}' \to \mathcal{D}_b$ that is not block or type surjective, then $\mathcal{D} \models \boldsymbol{q}'$ for some $\mathcal{D} \in E^-$ obtained from $\mathcal{D}_b$ by (a).*

(ii) If there exist a homomorphism $h\colon \boldsymbol{q}' \to \mathcal{D}_b$ and $(t < t') \in \boldsymbol{q}'$ or $(t \leq t') \in \boldsymbol{q}'$ such that $h(t) \neq h(t')$ and $h(t), h(t') \in \bar{\boldsymbol{q}}_i$, for some block $\bar{\boldsymbol{q}}_i$, then $\mathcal{D} \models \boldsymbol{q}'$ for some $\mathcal{D} \in E^-$ obtained from $\mathcal{D}_b$ by (b).

Suppose now that $h\colon \boldsymbol{q}' \to \mathcal{D}_b$ is a block and type surjective homomorphism, $(t \leq t') \in \boldsymbol{q}'$ and $h(t) = h(t') = \ell$ lies in $\bar{\boldsymbol{q}}_i$. Then $h^{-1}(\ell) = \{t_1, \ldots, t_k\}$ with $k \geq 2$ and

$(t_j \leq t_{j+1}) \in \boldsymbol{q}'$, $1 \leq j < k$. By (n3) and (n4), $\tau_{t_j} \neq \emptyset$ for at least one t_j, and so $\rho_\ell \neq \emptyset$. Consider possible locations of ℓ in $\bar{\boldsymbol{q}}_i$.

Case 1: ℓ has both a left and a right neighbour in $\bar{\boldsymbol{q}}_i$. Then there is $\mathcal{D} \in E^-$ obtained by (c)—i.e., by replacing the appropriate ρ_l^i with $\rho_l^i \emptyset^b \rho_l^i$—and a homomorphism $h' \colon \boldsymbol{q}' \to \mathcal{D}$, which 'coincides' with h except that $h'(t_1)$ is the point with the first ρ_l^i and $h'(t_j)$, for $j = 2, \dots, k$, is the point with the second ρ_l^i.

Case 2: ℓ has no neighbours in $\bar{\boldsymbol{q}}_i$, so this block is primitive and ρ_ℓ is a singleton (as $\boldsymbol{q}$ has no lone conjuncts by our assumption). Then t_1 is the last variable in its block in $\boldsymbol{q}'$, t_k is the first variable in its block in $\boldsymbol{q}'$, and the t_i with $1 < i < k$, if any, are all primitive blocks. But then the types τ_{t_i} and $\tau_{t_{i+1}}$ are not comparable with respect to $\subseteq$, contrary to ρ_ℓ being a singleton. Thus, Case 2 cannot happen.

Case 3: ℓ has a left neighbour in $\bar{\boldsymbol{q}}_i$ but no right neighbour. As h is type surjective and in view of (n3), $\tau_{t_1} \subsetneq \rho_\ell$. Let $A \in \rho_\ell \setminus \tau_{t_1}$ and let $\mathcal{D} \in E^-$ be obtained by the first part of (d) by replacing $\rho_{k_i}^i$ with $\rho_{k_i}^i \setminus \{A\} \emptyset^b \rho_{k_i}^i$. Then there is a homomorphism $h' \colon \boldsymbol{q}' \to \mathcal{D}$ that sends t_1 to the point with $\rho_{k_i}^i \setminus \{A\}$ and the remaining t_j to the point with $\rho_{k_i}^i$.

Case 4: ℓ has a right neighbour in $\bar{\boldsymbol{q}}_i$, $i \neq 0$, but no left neighbour. This case is dual to Case 3 and we use the second part of (d).

Case 5: $\ell = 0$. If $\bar{\boldsymbol{q}}_0$ is primitive, then all of the t_i are primitive blocks in $\boldsymbol{q}'$. By (n3), $\tau_{t_2} \not\subseteq \tau_{t_1}$; by type surjectivity, $\tau_{t_1} \subsetneq \rho_\ell$, and so there is $A \in \rho_\ell \setminus \tau_{t_1}$. By the first part of (e), we have $\mathcal{D} \in E^-$ obtained by replacing ρ_0^0 with $\rho_0^0 \setminus \{A\} \emptyset^b \rho_0^0$. Then there is a homomorphism $h' \colon \boldsymbol{q}' \to \mathcal{D}$ that sends t_1 to the point with $\rho_0^0 \setminus \{A\}$ and the remaining t_j to the point with ρ_0^0. Finally, if $\bar{\boldsymbol{q}}_0$ is not primitive, the second part of (e) gives $\mathcal{D} \in E^-$ by replacing ρ_0^0 in $\mathcal{D}_b$ with $\rho_0^0 \emptyset^b \rho_0^0$. We obtain a homomorphism from $\boldsymbol{q}'$ to $\mathcal{D}$ by sending t_1 to the first ρ_0^0 and the remaining t_j to the second ρ_0^0.

It remains to consider the case when there is a homomorphism $h \colon \boldsymbol{q}' \to \mathcal{D}_b$ that is an isomorphism between the blocks in $\boldsymbol{q}'$ and the blocks in $\mathcal{D}_b$, and so the difference between $\boldsymbol{q}'$ and $\boldsymbol{q}$ can only be in the sequences of $\Diamond$ and $\Diamond_r$ between blocks. To be more precise, $\boldsymbol{q}$ is of the form (4),

$$\boldsymbol{q}' = \boldsymbol{q}_0 \mathcal{R}_1' \boldsymbol{q}_1 \dots \mathcal{R}_n' \boldsymbol{q}_n \tag{9}$$

and $\mathcal{R}_i \neq \mathcal{R}_i'$ for some i. Four cases are possible:

(i) $\mathcal{R}_i = (r_0 \leq r_1)$ and $\mathcal{R}_i' = (s_0 < s_1) \dots (s_{l-1} < s_l)$, for $l \geq 1$. In this case, $\mathcal{D}_i \not\models \boldsymbol{q}'$, for $\mathcal{D}_i \in E^+$.

(ii) $\mathcal{R}_i = (r_0 < r_1) \dots (r_{k-1} < r_k)$, $\mathcal{R}_i' = (s_0 < s_1) \dots (s_{l-1} < s_l)$, for $l > k$. Then again $\mathcal{D}_i \not\models \boldsymbol{q}'$.

(iii) $\mathcal{R}_i = (r_0 < r_1) \dots (r_{k-1} < r_k)$, $\mathcal{R}_i' = (s_0 \leq s_1)$, for $k \geq 1$. In this case $\mathcal{D}_i^- \models \boldsymbol{q}'$, for $\mathcal{D}_i^- \in E^-$.

(iv) $\mathcal{R}_i = (r_0 < r_1) \dots (r_{k-1} < r_k)$ and $\mathcal{R}_i' = (s_0 < s_1) \dots (s_{l-1} < s_l)$, for $l < k$. Then again $\mathcal{D}_i^- \models \boldsymbol{q}'$.

($\Rightarrow$) Suppose $\boldsymbol{q}$ in normal form (4) does contain a lone conjunct $\boldsymbol{q}_i = \rho$. Let ρ^- be the last type of the block $\boldsymbol{q}_{i-1}$ and let ρ^+ be the first type of the block $\boldsymbol{q}_{i+1}$. Then ρ is a disjoint union of some nonempty τ and τ' such that at least one of the queries $\boldsymbol{s}_1'$ or $\boldsymbol{s}_1''$ below is in normal form:

$$\boldsymbol{s}_1' = \boldsymbol{q}_0 \mathcal{R}_1 \dots \mathcal{R}_i \tau (\leq) \tau' \mathcal{R}_{i+1} \dots \mathcal{R}_n \boldsymbol{q}_n,$$
$$\boldsymbol{s}_1'' = \boldsymbol{q}_0 \mathcal{R}_1 \dots \mathcal{R}_i \tau (\leq) \tau' (\leq) \tau \mathcal{R}_{i+1} \dots \mathcal{R}_n \boldsymbol{q}_n$$

For example, if $\rho^- = \{A, A'\}$, $\rho = \{A, B\}$, $\rho^+ = \{A, B'\}$ and $\mathcal{R}_i$ and $\mathcal{R}_{i+1}$ are both $\leq$, we take $\tau = \{B\}$, $\tau' = \{A\}$, for which $\boldsymbol{s}_1'$ is not in normal form, while $\boldsymbol{s}_1''$ is. Pick one of $\boldsymbol{s}_1'$ and $\boldsymbol{s}_1''$, which is in normal form, and denote it by $\boldsymbol{s}_1$. For $n \geq 2$, let $\boldsymbol{s}_n$ be the query obtained from $\boldsymbol{s}_1$ by duplicating n times the part $\tau(\leq)\tau'$ in $\boldsymbol{s}_1$ and inserting $\leq$ between the copies. It is readily seen that $\boldsymbol{s}_n$ is in normal form. Clearly, $\boldsymbol{q} \models \boldsymbol{s}_n$ and, similarly to the proof of Claim 1, one can show that $\boldsymbol{s}_n \not\models \boldsymbol{q}$, for any $n \geq 1$.

Now suppose $E = (E^+, E^-)$ characterises $\boldsymbol{q}$ and let $n = \max\{\max(\mathcal{D}) \mid \mathcal{D} \in E^-\} + 1$. Then $E \models \boldsymbol{s}_n$, which is impossible. Indeed, consider any $\mathcal{D} \in E^-$. To show that $\mathcal{D} \not\models \boldsymbol{s}_n$, suppose otherwise. Then there is a homomorphism $h \colon \boldsymbol{s}_n \to \mathcal{D}$. By the pigeonhole principle, h maps some variables of types τ and τ' in $\boldsymbol{s}_n$ to the same point in $\mathcal{D}$. But then h can be readily modified to obtain a homomorphism $h' \colon \boldsymbol{q} \to \mathcal{D}$, contrary to $E^- \not\models \boldsymbol{q}$.

(ii) follows from the proof of (i) as (E^+, E^-) is of polynomial size.

(iii) We aim to characterize $\boldsymbol{q}$ in normal form (4) which may contain lone conjuncts within the class of queries in $\mathcal{Q}_p[\bigcirc, \Diamond]$ of size at most n, where n is the size of $\boldsymbol{q}$. The set E^+ of positive examples is defined as before and we extend the set of rules (a) to (e) in the definition of E^- as follows: if $\boldsymbol{q}_i = \rho(s)$ with $\rho = \{A_1, \dots, A_k\}$ is a block in $\mathcal{D}_b$ with ρ a lone conjunct in $\boldsymbol{q}$, then

(f) replace ρ with $(\rho \setminus \{A_1\} \emptyset^b \cdots \emptyset^b \rho \setminus \{A_k\})^n$.

For the proof that (E^+, E^-) characterizes $\boldsymbol{q}$ within the class of queries of size at most n, observe that with the exception of *Case 2* the proof of (i) still goes through. In Case 2, however, we can now apply the assumption that the size of $\boldsymbol{q}'$ is bounded by n as we then obtain a data instance $\mathcal{D} \in E^-$ and a homomorphism $h' : \boldsymbol{q}' \to \mathcal{D}$.

(iv) Assume $\boldsymbol{q}$ in $\mathcal{Q}_p[\bigcirc, \Diamond]$ is given. The proof of (i) shows that (E^+, E^-), defined in the same way as in (i) except that the rules (c),(d), and (e) are not used to construct E^-, characterizes $\boldsymbol{q}$ within $\mathcal{Q}_p[\bigcirc, \Diamond]$ even if $\boldsymbol{q}$ contains lone conjuncts. ❑

C Proofs for Section 6

Theorem 9. *Any $\mathcal{Q}_p^\sigma[\mathsf{U}]$-queries $\boldsymbol{q} \not\equiv \boldsymbol{q}'$ can be separated by some $\mathcal{D}$ with $\max(\mathcal{D}) \leq O((\min\{tdp(\boldsymbol{q}), tdp(\boldsymbol{q}')\})^2)$.*

Proof. Let $\boldsymbol{q} = \rho_0 \lambda_1^* \rho_1 \dots \rho_n \emptyset^*$ and $\boldsymbol{q}' = \tau_0 \mu_1^* \tau_1 \dots \tau_k \emptyset^*$. If $n < k$, then $\rho_0 \rho_1 \dots \rho_n$ separates $\boldsymbol{q}$ from $\boldsymbol{q}'$. Suppose $n = k$. If $\rho_i \subsetneq \tau_i$, for some $i \leq n$, then again $\rho_0 \rho_1 \dots \rho_n$ separates $\boldsymbol{q}$ from $\boldsymbol{q}'$. So suppose $\rho_i = \tau_i$ for all $i \leq n$.

Let $\boldsymbol{q} \not\models \boldsymbol{q}'$. Then there is $\mathcal{D} = \rho_0 \lambda_1^{k_1} \rho_1 \dots \lambda_n^{k_n} \rho_n \emptyset^{k_{n+1}}$ separating $\boldsymbol{q}$ from $\boldsymbol{q}'$. We show that $k_i \leq n+1$, for all $i \leq n$. To see this, we convert $\mathfrak{A}_{\boldsymbol{q}'}$ to a DFA $\mathfrak{B}_{\boldsymbol{q}'}$ using the subset construction and observe that whenever there are transitions

$Q_1 \to_{\lambda_i} \cdots \to_{\lambda_i} Q_{n+1} \to_{\lambda_i} Q_{n+2}$ in $\mathfrak{B}_{\boldsymbol{q}'}$ (with $Q_j \subseteq [0, n+1]$), then $Q_{n+1} = Q_{n+2}$ because, by the structure of $\mathfrak{A}_{\boldsymbol{q}'}$, we have $\delta'_{\alpha^{n+1}} = \delta'_{\alpha^{n+2}}$, for any α, where δ'_w is the transition function on the states of $\mathfrak{B}_{\boldsymbol{q}'}$ corresponding to the word w. ❑

Lemma 12. *For any queries $\boldsymbol{q}$ and $\boldsymbol{q}'$ as above, either (i) each $\lambda_i \neq \bot$ subsumes μ_j occurring in some matching pair (λ_k, μ_j) or (ii) $\boldsymbol{q}$ and $\boldsymbol{q}'$ are separated by a data instance of the form $\mathcal{D}^i_{\boldsymbol{q}}$ or $\mathcal{D}^j_{\boldsymbol{q}'}$. Also, if $\boldsymbol{q}$ is peerless, λ_i can only subsume μ_j in the matching pair (λ_i, μ_j) with $i \geq j$, in which case $\mu_j = \rho_j = \cdots = \rho_{i-1} = \lambda_i$.*

Proof. If $\lambda_i \neq \bot$ does not subsume any μ_j, then, as we know, $\mathcal{D}^i_{\boldsymbol{q}} \models \boldsymbol{q}$ and $\mathcal{D}^i_{\boldsymbol{q}} \not\models \boldsymbol{q}'$. So suppose λ_i subsumes some μ_j. Then either (λ_i, μ_j) is a matching pair or $\mu_j \subsetneq \lambda_i$. Note that the latter is impossible if $\boldsymbol{q}$ is peerless. If μ_j does not subsume any λ_l, then $\mathcal{D}^j_{\boldsymbol{q}'} \models \boldsymbol{q}'$ and $\mathcal{D}^j_{\boldsymbol{q}'} \not\models \boldsymbol{q}$. Otherwise, we consider λ_l subsumed by μ_j, etc. Since $\lambda_l \subsetneq \lambda_i$, sooner or later this process will terminate. ❑

Lemma 34. *Suppose that $\boldsymbol{q} = \rho_0 \lambda_1^* \rho_1 \lambda_2^* \dots \lambda_n^* \rho_n \emptyset^*$ and $\boldsymbol{q}' = \rho_0 \rho_1 \lambda_2^* \dots \lambda_n^* \rho_n \emptyset^*$ with $\boldsymbol{q} \not\models \boldsymbol{q}'$. Then there is a data instance $\mathcal{D}$ of the form $\rho_0 \lambda_1 \rho_1 \lambda_2^{k_2} \dots \lambda_n^{k_n} \rho_n$ such that $\mathcal{D} \models \boldsymbol{q}$ and $\mathcal{D} \not\models \boldsymbol{q}'$.*

Proof. Take any $\mathcal{D} = \rho_0 \lambda_1^{k_1} \rho_1 \lambda_2^{k_2} \dots \lambda_n^{k_n} \rho_n$ with $\mathcal{D} \models \boldsymbol{q}$ and $\mathcal{D} \not\models \boldsymbol{q}'$. Let $\mathcal{D}_i$ be $\mathcal{D}$ with $k_1 = i$. Choose $\mathcal{D}_i \not\models \boldsymbol{q}'$ with $\mathcal{D}_l \models \boldsymbol{q}'$, for all $l < i$. Consider some $\mathcal{D}' = \rho_0 \rho_1 u$ with $\mathcal{D}' \Subset \mathcal{D}_{i-1}$ and set $\mathcal{D}'' = \rho_0 \lambda_1 \rho_1 u$. If $\mathcal{D}'' \models \boldsymbol{q}'$, then $\mathcal{D}_i \models \boldsymbol{q}'$ as $\mathcal{D}'' \Subset \mathcal{D}_i$, which is impossible. Thus, $\mathcal{D}''$ is the data instance we need. ❑

Theorem 14. *$\mathcal{P}^\sigma[\mathsf{U}]$ is polynomially characterisable within $\mathcal{Q}^\sigma_p[\mathsf{U}]$.*

Proof. We show that any $\boldsymbol{q} = \rho_0 \lambda_1^* \rho_1 \lambda_2^* \dots \lambda_n^* \rho_n \emptyset^*$ in $\mathcal{P}^\sigma[\mathsf{U}]$ is characterised by $E = (E^+, E^-)$, where E^+ contains all data instances of the following forms:

$(\mathfrak{p}_0)$ $\rho_0 \dots \rho_n$,

$(\mathfrak{p}_1)$ $\rho_0 \dots \rho_{i-1} \lambda_i \rho_i \dots \rho_n = \mathcal{D}^i_{\boldsymbol{q}}$,

$(\mathfrak{p}_2)$ $\rho_0 \dots \rho_{i-1} \lambda_i^k \rho_i \dots \rho_{j-1} \lambda_j \rho_j \dots \rho_n = \mathcal{D}^j_{i,k}$, for $i < j$ and $k = 1, 2$;

and E^- has all instances that are *not* in $\boldsymbol{L}(\boldsymbol{q})$ of the forms:

$(\mathfrak{n}_0)$ σ^n and $\sigma^{n-i} \sigma \setminus \{A\} \sigma^i$, for $A \in \rho_i$,

$(\mathfrak{n}_1)$ $\rho_0 \dots \rho_{i-1} \sigma \setminus \{A, B\} \rho_i \dots \rho_n$, for $A \in \lambda_i \cup \{\bot\}$, $B \in \rho_i \cup \{\bot\}$,

$(\mathfrak{n}_2)$ for all i and $A \in \lambda_i \cup \{\bot\}$, *some* data instance

$$\mathcal{D}^i_A = \rho_0 \dots \rho_{i-1} (\sigma \setminus \{A\}) \rho_i \lambda_{i+1}^{k_{i+1}} \dots \lambda_n^{k_n} \rho_n, \quad (10)$$

if any, such that $\max(\mathcal{D}^i_A) \leq (n+1)^2$ and $\mathcal{D}^i_A \not\models \boldsymbol{q}^\dagger$ for $\boldsymbol{q}^\dagger$ obtained from $\boldsymbol{q}$ by replacing λ_j, for all $j \leq i$, with $\bot$. Note that $\mathcal{D}^i_A \not\models \boldsymbol{q}$ for peerless $\boldsymbol{q}$.

By definition, $\boldsymbol{q}$ fits E and $|E|$ is polynomial in $|\boldsymbol{q}|$. Suppose $\boldsymbol{q}' = \tau_0 \mu_1^* \tau_1 \dots \mu_m^* \tau_m \emptyset^*$ also fits E. By $(\mathfrak{p}_0)$ and $(\mathfrak{n}_0)$, we have $n = m$ and $\rho_i = \tau_i$, for $i \leq n$. Consider the maximal i with $\lambda_i \neq \mu_i$ (if there is no such, $\boldsymbol{q} \equiv \boldsymbol{q}'$).

Case 1: $\mu_i \neq \bot$, $\mu_i \not\subseteq \lambda_i$. By Lemma 12, if μ_i does not subsume any λ_j, then $\mathfrak{n}_1$ separates $\boldsymbol{q}$ and $\boldsymbol{q}'$. So suppose μ_i subsumes λ_j. As $\boldsymbol{q}$ is peerless, $j > i$ and we have $\rho_i \subseteq \mu_i$. By the minimality of $\boldsymbol{q}'$, there is $\mathcal{D}$ such that $\mathcal{D} \models \rho_{i-1} \mu_i^* \rho_i \dots \mu_n^* \rho_n \emptyset^*$ but $\mathcal{D} \not\models \rho_{i-1} \rho_i \dots \mu_n^* \rho_n \emptyset^*$. By Lemma 34, we can choose $\mathcal{D} = \rho_{i-1} \mu_i \rho_i u$. Consider $\mathcal{D}' = \rho_{i-1}(\sigma \setminus \{A\}) \rho_i u$, where $A \in \lambda_i \setminus \mu_i$ if $\lambda_i \neq \bot$, or $A = \bot$ otherwise. Since $\rho_i \subseteq \mu_i \subseteq \sigma \setminus \{A\}$ and $\lambda_i \not\subseteq \sigma \setminus \{A\}$, we have $\rho_0 \dots \rho_{i-2} \mathcal{D}' \models \rho_0 \dots \rho_{i-1} (\sigma \setminus \{A\})^* \rho_i \lambda_{i+1}^* \dots \lambda_n^* \rho_n$ but $\mathcal{D}' \not\models \rho_0 \dots \rho_{i-1} \rho_i \lambda_{i+1}^* \dots \lambda_n^* \rho_n \emptyset^*$, for otherwise $\mathcal{D} \models \rho_{i-1} \rho_i \dots \mu_n^* \rho_n$ as $\lambda_j = \mu_j$ for all $j > i$. Therefore, there is $\mathcal{D}^i_A \in \mathfrak{n}_2$ with $\mathcal{D}^i_A \models \boldsymbol{q}'$.

Case 2: $\lambda_i \neq \bot$, $\mu_i \subsetneq \lambda_i$. By Lemma 12, if μ_i does not subsume any λ_j, then $\mathfrak{n}_1$ separates $\boldsymbol{q}$ and $\boldsymbol{q}'$. So suppose μ_i subsumes λ_j. As $\boldsymbol{q}$ is peerless, $j > i$ and we have $\rho_i \subseteq \mu_i$. But then $\rho_i \subseteq \lambda_i$, which is a contradiction.

Case 3: $\lambda_i \neq \bot$, $\mu_i = \bot$. Find the maximal $j < i$ such that λ_i subsumes μ_j. Then $\mu_j = \rho_j = \ldots = \rho_{i-1} = \lambda_i$ by Lemma 12.

Suppose $\lambda_{j'} = \bot$ for all $j' \in [j+1, i-1]$. By Lemma 12, $\mu_{j'} = \bot$ for all $j' \in [j+1, i-1]$. For if $\mu_{j'} \neq \bot$, then it either does not subsume anything, or it subsumes λ_i. In the latter case, we have $\rho_{j'-1} \subsetneq \mu_{j'}$ and either there is $j'' \in (j, j')$ with $\rho_{j''} \not\subset \mu_{j''}$, in which case $\mu_{j''}$ does not subsume anything, or we violate the minimality condition. The queries $\rho_{j-1} \mu_j^* \rho_j \dots \rho_{i-1} \rho_i \dots$ and $\rho_{j-1} \rho_j \dots \rho_{i-1} \mu_j^* \rho_i \dots$ are obviously equivalent, so we can change this part of $\boldsymbol{q}'$ and look for the next place where the queries differ.

Now assume that there is $\lambda_{j'} \neq \bot$ with $j' \in [j+1, i-1]$. Suppose there is no data instance in $(\mathfrak{p}_1)$, $(\mathfrak{p}_2)$ or $(\mathfrak{n}_1)$ separating $\boldsymbol{q}$ and $\boldsymbol{q}'$. Then $\mu_{j'} = \lambda_{j'}$. If $j' > j+1$, consider $\mathcal{D}^i_{j',1}$. Suppose $\mathcal{D}^i_{j',1} \models \boldsymbol{q}'$. Then there is $\mathcal{D}' = \rho_0 \dots \rho_{m-1} \mu_m \rho_m \dots \rho_{l-1} \mu_l \rho_l \dots \rho_n \Subset D^i_{j,k}$. By analysing all possible orderings of i, j, m, l we see that it is only possible when $\mu_{j'-1} \subseteq \lambda_j$. Then we have $\lambda_{j'-1} = \mu_{j'-1}$. After that, we consider $\mathcal{D}^i_{j'-1,1}$ and so on. We now have $\mu_{j+1} \subseteq \mu_{j+2} \subseteq \cdots \subseteq \lambda_{j'}$.

Consider $\mathcal{D}^i_{j+1,1}$. If $\mathcal{D}^i_{j+1,1} \models \boldsymbol{q}'$, then, in view of $\mu_j \not\subseteq \lambda_{j+1}$, we have $\rho_{j-1} \subseteq \lambda_{j+1}$.

Considering $\mathcal{D}^i_{j+1,2}$, we see that either $\rho_{j-2} \subseteq \lambda_{j+1}$ and $\mu_{j-1} \subseteq \rho_j$ or $\rho_{j-2} \subseteq \rho_j$ and $\mu_{j-1} \subseteq \lambda_{j+1}$ with μ_{j-1} and ρ_{j-2} incomparable, since otherwise we will violate the peerlessness of $\boldsymbol{q}$.

If $\rho_{j-2} \subseteq \lambda_{j+1}$ and $\mu_{j-1} \subseteq \rho_j$, then there is $j_1 < j-1$ such that $\mu_{j_1} \subseteq \rho_{j_1} \subseteq \ldots \subseteq \rho_{j-1} \subseteq \lambda_{j+1}$. This μ_{j_1} is paired with some λ_l for $l > j$. By peerlessness, we have $l = j$ and $\mu_j = \rho_{j_1} = \ldots = \rho_{j-1} = \lambda_j \subseteq \lambda_{j+1}$. It follows that $\lambda_{j-1} = \mu_{j-1}$ and we can repeat a previous argument and move further.

If $\rho_{j-2} \subseteq \rho_j$ and $\mu_{j-1} \subseteq \lambda_{j+1}$, then we have $\lambda_j = \rho_{j-1} = \mu_{j-1}$ and either $\mu_{j-2} \subseteq \rho_{j-1}$ or $\rho_{j-3} \subseteq \rho_{j-1}$.

Suppose $\mu_{j-2} \subseteq \rho_{j-1}$. Then $\rho_{j-3} \subseteq \rho_{j-2}$ and there is $j_2 \leq j-3$ such that $\mu_{j_2} \subseteq \rho_{j_2} \subseteq \ldots \subseteq \rho_{j-2}$. We know that μ_{j_2} subsumes some λ_l; by peerlessness it can only be λ_{j-1} with $\mu_{j_2} = \rho_{j_2} = \ldots = \rho_{j-2} = \lambda_{j-1}$, and $\rho_{j-2} \not\subseteq \mu_{j-2}$. So we repeat a previous argument and move further.

Suppose $\rho_{j-3} \subseteq \rho_{j-1}$. In this case we have either $\mu_{j-3} \subseteq \rho_{j-2}$ or $\rho_{j-4} \subseteq \rho_{j-2}$ and we move further.

Either way we cannot stop moving further and the process cannot terminate. Therefore, there is a data instance from $(\mathfrak{p}_1)$, $(\mathfrak{p}_2)$ or $(\mathfrak{n}_1)$ separating $\boldsymbol{q}$ and $\boldsymbol{q}'$.

This completes the proof of the theorem. ❑

D Proofs for Section 7

We start by giving the proof of Lemma 16.

Lemma 16. *For every $\boldsymbol{q} \in \mathcal{Q}[\Diamond]$ one can compute in polynomial time an equivalent query of the form $\boldsymbol{q}_1 \wedge \cdots \wedge \boldsymbol{q}_n$ with $\boldsymbol{q}_i \in \mathcal{Q}_p[\Diamond]$ for $i \leq n$.*

Proof. It is sufficient to observe that a query $\boldsymbol{q}$ of the form

$$\rho_0 \wedge \Diamond(\rho_1 \wedge \bigwedge_{i=1}^{n} \Diamond \boldsymbol{q}_i)$$

with $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n \in \mathcal{Q}[\Diamond]$ is equivalent to

$$\boldsymbol{q}' = \rho_0 \wedge \bigwedge_{i=1}^{n} \Diamond(\rho_1 \wedge \Diamond \boldsymbol{q}_i)$$

To see this, observe that $\boldsymbol{q} \models \boldsymbol{q}'$ is trivial. For the converse direction consider any data instance $\mathcal{D}$ with $\mathcal{D}, 0 \models \boldsymbol{q}'$. Then take the minimum ℓ_0 of all $\ell > 0$ such that $\mathcal{D}, \ell \models \rho_1 \wedge \Diamond \boldsymbol{q}_i$. We have $\mathcal{D}, \ell_0 \models \rho_1 \wedge \Diamond \boldsymbol{q}_i$ for all $i \leq n$ and so $\mathcal{D}, 0 \models \boldsymbol{q}$. ❑

Details for Example 18. Recall that

$$\boldsymbol{q}_1 = \emptyset(\boldsymbol{s}, \sigma)^n \boldsymbol{s}, \quad \boldsymbol{q}_2 = \emptyset \sigma^{2n+1}$$

where

$$\boldsymbol{s} = \{A_1, A_2\}\{B_1, B_2\}$$

and consider the set P of queries of the form

$$\emptyset \boldsymbol{s}_1 \cdots \boldsymbol{s}_{n+1}$$

with $\boldsymbol{s}_i$ either $\{A_1\}\{A_2\}$ or $\{B_1\}\{B_2\}$. P contains 2^{n+1} queries.

Lemma 35. *$\boldsymbol{q}_1 \wedge \boldsymbol{q}_2 \not\models \boldsymbol{q}$ for any $\boldsymbol{q} \in P$. For any data instance $\mathcal{D}$ with $\mathcal{D} \models \boldsymbol{q}_1 \wedge \boldsymbol{q}_2$ there is at most one $\boldsymbol{q} \in P$ with $\mathcal{D} \not\models \boldsymbol{q}$.*

Proof. We first construct for every $\boldsymbol{q} = \emptyset \boldsymbol{s}_1, \ldots \boldsymbol{s}_{n+1} \in P$ a data instance $\mathcal{D}$ with $\mathcal{D} \models \boldsymbol{q}_1 \wedge \boldsymbol{q}_2$ and $\mathcal{D} \not\models \boldsymbol{q}$. The data instance $\mathcal{D}_{\boldsymbol{q}}$ is defined by taking the data instance $\overline{\boldsymbol{q}}_1 = \rho_0, \ldots, \rho_{3n+2}$ and replacing ρ_i by σ in it if, for some $j \leq n$:

- $i = 3j+1$ and $\boldsymbol{s}_{j+1} = \{A_1\}\{A_2\}$; or
- $i = 3j+2$ and $\boldsymbol{s}_{j+1} = \{B_1\}\{B_2\}$.

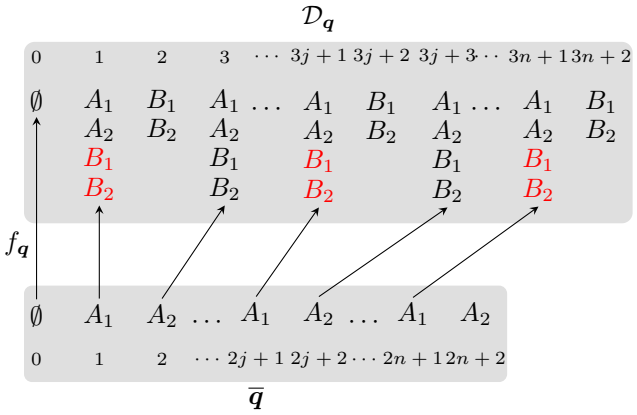


Figure 1: Definition of $\mathcal{D}_{\boldsymbol{q}}$ (additions to $\overline{\boldsymbol{q}}_1$ are indicated in red) and $f_{\boldsymbol{q}}$ for $\boldsymbol{q} = \emptyset(\{A_1\}\{A_2\})^{n+1}$.

The data instance $\mathcal{D}_{\boldsymbol{q}}$ with $\boldsymbol{q} = \emptyset(\{A_1\}\{A_2\})^{n+1}$ is depicted in Figure 1. It also illustrates that $\mathcal{D}_{\boldsymbol{q}} \not\models \boldsymbol{q}$.

We have $\mathcal{D}_{\boldsymbol{q}} \models \boldsymbol{q}_1 \wedge \boldsymbol{q}_2$. We next show that $\mathcal{D}_{\boldsymbol{q}} \not\models \boldsymbol{q}$ for all $\boldsymbol{q} \in P$. We introduce a method to prove this that regards the query $\boldsymbol{q}$ as a CQ. Namely, for $\boldsymbol{q} = \rho_0 \ldots \rho_n$ and $\mathcal{D} = \delta_0 \ldots \delta_m$ we have $\mathcal{D} \models \boldsymbol{q}$ iff the following inductive definition of a partial assignment f from $[0, n]$ to $[0, m]$ has domain $[0, n]$:

- $f(0) = 0$ if $\rho_0 \subseteq \delta_0$, otherwise $f(0)$ is undefined;
- if $f(i)$ is defined then let $f(i+1)$ be the minimal $j > f(i)$ such that $j \leq m$ and $\rho_{i+1} \subseteq \delta_j$, if such a j exists. Otherwise $f(i+1)$ is undefined.

If the domain of f does not equal $[0, n]$, then the definition either fails at $i_0 = 0$ or at some induction step $i_0 = i+1$. In either case we say that *the definition of a satisfying assignment fails at i_0*.

We return to the proof that $\mathcal{D}_{\boldsymbol{q}} \not\models \boldsymbol{q}$. We show that the definition of an assignment fails at $2n+2$. In fact, the partial assignment $f_{\boldsymbol{q}}$ in $\mathcal{D}_{\boldsymbol{q}}$ is defined as follows

- $f_{\boldsymbol{q}}(0) = 0$,
- $f_{\boldsymbol{q}}(2i+1) = 3i+1$ if $\boldsymbol{s}_{i+1} = \{A_1\}\{A_2\}$, for $0 \leq i \leq n$;
- $f_{\boldsymbol{q}}(2i+1) = 3i+2$ if $\boldsymbol{s}_{i+1} = \{B_1\}\{B_2\}$, for $0 \leq i \leq n$;
- $f_{\boldsymbol{q}}(2i+2) = 3i+3$, for $0 \leq i < n$,

and $f_{\boldsymbol{q}}(2n+2)$ remains undefined.

Next we show that there does not exist a database $\mathcal{D}$ with $\mathcal{D} \models \boldsymbol{q}_1 \wedge \boldsymbol{q}_2$ such that $\mathcal{D} \not\models \boldsymbol{q}$ and $\mathcal{D} \not\models \boldsymbol{q}'$ for distinct $\boldsymbol{q}, \boldsymbol{q}'$ in P. Assume for a proof by contradiction that $\mathcal{D} = \delta_0 \ldots \delta_m$ is such a data instance.

Let $\mathcal{D}_{\boldsymbol{q}} = \delta'_0 \ldots \delta'_{3n+2}$ and $\mathcal{D}_{\boldsymbol{q}'} = \delta''_0 \ldots \delta''_{3n+2}$ be the data instances defined above by extending $\rho_0 \ldots \rho_{3n+2}$ with σ depending on $\boldsymbol{q}$ and $\boldsymbol{q}'$, respectively.

As $\mathcal{D} \models \boldsymbol{q}_1$ there exists a bijective function g from some subset $A \subseteq [1, m]$ to $[1, 3n+2]$ satisfying the following

- $i < j$ iff $g(i) < g(j)$;
- $\delta_i \supseteq \rho_{g(i)}$ for all $i \in A$.

As $\mathcal{D} \models \boldsymbol{q}_2$ and $\boldsymbol{q} \neq \boldsymbol{q}'$,

- there exists $i \in [1,m] \setminus A$ such that $\delta_i = \sigma$; or
- there exists $i \in A$ such that $\delta_i = \sigma$ and $\delta'_{g(i)} \neq \sigma$; or
- there exists $i \in A$ such that $\delta_i = \sigma$ and $\delta''_{g(i)} \neq \sigma$.

We may thus assume w.l.o.g. that $\mathcal{D}$ is obtained from $\rho_0 \dots \rho_{3n+2}$ by (1) inserting σ after some ρ_i with $i > 0$ or (2) by replacing some ρ_i with $i > 0$ and $\delta'_i \neq \sigma$ by σ or (3) by replacing some ρ_i with $i > 0$ and $\delta''_i \neq \sigma$ by σ.

Case 2 and Case 3 are equivalent, so we only only consider Case 1 and Case 2.

Case 1. One can show that then $\mathcal{D} \models \boldsymbol{q}''$ for all $\boldsymbol{q}'' = \emptyset \boldsymbol{s}''_1 \dots \boldsymbol{s}''_{n+1} \in P$. We do this for $\boldsymbol{q}$.

Assume first that σ is inserted in $\rho_0 \dots \rho_m$ directly before timepoint 1. We define an assignment $f^*_{\boldsymbol{q}}$ by mapping 1 to the new node and (we assume that the original nodes are still numbered as before and not shifted):

- $f^*_{\boldsymbol{q}}(0) = 0$,
- $f^*_{\boldsymbol{q}}(2i+2) = 3i+1$ if $\boldsymbol{s}_{i+1} = \{A_1\}\{A_2\}$, for $0 \leq i \leq n$;
- $f^*_{\boldsymbol{q}}(2i+2) = 3i+2$ if $\boldsymbol{s}_{i+1} = \{B_1\}\{B_2\}$, for $0 \leq i \leq n$;
- $f^*_{\boldsymbol{q}}(2(i+1)+1) = 3i+3$, for $0 \leq i < n$.

Hence $\mathcal{D} \models \boldsymbol{q}$. If σ is inserted later, the argument is similar. In this case the assignment is defined by first taking $f_{\boldsymbol{q}}$ as defined above and then from the point where σ is inserted $f^*_{\boldsymbol{q}}$.

Case 2. This is proved similarly to Case 1. ❑

We next provide further details of the proof of Theorem 19. We first provide the missing proof of Lemma 22.

Lemma 22. *(i) For any $\mathcal{D} \in E^-_{\boldsymbol{q},m}$, we have $\mathcal{D} \not\models \boldsymbol{q}$.*

(ii) For any $\boldsymbol{q}' \in \mathcal{Q}^\sigma[\Diamond]$ with $\boldsymbol{q}' \not\models \boldsymbol{q}$ and tdp$(\boldsymbol{q}') \leq m$, there exists $\mathcal{D} \in E^-_{\boldsymbol{q},m}$ with $\mathcal{D} \models \boldsymbol{q}'$.

Proof. $\mathcal{D} \not\models \boldsymbol{q}$ for all $\mathcal{D} \in E^-_{\boldsymbol{q},m}$ holds by definition and Lemma 21.

Now assume $\boldsymbol{q}' \in \mathcal{Q}^\sigma[\Diamond]$ with $\boldsymbol{q}' \not\models \boldsymbol{q}$ and tdp$(\boldsymbol{q}') \leq m$. $\boldsymbol{q}'$ is equivalent to a conjunction $\boldsymbol{q}'_1 \wedge \dots \wedge \boldsymbol{q}'_k$ with $\boldsymbol{q}'_i \in \mathcal{Q}_p[\Diamond]$ and

$$\boldsymbol{q}'_i = \tau^i_0 \wedge \Diamond(\tau^i_1 \wedge \Diamond(\tau^i_2 \wedge \dots \wedge \Diamond\tau^i_{\ell_i}))$$

If there exists $A \in \rho$ with A not in any τ^i_0, then $\sigma^A\sigma^m \models \boldsymbol{q}'$, as required. Otherwise, $\boldsymbol{q}' \not\models \bigwedge_{i=1}^n \boldsymbol{q}^-_i$. Pick an i with $\boldsymbol{q}' \not\models \boldsymbol{q}^-_i$. Then $\boldsymbol{q}'_j \not\models \boldsymbol{q}^-_i$ for all j. Hence $\mathcal{D}_{\boldsymbol{q}^-_i,m} \models \boldsymbol{q}'_j$ for all j, and we obtain $\mathcal{D}_{\boldsymbol{q}^-_i,m} \models \boldsymbol{q}'$. ❑

We come to the construction of positive examples required for the proof of Part 1 of Theorem 19. For

$$\boldsymbol{q} = A_0 \wedge \Diamond(A_1 \wedge \Diamond(A_2 \wedge \dots \wedge \Diamond A_n))$$

we call any $\boldsymbol{q}_{|k} := A_1, \dots, A_k$ with $k \leq n$ a *prefix* of $\boldsymbol{q}$ (note that A_0 is not taken into account).

Lemma 36. *Let $w = A_1, \dots, A_k$ be a sequence of atomic concepts in a signature σ and assume $k < n$. Then one can construct in polynomial time a σ-data instance $\mathcal{D}_{w,n}$ such that for all simple $\boldsymbol{q} \in \mathcal{Q}^\sigma_p[\Diamond]$ with tdp$(\boldsymbol{q}) = n$: $\boldsymbol{q}_{|k} = w$ iff $\mathcal{D}_{w,n} \not\models \boldsymbol{q}$.*

Proof. Assume w and n are given. Then $\mathcal{D}_{w,n}$ defined as

$$\sigma(\sigma \setminus \{A_1\})^n\sigma(\sigma \setminus \{A_2\})^n \dots (\sigma \setminus \{A_k\})^n\sigma^{n-k-1}$$

is as required. ❑

Assume $\boldsymbol{q} = \boldsymbol{q}_1 \wedge \dots \wedge \boldsymbol{q}_m$ is balanced, simple, $n =$ tdp$(\boldsymbol{q})$, and

$$\boldsymbol{q}_i = A^i_0 \wedge \Diamond(A^i_1 \wedge \Diamond(A^i_2 \wedge \dots \wedge \Diamond A^i_n))$$

with $A_0 = A^1_0 = \dots = A^m_0$. Then let E^+ contain the σ-data instances

$$\{A_0\}\sigma^n, \quad \{A^1_0\} \dots \{A^1_n\} \dots \{A^m_0\} \dots \{A^m_n\},$$

and $\mathcal{D}_{w,n}$, for $w \in I$, where I is the set of all $A^i_1, \dots, A^i_k, A$ such that $A \in \sigma$, $1 \leq i \leq m$, and $A^i_1, \dots, A^i_k, A$ is not a prefix of any $\boldsymbol{q}_j$. Clearly $\mathcal{D} \models \boldsymbol{q}$ for all $\mathcal{D} \in E^+$.

Lemma 37. *(E^+, E^-) characterises $\boldsymbol{q}$ within the class of balanced queries in $\mathcal{Q}_b[\Diamond]$.*

Proof. First observe that

$$\{A^1_0\} \dots \{A^1_n\} \dots \{A^m_0\} \dots \{A^m_n\} \not\models \boldsymbol{q}'$$

for any $\boldsymbol{q}'$ that uses symbols not in σ or that is not simple. It remains to show that if $\boldsymbol{q} \not\models \boldsymbol{q}'$ and $\boldsymbol{q}' \models \boldsymbol{q}$, where $\boldsymbol{q}' = \boldsymbol{q}'_1 \wedge \dots \wedge \boldsymbol{q}_{m'}$ is simple, balanced, and uses symbols in σ only, then there exists $\mathcal{D} \in E^+$ with $\mathcal{D} \not\models \boldsymbol{q}'$. Recall that $n =$ tdp$(\boldsymbol{q})$. If tdp$(\boldsymbol{q}') > n$, then $\{A_0\}\sigma^n \not\models \boldsymbol{q}'$ and we are done. As $\boldsymbol{q}' \models \boldsymbol{q}$, we then have tdp$(\boldsymbol{q}') = n$. Then there exists $\boldsymbol{q}'_i$ with $\boldsymbol{q} \not\models \boldsymbol{q}'_i$. If $\boldsymbol{q}_i$ does not start with A_0, then $\{A_0\}\sigma^n \not\models \boldsymbol{q}'_i$ and we are done. Otherwise let k be maximal such that there exists $\boldsymbol{q}_j$ with $\boldsymbol{q}_{j\,|k} = \boldsymbol{q}'_{i\,|k}$. Then $k < n$ and $w = \boldsymbol{q}'_{i\,|k+1} \in I$. Hence $\mathcal{D}_{w,n} \not\models \boldsymbol{q}'_i$. ❑

We finally show that the positive examples defined in the main paper are as required for Part 2 of Theorem 19. Assume $\boldsymbol{q} = \boldsymbol{q}_1 \wedge \dots \wedge \boldsymbol{q}_m \in \mathcal{Q}^\sigma_b[\Diamond] \cap \mathcal{Q}^\sigma_{\leq n}[\Diamond]$ and that $\boldsymbol{q} \not\models \boldsymbol{q}'$ with $\boldsymbol{q}' \in \mathcal{Q}_b[\Diamond] \cap \mathcal{Q}_{\leq n}[\Diamond]$. Assume $\boldsymbol{q}' = \boldsymbol{q}'_1 \wedge \dots \wedge \boldsymbol{q}'_m$. If $\boldsymbol{q}'$ uses a symbol not in σ or tdp$(\boldsymbol{q}') > N$, then $\rho\sigma^N \not\models \boldsymbol{q}'$ and we are done. If the initial conjunct of some $\boldsymbol{q}'_i$ contains an $A \notin \rho$, then $\rho\sigma^N \not\models \boldsymbol{q}'$, and we are done. If tdp$(\boldsymbol{q}') < N$, then $\boldsymbol{q}' \not\models \boldsymbol{q}$, and so there exists $\mathcal{D} \in E^-$ with $\mathcal{D} \models \boldsymbol{q}'$. Hence we may assume that tdp$(\boldsymbol{q}') = N$. Take a j with $\boldsymbol{q} \not\models \boldsymbol{q}'_j$ and assume that

$$\boldsymbol{q}'_j = \tau_0 \wedge \Diamond(\tau_1 \wedge \Diamond(\tau_2 \wedge \dots \wedge \Diamond\tau_N))$$

Define $f\colon \{1, \dots, m\} \to \{1, \dots, N\}$ by taking for every i a ρ^i_j such that $\tau_j \not\subseteq \rho^i_j$ and setting $f(i) = j$. Such j exist since $\boldsymbol{q} \not\models \boldsymbol{q}'_j$ and since we excluded any other reason for non-entailment already. Then $\mathcal{D}_f \not\models \boldsymbol{q}'_j$, as required.

E Proofs for Section 8

We begin by introducing some notation. A *pointed* atemporal Σ-data instance is a pair $(\mathcal{A}, a)$ with $\mathcal{A}$ an atemporal Σ-data instance and $a \in \text{ind}(\mathcal{A})$. We associate with $\mathcal{A}$ the undirected graph

$$G_{\mathcal{A}} = (\text{ind}(\mathcal{A}), \bigcup_{P \in \sigma} \{\{a,b\} \mid P(a,b) \in \mathcal{A}\})$$

and call $\mathcal{A}$ *acyclic* if $G_{\mathcal{A}}$ is acyclic and $P(a,b) \in \mathcal{A}$ implies $Q(a,b) \notin \mathcal{A}$ for any $Q \neq P$ and $Q(b,a) \notin \mathcal{A}$ for any Q. $\mathcal{A}$ is *connected* if $G_{\mathcal{A}}$ is connected. We sometimes call acyclic and connected data instances *tree-shaped*. We assume the standard representation of an $\mathcal{ELI}$-query $\boldsymbol{q}$ as a set of atoms of the form $A(x)$, $P(x,y)$ with x, y variables and a *distinguished variable*, also called the *answer variable* of $\boldsymbol{q}$. For instance, $\boldsymbol{r} = B \wedge \exists P.\exists P^{-}.A$ is represented as $\{B(x), P(x,y), P(z,y), A(z)\}$ with the distinguished variable x.

Every $\mathcal{ELI}$-query $\boldsymbol{q}$ defines a pointed data instance $\hat{\boldsymbol{q}} = (\boldsymbol{q}, x)$ (with $ind(\boldsymbol{q})$ being the variables), where $\boldsymbol{q}$ is tree shaped. Conversely, every pointed database $(\mathcal{A}, a)$ with tree-shaped $\mathcal{A}$ defines an $\mathcal{ELI}$-query.

Let $\mathcal{A}$ and $\mathcal{B}$ be data instances. We call a mapping h from ind($\mathcal{A}$) to ind($\mathcal{B}$) a *homomorphism* from $\mathcal{A}$ to $\mathcal{B}$, in symbols $h : \mathcal{A} \to \mathcal{B}$, if

- $A(a) \in \mathcal{A}$ implies $A(h(a)) \in \mathcal{B}$;
- $P(a,b) \in \mathcal{A}$ implies $P(h(a), h(b)) \in \mathcal{B}$.

We call h a *homomorphism* from pointed $(\mathcal{A}, a)$ to pointed $(\mathcal{B}, b)$ if it is a homomorphism from $\mathcal{A}$ to $\mathcal{B}$ and $h(a) = b$. We write $(\mathcal{A}, a) \to (\mathcal{B}, b)$ if there exists a homomorphism from $(\mathcal{A}, a)$ to $(\mathcal{B}, b)$.

Lemma 38. *For all $\mathcal{ELI}$-queries $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$, we have $\boldsymbol{q}_1 \models \boldsymbol{q}_2$ iff $\hat{\boldsymbol{q}}_2 \to \hat{\boldsymbol{q}}_2$.*

We call a pointed data instance $(\mathcal{A}, a)$ *core* if every homomorphism $h : (\mathcal{A}, a) \to (\mathcal{A}, a)$ is an isomorphism. Pointed structures $(\mathcal{A}, a)$ and $(\mathcal{B}, b)$ are *homomorphically equivalent* if $(\mathcal{A}, a) \to (\mathcal{B}, b)$ and $(\mathcal{B}, b) \to (\mathcal{A}, a)$.

Theorem 39. *For every tree-shaped $(\mathcal{A}, a)$, one can construct in polynomial time a core that is tree-shaped and homomorphically equivalent to $(\mathcal{A}, a)$.*

We have defined frontiers within the set of $\mathcal{ELI}$-queries partially ordered by entailment. It is sometimes more convenient to define frontiers on the class of tree-shaped data instance partially ordered by homomorphisms. A set $\mathcal{F}$ of tree-shaped $(\mathcal{A}', a')$ is called a *frontier* of tree-shaped $(\mathcal{A}, a)$ if

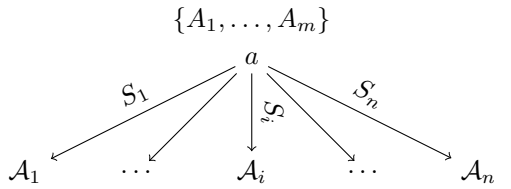
- $(\mathcal{A}', a') \to (\mathcal{A}, a)$ for all $(\mathcal{A}', a') \in \mathcal{F}$;
- $(\mathcal{A}, a) \not\to (\mathcal{A}', a')$ for all $(\mathcal{A}', a') \in \mathcal{F}$;
- if $(\mathcal{B}, b) \to (\mathcal{A}, a)$, then either $(\mathcal{A}, a) \to (\mathcal{B}, b)$ or there exists $(\mathcal{A}', a') \in \mathcal{F}$ with $(\mathcal{B}, b) \to (\mathcal{A}', a')$.

The frontier of $(\mathcal{A}, a)$ is denoted by $\mathcal{F}(\mathcal{A}, a)$.

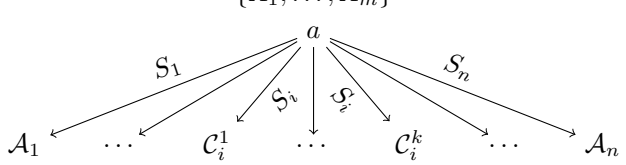
E.1 Proof of Theorem 23

We describe the construction of the frontier $\mathcal{F}(\mathcal{A}, a)$ for an tree-shaped pointed data instance $\mathcal{A}, a$. (This construction is a minor adaptation of the construction in (ten Cate and Dalmau 2021), where the proof of its correctness is given.) In this case, we can view $(\mathcal{A}, a)$ as a labelled directed tree with the set of nodes $ind(\mathcal{A})$ labelled with (possibly empty) sets of concept names $\{A_1, \ldots, A_n\}$, edges (b, c) labelled with either P or P^{-}, for a role name $P \in \Sigma$, and rooted at a. First, we construct *pre-frontier*, which is a set of structures $\mathcal{P}(\mathcal{A}, a)$ defined inductively as follows:

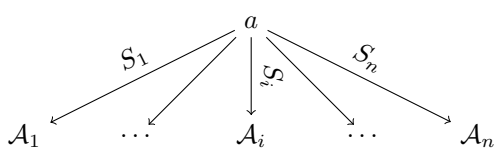
- if a is a leave of $\mathcal{A}$ and a has no concept name labels, then $\mathcal{P}(\mathcal{A}, a) = \emptyset$;
- if a is a leave of $\mathcal{A}$ labelled with $\{A_1, \ldots, A_m\}$, then $\mathcal{P}(\mathcal{A}, a) = \{(\mathcal{B}^1, a), \ldots, (\mathcal{B}^m, a)\}$, where $\mathcal{B}^i$ is
  $\{A_1, \ldots, A_{i-1}, A_{i+1}, A_m\}$
  a
- otherwise, let $\mathcal{A}$ be



where each $\mathcal{A}_i$ has a root a_i and let $\mathcal{P}(\mathcal{A}_i, a_i) = \{(\mathcal{B}_i^1, a_i), \ldots, (\mathcal{B}_i^k, a_i)\}$, for $k \geq 0$. We construct another set $\{(\mathcal{C}_i^j, c_i^j)\}$ of pointed structures, where the domain of $\mathcal{C}_i^j$ is equal to the set of sequences bj where b is in the domain of $\mathcal{B}_i^j$, and $S(bj, cj)$ is in $\mathcal{C}_i^j$ iff $S(b,c)$ is in $\mathcal{B}_i^j$ (and similarly for concept assertions $A(b)$). We set $c_i^j = a_i j$. We note that every element d in the domain of $\mathcal{B}_i^j$ or $\mathcal{C}_i^j$ has the form $aj_1 \ldots j_\ell$, for $\ell \geq 0$ and a from the domain of $\mathcal{A}$. We write $orig(d) = a$ to indicate that a is the *original* of d. Then, $\mathcal{P}(\mathcal{A}, a)$ is defined as $\{(\mathcal{D}_1, a), \ldots, (\mathcal{D}_n, a)\} \cup \{(\mathcal{E}_1, a), \ldots, (\mathcal{E}_m, a)\}$, where $\mathcal{D}_i$ is the structure:



and each subtree $\mathcal{C}_i^j$ is rooted at c_i^j, and $\mathcal{E}_i$ is



Finally, the fronteer $\mathcal{F}(\mathcal{A}, a)$ is $\{(\mathcal{F}_1, a), \ldots, (\mathcal{F}_n, a)\}$, where each $(\mathcal{F}_i, a)$ is obtained from $(\mathcal{D}_i, a) \in \mathcal{F}(\mathcal{A}, a)$ in the following way. Let the domain of $\mathcal{D}_i$ be $\{a, d_1, \ldots, d_k\}$. We construct a j-th copy $\mathcal{A}^j$ of $\mathcal{A}$, where domain of the structure $\mathcal{A}^j$ is the set of $b\ -j$, for b in the domain of $\mathcal{A}$ and $S(b\ -j, c\ -j)$ is in $\mathcal{A}^j$ iff $S(b,c)$ is in $\mathcal{A}$ (and similarly for concept assertions $A(b)$; note that we use negative numbers). Then, $\mathcal{F}_i$ is the union of $\mathcal{D}_i$, $\mathcal{A}^1$, ..., $\mathcal{A}^k$ (note that they are all pairwise disjoint), where we add $S(b\ -j, d_j)$ to $\mathcal{F}_i$ iff $S(b,c)$ is in $\mathcal{A}$ and $orig(d_j) = c$.

E.2 Proof of Theorem 25 (i)

Let $tdp(\boldsymbol{q})$ be the maximum number of nested temporal operators in $\boldsymbol{q}$ (assuming that no subquery of $\boldsymbol{q}$ starting with

a temporal operator is equivalent to $\top$). Let $rdp(\boldsymbol{q})$ be the length n of the longest sequence $\exists S_1 \dots \exists S_n$ such that $\exists S_{i+1}$ is in the scope of $\exists S_i$ but not in the scope of a temporal operator in the scope of $\exists S_i$. The set of $\mathcal{Q}[\bigcirc, \Diamond] \otimes \mathcal{EL}$-queries $\boldsymbol{q}$ with $tdp(q) \leq d$ and $rdp(\boldsymbol{q}) \leq r$, for some fixed $d, r < \omega$, contains finitely-many, say N_{dr}-many, non-equivalent queries. It is easy to construct data instances $\mathcal{D}^d$ and $\mathcal{D}^{dr}$ such that

– $\mathcal{D}^d, a, 0 \models \boldsymbol{q}$ iff $tdp(\boldsymbol{q}) \leq d$, for all $\boldsymbol{q}$;

– $\mathcal{D}^{dr}, a, 0 \models \boldsymbol{q}$ iff $rdp(\boldsymbol{q}) \leq r$, for all $\boldsymbol{q}$ with $tdp(\boldsymbol{q}) = d$.

It is also not hard to show that any pair of nonequivalent queries $\boldsymbol{q}, \boldsymbol{q}' \in \mathcal{Q}[\bigcirc, \Diamond] \otimes \mathcal{EL}$ with $tdp(\boldsymbol{q}) = tdp(\boldsymbol{q}')$ and $rdp(\boldsymbol{q}) = rdp(\boldsymbol{q}')$ is distinguished by a data instance $\mathcal{D}$ with $\max(\mathcal{D}) \leq N_{dr}$ and $|ind(\mathcal{D})| \leq N_{dr}$.

Now, given a $\boldsymbol{q}$ with $tdp(\boldsymbol{q}) = d$ and $rdp(\boldsymbol{q}) = r$, we construct an example set $(E, a, 0)$ with

$$E^+ = \{\mathcal{D} \mid \mathcal{D}, a, 0 \models \boldsymbol{q}, \max(\mathcal{D}) \leq N_{dr}, |ind(\mathcal{D})| \leq N_{dr}\},$$
$$E^- = \{\mathcal{D} \mid \mathcal{D}, a, 0 \not\models \boldsymbol{q}, \max(\mathcal{D}) \leq N_{dr}, |ind(\mathcal{D})| \leq N_{dr}\}.$$

It follows from the observations above that this example set uniquely characterises $\boldsymbol{q}$.

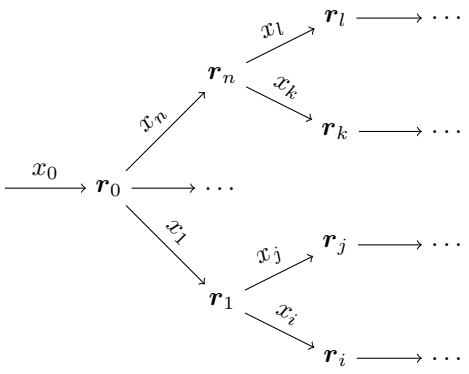
E.3 Proof of Theorem 25 (ii)

We observe that the queries $\boldsymbol{q} \in \mathcal{Q}[\bigcirc, \Diamond] \otimes \mathcal{ELI}$ can be viewed as the formulas $\boldsymbol{q}(x)$ of the monodic fragment of temporal FO (Hodkinson, Wolter, and Zakharyaschev 2000) (see also (Schild 1993)). In particular, the queries $\boldsymbol{q} \in \mathcal{Q}_p[\Diamond] \otimes \mathcal{ELI}$ naturally correspond to the formulas of the form:

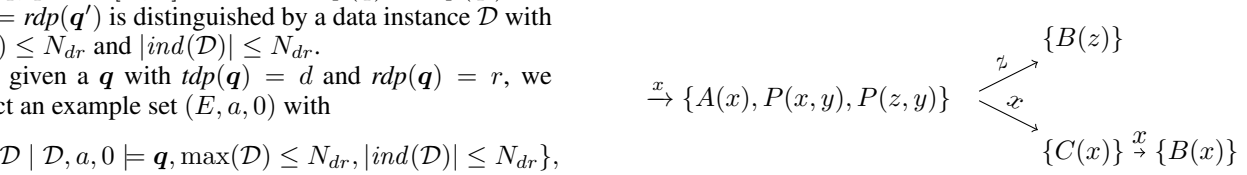
$$\boldsymbol{q}_0(x_0) = \exists x_1, \dots, x_s \Big(\bigwedge_{j \in I_1} A_j(x_{i_j}) \wedge \bigwedge_{k,\ell \in I_2} P_{k,\ell}(x_{i_k}, x_{i_\ell}) \wedge \bigwedge_{m \in I_3} \Diamond \boldsymbol{q}_m(x_{i_m}) \Big),$$

where $s \geq 0$, $0 \leq i_j, i_k, i_\ell, i_m \leq s$, $\boldsymbol{q}_m$ are of the same form as $\boldsymbol{q}_0$, and $x_{i_m} \neq x_{i_{m'}}$ for all $m, m' \in I_3$ with $m \neq m'$ (i.e., we do not allow more than one $\Diamond$-subformula for with the same free variable). Moreover, the set of atoms $\boldsymbol{r}_0 = \{A_j(x_{i_j} \mid j \in I_1\} \cup \{P_{k,\ell}(x_{i_k}, x_{i_\ell}) \mid k, \ell \in I_2\}$ is an $\mathcal{ELI}$-query with a distinguished variable x_0 (i.e., $\hat{\boldsymbol{r}}$ it is tree-shaped pointed data instance).

Thus, the $\boldsymbol{q} \in \mathcal{Q}_p[\Diamond] \otimes \mathcal{ELI}$ queries $\boldsymbol{q}$ can be represented as trees:



where the nodes are labelled with $\mathcal{ELI}$-queries $\boldsymbol{r}_i$ and the edges between the nodes are labelled with (query) variables, constructed as follows. The nodes of this tree, for a query $\boldsymbol{q}_0$, are from the set $\{\boldsymbol{q}_0\} \cup \{\boldsymbol{q}_i \mid \Diamond \boldsymbol{q}_i$ is a subformula of $\boldsymbol{q}_0\}$. The label of each $\boldsymbol{q}_i$ is the $\mathcal{ELI}$-query $\boldsymbol{r}_i$ constructed from the nontemporal atoms of $\boldsymbol{q}_i$; see $\boldsymbol{r}_0$ for $\boldsymbol{q}_0$ above. (We always omit the nodes on pictures as above; they are clear from the context.) The set of edges is defined as $\{(\boldsymbol{q}_i, \boldsymbol{q}_j) \mid \Diamond \boldsymbol{q}_j$ occurs in $\boldsymbol{q}_i$ not in scope of a $\Diamond\}$. The label of the edge $(\boldsymbol{q}_i, \boldsymbol{q}_j)$ is the (unique) free variable of $\boldsymbol{q}_j$. By the construction of $\boldsymbol{q}_i$, we note that there are no outgoing edges of a node with the same label. For example, the query $\boldsymbol{q} = \Diamond(C \wedge \Diamond B) \wedge A \wedge \exists P. \exists P^-. \Diamond B$ is represented as follows:



For every $\boldsymbol{q} \in \mathcal{Q}_p[\Diamond] \otimes \mathcal{ELI}$ (in what follows we do not distinguish between $\boldsymbol{q}$ and its tree representation) and every node $\boldsymbol{r}_i$ in it, we define $dep(\boldsymbol{r}_i)$ to be equal to the length of the path between the root $\boldsymbol{r}$ and $\boldsymbol{r}_i$. With such a $\boldsymbol{q}$, we associate the data instance $\mathcal{D}^{\boldsymbol{q}} = \mathcal{D}_0 \dots \mathcal{D}_n$ with $n = \max\{dep(\boldsymbol{r}_i) \mid \boldsymbol{r}_i$ in $\boldsymbol{q}\}$, $ind(\mathcal{D}^{\boldsymbol{q}})$ equal to the set of variables of $\boldsymbol{q}$, and $\mathcal{D}_i = \{A(x) \mid A(x) \in \boldsymbol{r}_i$ for some $\boldsymbol{r}_i$ in $\boldsymbol{q}\} \cup \{P(x,y) \mid P(x,y) \in \boldsymbol{r}_i$ for some $\boldsymbol{r}_i$ in $\boldsymbol{q}\}$. We also set $a^{\boldsymbol{q}} = x$, for the distinguished variable x of $\boldsymbol{q}$. On the other hand, we associate with $\boldsymbol{q}$ the atemporal pointed database $(\mathcal{A}^{\boldsymbol{q}}, (x, 0))$, where x is the distinguished variable of $\boldsymbol{q}$, with $ind(\mathcal{A}^{\boldsymbol{q}}) = \{(y, m) \mid y$ occurs in some $\boldsymbol{r}_i$ in $\boldsymbol{q}$ and $dep(\boldsymbol{r}_i) = m\}$, $A((y,m)) \in \mathcal{A}^{\boldsymbol{q}}$ iff $A(y) \in \boldsymbol{r}_i$ and $dep(\boldsymbol{r}_i) = m$, and $P((y,m),(z,m)) \in \mathcal{A}^{\boldsymbol{q}}$ iff $P(y,z) \in \boldsymbol{r}_i$ and $dep(\boldsymbol{r}_i) = m$. Moreover, for a fresh role name T, we add $T((y,m),(y,n))$ to $\mathcal{A}^{\boldsymbol{q}}$ for all $(y,m),(y,n) \in ind(\mathcal{A}^{\boldsymbol{q}})$, such that $n = m+1$. We observe that $\mathcal{A}^{\boldsymbol{q}}$ is tree-shaped. Also, every connected (taking into account T-atoms) substructure of $\mathcal{A}^{\boldsymbol{q}}$ induces a query $\boldsymbol{q}' \in \mathcal{Q}_p[\Diamond] \otimes \mathcal{ELI}$. Denote by $cl(\mathcal{A}^{\boldsymbol{q}})$ the structure that extends $\mathcal{A}^{\boldsymbol{q}}$ by adding the T-transitive closure.

Lemma 40. *Let $\boldsymbol{q}, \boldsymbol{q}' \in \mathcal{Q}_p[\Diamond] \otimes \mathcal{ELI}$ and assume that $\boldsymbol{q}$ (respectively, $\boldsymbol{q}'$) has a distinguished variable x (y). Then $\boldsymbol{q} \models \boldsymbol{q}'$ iff $\mathcal{D}^{\boldsymbol{q}}, a^{\boldsymbol{q}} \models \boldsymbol{q}'$ iff $(cl(\mathcal{A}^{\boldsymbol{q}}), x) \to (cl(\mathcal{A}^{\boldsymbol{q}'}), y)$.*

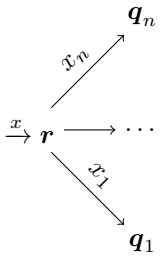
The following result shows the important property of the core of $\mathcal{A}^{\boldsymbol{q}}$ (note that (i) does not follow from Theorem 23):

Theorem 41. *Let $\boldsymbol{q} \in \mathcal{Q}_p[\Diamond] \otimes \mathcal{ELI}$. Then*

- *(i) a pointed substructure $(\mathcal{C}, (x,0))$ of $(cl(\mathcal{A}^{\boldsymbol{q}}), (x,0))$ that is a core is computable in polynomial time;*
- *(ii) there is a query $\boldsymbol{q}' \in \mathcal{Q}_p[\Diamond] \otimes \mathcal{ELI}$ such that $cl(\mathcal{A}^{\boldsymbol{q}'}) = \mathcal{C}$.*

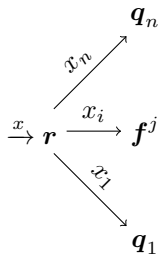
Frontier for 2D queries. We now define a *frontier* $\mathcal{F}(\boldsymbol{q}, \boldsymbol{r}, x)$ for a node $\boldsymbol{r}$ with a distinguished variable x in a query $\boldsymbol{q} \in \mathcal{Q}_p[\Diamond] \otimes \mathcal{ELI}$ by induction on the construction of (the tree of) $\boldsymbol{q}$:

- if $\boldsymbol{q}$ is $\xrightarrow{x} \boldsymbol{r}$ (i.e., an $\mathcal{ELI}$-query $\boldsymbol{r}$ with a distinguished variable x), then we set $\mathcal{F}(\boldsymbol{q}, \boldsymbol{r}, x)$ equal to $\mathcal{F}(\boldsymbol{r}, x)$, which takes the form $\{\xrightarrow{x} \boldsymbol{r}^1, \ldots, \xrightarrow{x} \boldsymbol{r}^k\}$.
- otherwise, let $\boldsymbol{q}$ be

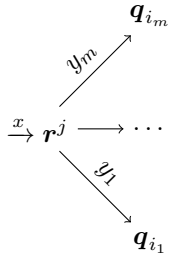


In this case, we set $\mathcal{F}(\boldsymbol{q}, \boldsymbol{r}, x)$ equal to $\boldsymbol{Q}_1 \cup \cdots \cup \boldsymbol{Q}_n \cup \boldsymbol{Q}'$, where

– assuming $\mathcal{F}(\boldsymbol{q}_i, \boldsymbol{r}_i, x_i) = \{\xrightarrow{x_i} \boldsymbol{f}^1, \ldots, \xrightarrow{x_i} \boldsymbol{f}^\ell\}$, we set $\boldsymbol{Q}_i = \{\xrightarrow{x} \boldsymbol{q}_i^1, \ldots, \xrightarrow{x} \boldsymbol{q}_i^\ell\}$ and $\boldsymbol{q}_i^j$:

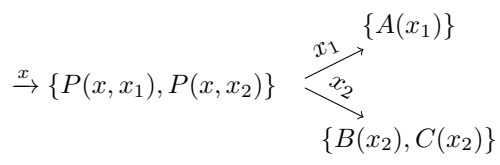


– assuming $\mathcal{F}(\boldsymbol{r}, x) = \{\xrightarrow{x} \boldsymbol{r}^1, \ldots, \xrightarrow{x} \boldsymbol{r}^k\}$, we set $\boldsymbol{Q}' = \{\xrightarrow{x} \boldsymbol{q}^1, \ldots, \xrightarrow{x} \boldsymbol{q}^k\}$ and $\boldsymbol{q}^j$:



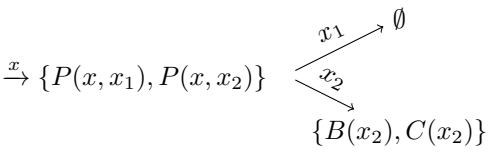
where $\{y_1, \ldots, y_m\}$ is the domain of $\boldsymbol{r}^j$ and $orig(y_\ell) = x_{i_\ell}$.

Example 42. Let $\boldsymbol{q} = \exists P.\Diamond A \wedge \exists P.\Diamond(B \wedge C)$, i.e.,

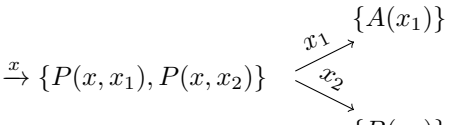


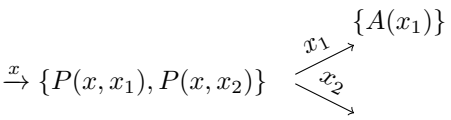
Clearly, $\boldsymbol{q}$ is core. Then, $\mathcal{F}(\boldsymbol{q}, \{A(x_1)\}, x_1) = \{(\emptyset, x_1)\}$ and $\mathcal{F}(\boldsymbol{q}, \{B(x_2), C(x_2)\}, x_2) = \{(\{B(x_2)\}, x_2), (\{C(x_2)\}, x_2)\}$. Therefore, $\mathcal{F}(\boldsymbol{q}, \{P(x, x_1), P(x, x_2)\}, x) = \boldsymbol{Q}_1 \cup \boldsymbol{Q}_2 \cup \boldsymbol{Q}'$, where:

- $\boldsymbol{Q}_1$ contains a single structure:

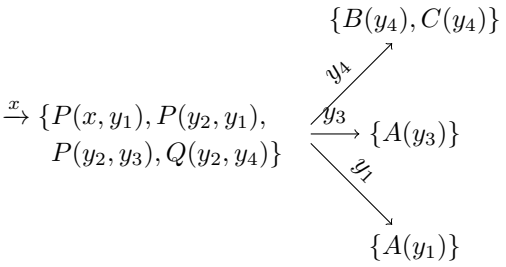


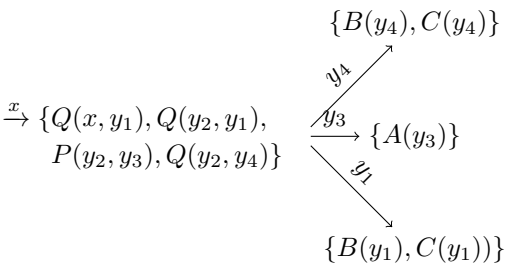
- $\boldsymbol{Q}_2$ contains two structures:





- $\boldsymbol{Q}'$ contains two structures:





We observe the following important property, which can be proved by induction on the construction of (the tree of) $\boldsymbol{q}$ using Lemma 40 and Theorem 41:

Theorem 43. *Let $\boldsymbol{q} \in \mathcal{Q}_p[\Diamond] \otimes \mathcal{ELI}$ be such that $cl(\mathcal{A}^{\boldsymbol{q}})$ is a core and $\boldsymbol{r}$ be the root of $\boldsymbol{q}$ with a distinguished variable x. Then (i) $\mathcal{F}(\boldsymbol{q}, \boldsymbol{r}, x) \subseteq \mathcal{Q}_p[\Diamond] \otimes \mathcal{ELI}$; (ii) $\mathcal{F}(\boldsymbol{q}, \boldsymbol{r}, x)$ is polynomial in the size of $\boldsymbol{q}$; (iii) the following properties hold:*

- *$\boldsymbol{q} \models \boldsymbol{q}'$ for every $\boldsymbol{q}' \in \mathcal{F}(\boldsymbol{q}, \boldsymbol{r}, x)$;*
- *$\boldsymbol{q}' \not\models \boldsymbol{q}$ for every $\boldsymbol{q}' \in \mathcal{F}(\boldsymbol{q}, \boldsymbol{r}, x)$;*
- *if $\boldsymbol{q} \models \boldsymbol{q}''$, then either $\boldsymbol{q}'' \models \boldsymbol{q}$ or there exits $\boldsymbol{q}' \in \mathcal{F}(\boldsymbol{q}, \boldsymbol{r}, x)$ such that $\boldsymbol{q}' \models \boldsymbol{q}''$, for any $\boldsymbol{q}'' \in \mathcal{Q}_p[\Diamond] \otimes \mathcal{ELI}$.*

Constructing examples. Given a $\boldsymbol{q} \in \mathcal{Q}_p[\Diamond] \otimes \mathcal{ELI}$, we take $\boldsymbol{q}'$ from Theorem 41 and set $E^+_{\boldsymbol{q}} = \{\mathcal{D}^{\boldsymbol{q}'}, a^{\boldsymbol{q}'}\}$. We take $E^-_{\boldsymbol{q}} = \{\mathcal{D}^{\boldsymbol{q}''}, a^{\boldsymbol{q}''} \mid \boldsymbol{q}'' \in \mathcal{F}(\boldsymbol{q}', \boldsymbol{r}, x')\}$, where $\boldsymbol{r}$ is the root of $\boldsymbol{q}'$ with distinguished variable x'. Finally, using

Lemma 40 and Theorems 41 and 43, we obtain the following:

Theorem 44. *$(E^+_{\boldsymbol{q}}, E^-_{\boldsymbol{q}})$ uniquely characterises $\boldsymbol{q}$, for any $\boldsymbol{q} \in \mathcal{Q}_p[\Diamond] \otimes \mathcal{ELI}$.*

For $\boldsymbol{q} \in \mathcal{Q}_p[\bigcirc] \otimes \mathcal{ELI}$ the construction of $\mathcal{F}(\boldsymbol{q}, \boldsymbol{r})$ is even easier. Indeed, it is sufficient to take $cl(\mathcal{A}^{\boldsymbol{q}}) = \mathcal{A}^{\boldsymbol{q}}$. This completes the proof of Theorem 25 (ii).

E.4 Proof of Theorem 26

For $n \geq 1$, $1 \leq i \leq n$, $j \leq i$, define $\mathcal{Q}_p[\mathcal{EL}/\bigcirc, \Diamond]$-queries $\boldsymbol{q}^n_{i,j}$ recursively by taking:

$$\boldsymbol{q}^n_{ii} = \boldsymbol{r}^n_i, \quad \boldsymbol{q}^n_{i,j-1} = B \wedge \bigcirc(B \wedge \Diamond(A \wedge \bigcirc \boldsymbol{q}^n_{i,j})),$$
$$\boldsymbol{r}^n_n = \Diamond(B \wedge \Diamond A), \;\; \boldsymbol{r}^n_l = \Diamond(B \wedge \Diamond(A \wedge \bigcirc \boldsymbol{s}_{n-l})), \;\; l < n,$$
$$\boldsymbol{s}_1 = B \wedge \bigcirc(B \wedge \Diamond A), \;\; \boldsymbol{s}_{i+1} = B \wedge \bigcirc(B \wedge \Diamond(A \wedge \bigcirc \boldsymbol{q}_i))$$

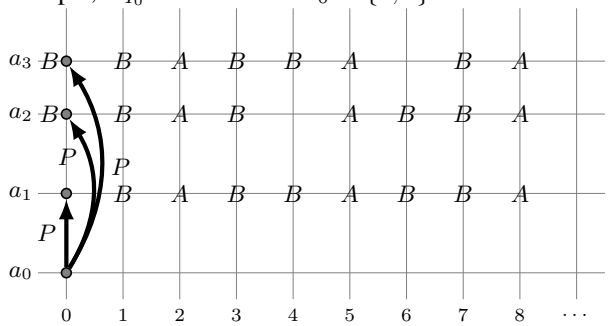
and set $\boldsymbol{q}^n_i = \boldsymbol{q}^n_{i,1}$. Minimal models of $\boldsymbol{q}^n_i$ look as follows:

$$\underbrace{BB\emptyset^*A}_{1} \dots \underbrace{BB\emptyset^*A}_{i-1} \underbrace{\emptyset^*B\emptyset^*A}_{i} \underbrace{BB\emptyset^*A}_{i+1} \dots \underbrace{BB\emptyset^*A}_{n}$$

Consider the $\mathcal{Q}_p[\mathcal{EL}/\bigcirc, \Diamond]$-query $\boldsymbol{q} = \exists P.\boldsymbol{q}^n_1 \wedge \dots \wedge \exists P.\boldsymbol{q}^n_n$. We claim that any unique characterisation of $\boldsymbol{q}$ contains at least 2^n positive examples (in E^+). Indeed, let $\mathcal{Q}$ be the set of all queries of the form $\boldsymbol{q} \wedge \exists P.\boldsymbol{s}$ with

$$\boldsymbol{s} = \boldsymbol{o}_1(B \wedge \Diamond(A \wedge \boldsymbol{o}_2(B \wedge \Diamond(A \wedge \dots \wedge \boldsymbol{o}_n(B \wedge \Diamond A) \dots)))),$$

where each $\boldsymbol{o}_i$ is either $\bigcirc$ or $\Diamond\bigcirc$ if $i > 1$, and either blank or $\Diamond$ if $i = 1$. Observe that $|\mathcal{Q}| = 2^n$ and $\boldsymbol{q}' \models \boldsymbol{q}$ for each $\boldsymbol{q}' \in \mathcal{Q}$. On the other hand, $\boldsymbol{q} \not\models \boldsymbol{q}'$. Indeed, let I_0 be the set of indices i such that $\boldsymbol{o}_i = \bigcirc$ and $i > 1$ or $\boldsymbol{o}_i$ is blank and $i = 1$. Let $I_1 = [1, n] \setminus I_0$. Define a data instance $\mathcal{D}_{I_0}$ with $ind(\mathcal{D}_{I_0}) = \{a_0, \dots, a_n\}$ and $\max(\mathcal{D}_{\mathcal{I}_0}) = 3n - 1$ as follows. To begin with, $\mathcal{D}_{I_0}$ contains $P(a_0, a_1, 0), \dots, P(a_0, a_n, 0)$. For $i \in [1, n]$ and $k \in [0, 3n-1]$, $\mathcal{D}_{I_0}$ contains $A(a_i, k)$ if $k \equiv 2 \,(\mathrm{mod}\, 3)$, and $B(a_i, k)$ if $k \notin [3(i-1), 3i)$ and $k \not\equiv 2 \,(\mathrm{mod}\, 3)$. Also, for $i \in I_0$ and $k \in [3(i-1), 3i)$, $\mathcal{D}_{I_0}$ contains $B(a_i, k)$ if $k \equiv 1 \,(\mathrm{mod}\, 3)$. Finally, for each $i \in I_1$ and $k \in [3(i-1), 3i)$, it has $B(a_i, k)$ if $k \equiv 0 \,(\mathrm{mod}\, 3)$. For example, $\mathcal{D}_{I_0}$ for $n = 3$ and $I_0 = \{1, 3\}$ is shown below:



Then $\mathcal{D}_{I_0}, a_0, 0 \models \boldsymbol{q}$ but $\mathcal{D}_{I_0}, a_0, 0 \not\models \boldsymbol{q}'$. To illustrate, take $\boldsymbol{s} = B \wedge \Diamond(A \wedge \wedge \Diamond\bigcirc(B \wedge \Diamond(A \wedge \bigcirc(B \wedge \Diamond A))))$, for which $\boldsymbol{q}' = \boldsymbol{q} \wedge \exists P.\boldsymbol{s} \in \mathcal{Q}$ with $I_0 = \{1, 3\}$. One can readily see that $\boldsymbol{q}'$ cannot be satisfied at $a_0, 0$ in $\mathcal{D}_{I_0}$ depicted above.

To complete the proof that $|E^+| \geq 2^n$, we prove the following property (cf. Example 17):

$$\mathcal{D} \models \boldsymbol{q} \;\; \& \;\; \mathcal{D} \not\models \boldsymbol{q}' \;\; \& \;\; (\boldsymbol{q}' \neq \boldsymbol{q}'') \;\; \Rightarrow \;\; \mathcal{D} \models \boldsymbol{q}'',$$

for all data instances $\mathcal{D}$ and $\boldsymbol{q}', \boldsymbol{q}'' \in \mathcal{Q}$. Indeed, take an arbitrary $\boldsymbol{q}$ such that $\mathcal{D} \models \boldsymbol{q}$ and let $\{a_0, \dots a_n\} \subseteq ind(\mathcal{D})$ be the elements such that $P(a_0, a_i, 0)$ is in $\mathcal{D}$ and $\mathcal{D}, a_i, 0 \models \boldsymbol{q}^n_i$, for each $1 \leq i \leq n$. Let $(\boldsymbol{K}_{i,1}, \dots, \boldsymbol{K}_{i,n})$ be a vector of pairs/triples over $\mathbb{N}$ such that, for $j \neq i$, $\boldsymbol{K}_{i,j} = (b, b', a)$ and $\boldsymbol{K}_{i,i} = (b, a)$, where:

- $b < b' < a$, $b < a$ in every $\boldsymbol{K}_{i,j}$;
- b from $\boldsymbol{K}_{i,j}$ is strictly greater than a from $\boldsymbol{K}_{i,j-1}$;
- a from $\boldsymbol{K}_{i,j}$ is strictly smaller than b from $\boldsymbol{K}_{i,j+1}$;
- $\mathcal{D}, a_i, b \models B$, $\mathcal{D}, a_i, b' \models B$, $\mathcal{D}, a_i, a \models A$ for all $(b, b', a) \in \boldsymbol{K}_{i,j}$;
- $\mathcal{D}, a_i, b \models B, \mathcal{D}, a_i, a \models A$ for all $(b, a) \in \boldsymbol{K}_{i,i}$.

Note that, for each $1 \leq i \leq n$, a vector $(\boldsymbol{K}_{i,1}, \dots, \boldsymbol{K}_{i,n})$ as above exists. Take $\boldsymbol{q}' \in \mathcal{Q}$ such that $\mathcal{D} \not\models \boldsymbol{q}'$ and let I_0 be the set corresponding to $\boldsymbol{q}'$. We observe that $\mathcal{D}, a_i, 0 \not\models \boldsymbol{s}$, for all $1 \leq i \leq n$. Therefore, for any $i \in I_0$ and any vector $(\boldsymbol{K}_{i,1}, \dots, \boldsymbol{K}_{i,n})$, we have that $b > a+1$ for a from $\boldsymbol{K}_{i,i-1}$ and b from $\boldsymbol{K}_{i,i}$, if $i > 0$, and $b > 0$ for b from $\boldsymbol{K}_{i,i}$, if $i = 1$. Similarly, for any $i \in [1, n] \setminus I_0$ and any vector $(\boldsymbol{K}_{i,1}, \dots, \boldsymbol{K}_{i,n})$, we have that $b = a+1$ for a from $\boldsymbol{K}_{i,i-1}$ and b from $\boldsymbol{K}_{i,i}$, if $i > 0$, and $b = 0$ for b from $\boldsymbol{K}_{i,i}$, if $i = 1$. Now, take an arbitrary $\boldsymbol{q}'' = \boldsymbol{q} \wedge \exists P.\boldsymbol{s}'' \in \mathcal{Q}$ such that $\boldsymbol{q}'' \neq \boldsymbol{q}'$ and let J_0 be the set corresponding to it. Clearly, $J_0 \neq I_0$. Suppose first that there exists $i \in J_0 \setminus I_0$. Therefore, $i \in [1, n] \setminus I_0$ and then, as it is easy to see, $\mathcal{D}, a_i, 0 \models \boldsymbol{s}''$. Thus, $\mathcal{D}, a_0, 0 \models \boldsymbol{q}''$ as was required. Now, if there is $i \in I_0 \setminus J_0$, the proof is analogous and left to the reader.

E.5 Proof of Theorem 27

Recall that we consider the class $\mathcal{Q}_p[\bigcirc, \Diamond_r](\mathcal{ELI})$ of queries of the form

$$\boldsymbol{q} = \boldsymbol{r}_0 \wedge \boldsymbol{o}_1(\boldsymbol{r}_1 \wedge \boldsymbol{o}_2(\boldsymbol{r}_2 \wedge \dots \wedge \boldsymbol{o}_n \boldsymbol{r}_n)), \qquad (11)$$

where the $\boldsymbol{r}_i$ are $\mathcal{ELI}$-queries and $\boldsymbol{o}_i \in \{\bigcirc, \Diamond, \Diamond_r\}$. We first generalise the normal form introduced for $\mathcal{Q}_p[\Diamond_r, \bigcirc]$. Any $\boldsymbol{q}$ in $\mathcal{Q}_p[\bigcirc, \Diamond_r](\mathcal{ELI})$ can be represented as a sequence

$$\boldsymbol{r}_0(t_0), R_1(t_0, t_1), \dots, \boldsymbol{r}_{n-1}(t_{n-1}), R_n(t_{n-1}, t_n), \boldsymbol{r}_n(t_n),$$

where $R_i \in \{suc, <, \leq\}$ and $\boldsymbol{r}_i$ is an $\mathcal{ELI}$-query. As before, we divide $\boldsymbol{q}$ into *blocks* $\boldsymbol{q}_i$ such that

$$\boldsymbol{q} = \boldsymbol{q}_0 \mathcal{R}_1 \boldsymbol{q}_1 \dots \mathcal{R}_n \boldsymbol{q}_n \qquad (12)$$

with $\mathcal{R}_i = R^i_1(t^i_0, t^i_1) \dots R^i_{n_i}(t^i_{n_i-1}, t^i_{n_i})$, for $R^i_j \in \{<, \leq\}$,

$$\boldsymbol{q}_i = \boldsymbol{r}^i_0(s^i_0) suc(s^i_0, s^i_1) \boldsymbol{r}^i_1(s^i_1) \dots suc(s^i_{k_i-1}, s^i_{k_i}) \boldsymbol{r}^i_{k_i}(s^i_{k_i})$$

and $s^i_{k_i} = t^{i+1}_0$, $t^i_{n_i} = s^i_0$. If $k_i = 0$, the block $\boldsymbol{q}_i$ is *primitive*. A primitive block $\boldsymbol{q}_i = \boldsymbol{r}^i_0(s^i_0)$ with $i > 0$ such that $\boldsymbol{r}^i_0$ is not equivalent to a conjunction of $\mathcal{ELI}$-queries that are not equivalent to $\boldsymbol{r}^i_0$ is called a *lone conjunct*. Now, we say that $\boldsymbol{q}$ is in *normal form* if the following conditions hold:

(n1′) $\boldsymbol{r}^i_0 \not\equiv \top$ if $i > 0$, and $\boldsymbol{r}^i_{k_i} \not\equiv \top$ if $i > 0$ or $k_i > 0$ (thus, of all the first/last $\boldsymbol{r}$ in a block only $\boldsymbol{r}^0_0$ can be trivial);

(n2′) each $\mathcal{R}_i$ is either a single $t^i_0 \leq t^i_1$ or a sequence of $<$;

(n3′) $\boldsymbol{r}^i_{k_i} \not\models \boldsymbol{r}^{i+1}_0$ if $\boldsymbol{q}_{i+1}$ is primitive and R_{i+1} is $\leq$;

(n4′) $\boldsymbol{r}^{i+1}_0 \not\models \boldsymbol{r}^i_{k_i}$ if $i > 0$, $\boldsymbol{q}_i$ is primitive and R_{i+1} is $\leq$.

Lemma 45. *Every query in $\mathcal{Q}_p[\bigcirc, \Diamond_r](\mathcal{ELI})$ is equivalent to a query in normal form that can be computed in polynomial time.*

We call a query in $\mathcal{Q}_p[\bigcirc, \Diamond_r](\mathcal{ELI})$ *safe* if it is equivalent to a query in normal form in $\mathcal{Q}_p[\bigcirc, \Diamond_r](\mathcal{ELI})$ without lone conjuncts.

Theorem 27. *(i) A query $\boldsymbol{q} \in \mathcal{Q}_p[\bigcirc, \Diamond_r](\mathcal{ELI})$ is uniquely characterisable within $\mathcal{Q}_p[\bigcirc, \Diamond_r](\mathcal{ELI})$ iff $\boldsymbol{q}$ is safe.*

(ii) Those queries that are uniquely characterisable within $\mathcal{Q}_p[\bigcirc, \Diamond_r](\mathcal{ELI})$ are actually polynomially characterisable within $\mathcal{Q}_p[\bigcirc, \Diamond_r](\mathcal{ELI})$.

(iii) The class $\mathcal{Q}_p[\bigcirc, \Diamond_r](\mathcal{ELI})$ is polynomially characterisable for bounded query size.

(iv) The class $\mathcal{Q}_p[\bigcirc, \Diamond](\mathcal{ELI})$ is polynomially characterisable.

We show how the construction of positive and negative examples provided in the proof of Theorem 8 can be generalised.

Suppose $\boldsymbol{q}$ in normal form (12) does not contain lone conjuncts. Let b be again the number of $\bigcirc$ and $\Diamond$ in $\boldsymbol{q}$ plus 1. We construct $E = (E^+, E^-)$ characterising $\boldsymbol{q}$ as follows. Pick an individual name a and let, for every $\mathcal{ELI}$-query $\boldsymbol{r}$ in $\boldsymbol{q}$, $\hat{\boldsymbol{r}}$ denote the pointed tree-shaped data instance defined by $\boldsymbol{r}$ (note that we take the same individual a for every $\boldsymbol{r}$).

For each block $\boldsymbol{q}_i$ in (12), we take two temporal data instances

$$\bar{\boldsymbol{q}}_i = \hat{\boldsymbol{r}}^i_0, \dots \hat{\boldsymbol{r}}^i_{k_i}$$
$$\bar{\boldsymbol{q}}_i \bowtie \bar{\boldsymbol{q}}_{i+1} = \hat{\boldsymbol{r}}^i_0, \dots \hat{\boldsymbol{r}}^i_{k_i} \cup \hat{\boldsymbol{r}}^{i+1}_0, \dots \hat{\boldsymbol{r}}^{i+1}_{k_{i+1}}.$$

The set E^+ contains the data instances given by

- $\mathcal{D}_b = \bar{\boldsymbol{q}}_0 \emptyset^b \dots \bar{\boldsymbol{q}}_i \emptyset^b \bar{\boldsymbol{q}}_{i+1} \dots \emptyset^b \bar{\boldsymbol{q}}_n$,
- $\mathcal{D}_i = \bar{\boldsymbol{q}}_0 \emptyset^b \dots \bar{\boldsymbol{q}}_i \bowtie \bar{\boldsymbol{q}}_{i+1} \dots \emptyset^b \bar{\boldsymbol{q}}_n$ if $\mathcal{R}_{i+1}$ is $\leq$,
- $\mathcal{D}_i = \bar{\boldsymbol{q}}_0 \emptyset^b \dots \bar{\boldsymbol{q}}_i \emptyset^{n_{i+1}} \bar{\boldsymbol{q}}_{i+1} \dots \emptyset^b \bar{\boldsymbol{q}}_n$ otherwise.

The set E^- contains all data instances of the form

- $\mathcal{D}^-_i = \bar{\boldsymbol{q}}_0 \emptyset^b \dots \bar{\boldsymbol{q}}_i \emptyset^{n_{i+1}-1} \bar{\boldsymbol{q}}_{i+1} \dots \emptyset^b \bar{\boldsymbol{q}}_n$ if $n_{i+1} > 1$;
- $\mathcal{D}^-_i = \bar{\boldsymbol{q}}_0 \emptyset^b \dots \bar{\boldsymbol{q}}_i \bowtie \bar{\boldsymbol{q}}_{i+1} \dots \emptyset^b \bar{\boldsymbol{q}}_n$ if $\mathcal{R}_{i+1}$ is a single $<$,

and also the data instances obtained from $\mathcal{D}_b$ by

(a) replacing $\hat{\boldsymbol{r}}^i_j \neq \emptyset$ by an element of $\mathcal{F}(\hat{\boldsymbol{r}}^i_j)$ or removing the whole $\hat{\boldsymbol{r}}^i_j = \emptyset$, for $i \neq 0$ and $j \neq 0$, from some $\bar{\boldsymbol{q}}_i$;

(b) replacing $\bar{\boldsymbol{q}}_i = \hat{\boldsymbol{r}}^i_0, \dots \hat{\boldsymbol{r}}^i_l, \hat{\boldsymbol{r}}^i_{l+1} \dots \hat{\boldsymbol{r}}^i_{k_i}$ ($k_i > 0$) by $\bar{\boldsymbol{q}}'_i \emptyset^b \bar{\boldsymbol{q}}''_i$, where $\bar{\boldsymbol{q}}'_i = \hat{\boldsymbol{r}}^i_0 \dots \hat{\boldsymbol{r}}^i_l$, $\bar{\boldsymbol{q}}''_i = \hat{\boldsymbol{r}}^i_{l+1} \dots \hat{\boldsymbol{r}}^i_{k_i}$ and $l \geq 0$;

(c) replacing some $\hat{\boldsymbol{r}}^i_l \neq \emptyset$, $0 < l < k_i$, by $\hat{\boldsymbol{r}}^i_l \emptyset^b \hat{\boldsymbol{r}}^i_l$;

(d) replacing $\hat{\boldsymbol{r}}^i_{k_i}$ ($k_i > 0$) with $\mathcal{A} \emptyset^b \hat{\boldsymbol{r}}^i_{k_i}$, for some $\mathcal{A} \in \mathcal{F}(\hat{\boldsymbol{r}}^i_{k_i})$, or replacing $\hat{\boldsymbol{r}}^i_0$ ($k_i > 0$) with $\hat{\boldsymbol{r}}^i_0 \emptyset^b \mathcal{A}$, for some $\mathcal{A} \in \mathcal{F}(\hat{\boldsymbol{r}}^i_0)$;

(e) replacing $\hat{\boldsymbol{r}}^0_0 \neq \emptyset$ with $\mathcal{A} \emptyset^b \hat{\boldsymbol{r}}^0_0$, for $\mathcal{A} \in \mathcal{F}(\hat{\boldsymbol{r}}^0_0)$, if $k_0 = 0$, and with $\hat{\boldsymbol{r}}^0_0 \emptyset^b \hat{\boldsymbol{r}}^0_0$ if $k_0 > 0$.

Let $\boldsymbol{q}$ be of the form (12). We generalise the notion of a homomorphism as follows. Let $\mathcal{D} = \mathcal{A}_0, \dots, \mathcal{A}_n$ and $a \in ind(\mathcal{D})$. A mapping h from the set $var(\boldsymbol{q})$ of variables in $\boldsymbol{q}$ to $[0, \max(\mathcal{D})]$ is a *generalised homomorphism* from $\boldsymbol{q}$ to $\mathcal{D}$ for a if $h(t_0) = 0$, $\mathcal{A}_{h(t)} \models \boldsymbol{r}(a)$ if $\boldsymbol{r}(t)$, $h(t') = h(t) + 1$ if $suc(t, t') \in \boldsymbol{q}$, and $h(t)\, R\, h(t')$ if $R(t, t') \in \boldsymbol{q}$ for $R \in \{<, \leq\}$. Then one can show that $\mathcal{D}, a, 0 \models \boldsymbol{q}$ if there exists a generalised homomorphism from $\boldsymbol{q}$ to $\mathcal{D}$ for a.

It is now almost trivial to extend the proof of Theorem 8 to a proof of Theorem 27 by replacing homomorphisms by generalised homomorphisms. For example, block surjectivity is generalised in a straightforward way as follows: a generalised homomorphism $h \colon \boldsymbol{q}' \to D_b$ is *block surjective* if every point in every block $\bar{\boldsymbol{q}}_i$ of $\mathcal{D}_b$ is in the range $ran(h)$ of h. To define type surjectivity let, for $\ell \in ran(h)$, $\boldsymbol{r}_\ell$ denote the conjunction of all $\mathcal{ELI}$-queries $\boldsymbol{r}'$ with $\boldsymbol{r}'(t)$ in $\boldsymbol{q}'$ such that $h(t) = \ell$. Clearly $\mathcal{A}_\ell \models \boldsymbol{r}_\ell(a)$. h is *type surjective* if $\mathcal{A} \not\models \boldsymbol{r}_\ell(a)$, for every $(\mathcal{A}, a) \in \mathcal{F}(\hat{\boldsymbol{r}})$.

Theorem 30. *Let $n > 0$ be fixed. For every set Q of $\mathcal{EL}(\Sigma)$-queries with $|Q| \leq n$, one can compute in polynomial time a split partner $\mathcal{S}(Q)$ of Q in $\mathcal{EL}(\Sigma)$. For $\mathcal{ELI}$, one can compute a split partner in exponential time.*

Proof. We prove the statement for $n = 1$, the generalisation is straightforward. Let $Q = \{\boldsymbol{q}\}$. The construction is by induction over the depth of $\boldsymbol{q}$. Assume $\mathrm{depth}(\boldsymbol{q}) = 0$. Thus $\boldsymbol{q} = \bigwedge_{i=1}^k A_i$ with A_i atomic concepts. Then let for $i \leq k$:

$$\begin{aligned} \mathcal{A}_{A_i} \quad = \quad & \{B(a) \mid B \in \Sigma \setminus \{A\}\} \cup \\ & \{R(a,b), R(b,b) \mid R \in \Sigma\} \cup \\ & \{B(b) \mid B \in \Sigma\} \end{aligned}$$

and set $\mathcal{S}(\boldsymbol{q}) = \{(\mathcal{A}_{A_i}, a) \mid 1 \leq i \leq k\}$. We show that $\mathcal{S}(\boldsymbol{q})$ is as required. Assume

$$\boldsymbol{q}' = \bigwedge_{i=1}^{m_1} B_i \wedge \bigwedge_{i=1}^{m_2} \exists R_i.\boldsymbol{q}_i$$

If $\boldsymbol{q}' \not\models \boldsymbol{q}$, then there exist A_i with $A_i \notin \{B_i \mid 1 \leq i \leq m_1\}$. Then $\mathcal{A}_{A_i} \models \boldsymbol{q}'(a)$, as required. Conversely, if $\mathcal{A}_{A_i} \models \boldsymbol{q}'(a)$ for some A_i, then $A_i \notin \{B_i \mid 1 \leq i \leq m_1\}$. Hence $\boldsymbol{q}' \not\models \boldsymbol{q}$, as required.

Assume now that $\mathrm{depth}(\boldsymbol{q}) = n + 1$ and that split partners $\mathcal{S}(\boldsymbol{q}')$ have been defined for queries $\boldsymbol{q}'$ of depth $\leq n$. Assume

$$\boldsymbol{q} = \bigwedge_{i=1}^{n_1} A_i \wedge \bigwedge_{i=1}^{n_2} \exists S_i.\boldsymbol{q}_i.$$

Then assume $\mathcal{S}(\boldsymbol{q}_i) = \{(\mathcal{A}_1, a_1), \dots, (\mathcal{A}_{k_i}, a_{k_i})\}$ and let $c_1, \dots, c_{k_i}$ be fresh individuals. Define for $i \leq n_2$ the data instance

$$\begin{aligned} \mathcal{A}_i \quad = \quad & \{B(a) \mid B \in \Sigma\} \cup \\ & \{R(a,b), S(b,b), B(b) \mid R \in \Sigma \setminus \{S_i\}, B, S \in \Sigma\} \cup \\ & \{S_i(a, c_j) \mid 1 \leq j \leq k_i\} \cup \\ & \mathcal{A}_1(c_1/a_1) \cup \dots \cup \mathcal{A}_{k_i}(c_{k_i}/a_{k_i}) \end{aligned}$$

with $\mathcal{A}(c/a)$ the result of replacing a by c in $\mathcal{A}$. Let $\mathcal{S}(\boldsymbol{q})$ be the union of $\mathcal{S}(\bigwedge_{i=1}^{n_1} A_i)$ and $\{(\mathcal{A}_i, a) \mid 1 \leq i \leq n_2\}$.

We show that $\mathcal{S}(\boldsymbol{q})$ is as required. Assume

$$\boldsymbol{q}' = \bigwedge_{i=1}^{m_1} B_i \wedge \bigwedge_{i=1}^{m_2} \exists R_i.\boldsymbol{q}'_i$$

If $\boldsymbol{q}' \not\models \boldsymbol{q}$, then either there exists A_i with $A_i \notin \{B_i \mid 1 \leq i \leq m_1\}$ or there exists $\exists S_i.\boldsymbol{q}_i$ such that for every $R_j.\boldsymbol{q}_j$ with $S_i = R_j$, we have $\boldsymbol{q}'_j \not\models \boldsymbol{q}_i$. In the former case $\mathcal{A}_{A_i} \models \boldsymbol{q}'(a)$ and in the latter case, by induction hypothesis, $\mathcal{A}_i \models \boldsymbol{q}'(a)$. Conversely, if $\mathcal{A} \not\models \boldsymbol{q}'(a)$ for some $\mathcal{A}, a \in \mathcal{S}(\boldsymbol{q})$, then either there exists A_i with $A_i \notin \{B_i \mid 1 \leq i \leq m_1\}$ or there exists $\exists S_i.\boldsymbol{q}_i$ such that for every $R_j.\boldsymbol{q}_j$ with $S_i = R_j$, there exists $\mathcal{A}, a \in \mathcal{S}(\boldsymbol{q}_i)$ such that $\mathcal{A} \not\models \boldsymbol{q}'_i(a)$. By induction hypothesis, $\boldsymbol{q}'_i \not\models \boldsymbol{q}_i$. But then $\boldsymbol{q}' \not\models \boldsymbol{q}$, as required. ❑

E.6 Proof of Theorem 28

Suppose we are given a $\mathcal{P}^{\Sigma}[\mathsf{U}](\mathcal{EL})$-query

$$\boldsymbol{q} = \boldsymbol{r}_0 \wedge (\boldsymbol{l}_1 \,\mathsf{U}\, (\boldsymbol{r}_1 \wedge (\boldsymbol{l}_2 \,\mathsf{U}\, (\ldots (\boldsymbol{l}_n \,\mathsf{U}\, \boldsymbol{r}_n) \ldots))))).$$

To show that it is uniquely characterised by the polynomial-size example set with $(\mathfrak{p}'_0)$–$(\mathfrak{p}'_2)$ and $(\mathfrak{n}'_0)$–$(\mathfrak{n}'_2)$, consider any $\mathcal{Q}^{\Sigma}[\mathsf{U}](\mathcal{EL})$-query

$$\boldsymbol{q}' = \boldsymbol{r}'_0 \wedge (\boldsymbol{l}'_1 \,\mathsf{U}\, (\boldsymbol{r}'_1 \wedge (\boldsymbol{l}'_2 \,\mathsf{U}\, (\ldots (\boldsymbol{l}'_m \,\mathsf{U}\, \boldsymbol{r}'_m) \ldots))))).$$

such that $\boldsymbol{q} \not\equiv \boldsymbol{q}'$.

We now define a map f that reduces the 2D case to the 1D case. Consider the alphabet

$$\Gamma = \{\boldsymbol{r}_0, \ldots, \boldsymbol{r}_n, \boldsymbol{l}_1, \ldots, \boldsymbol{l}_n, \boldsymbol{r}'_0, \ldots, \boldsymbol{r}_m, \boldsymbol{l}'_1, \ldots, \boldsymbol{l}'_m, \}\backslash\{\bot\}$$

in which we regard the $\mathcal{EL}$-queries $\boldsymbol{r}_i, \boldsymbol{l}_i, \boldsymbol{r}'_j, \boldsymbol{l}'_j$ as symbols. Let $\hat{\Gamma} = \{(\hat{\boldsymbol{a}}, a) \mid \boldsymbol{a} \in \Gamma\}$, that is, $\hat{\Gamma}$ consists of the pointed databases corresponding to the $\mathcal{EL}$-queries $\boldsymbol{a} \in \Gamma$.

For any $\mathcal{EL}$ instance query $\boldsymbol{a}$, we set

$$f(\boldsymbol{a}) = \{\boldsymbol{b} \in \Gamma \mid (\hat{\boldsymbol{a}}, a) \models \boldsymbol{b}\}.$$

Similarly, for any $\mathcal{EL}$ pointed data instance $(\mathcal{A}, a)$, we set

$$f(\mathcal{A}, a) = \{\boldsymbol{b} \in \Gamma \mid (\mathcal{A}, a) \models \boldsymbol{b}\}$$

and, for any temporal data instance $\mathcal{D} = (\delta_0, \ldots, \delta_k)$ with $\mathcal{EL}$ pointed data instances (δ_i, a_i), set

$$f(\mathcal{D}) = (f(\delta_0, a_0), \ldots, f(\delta_k, a_k)),$$

which is an *LTL*-data instance over the signature Γ. Finally, we define a query

$$f(\boldsymbol{q}) = \rho_0 \wedge (\lambda_1 \,\mathsf{U}\, (\rho_1 \wedge (\lambda_2 \,\mathsf{U}\, (\ldots (\lambda_n \,\mathsf{U}\, \rho_n) \ldots))))$$

by taking $\rho_i = f(\boldsymbol{r}_i)$ and $\lambda_i = f(\boldsymbol{l}_i)$, and similarly for $\boldsymbol{q}'$.

By definition, $f(\boldsymbol{q})$ is a $\mathcal{P}^{\Gamma}[\mathsf{U}]$-query (indeed, since $\hat{\boldsymbol{r}}_i, a \not\models \boldsymbol{l}_i$, we have $\boldsymbol{l}_i \in f(\boldsymbol{l}_i) \setminus f(\boldsymbol{r}_i)$, and since $\hat{\boldsymbol{l}}_i, a \not\models \boldsymbol{r}_i$, we have $\boldsymbol{r}_i \in f(\boldsymbol{r}_i) \setminus f(\boldsymbol{l}_i)$), and $f(\boldsymbol{q}')$ is a $\mathcal{Q}_p^{\Gamma}[\mathsf{U}]$-query such that $f(\boldsymbol{q}) \not\equiv f(\boldsymbol{q}')$: it follows immediately from the definition that, for any data instance $\mathcal{D}$, we have $\mathcal{D} \models \boldsymbol{q}$ iff $f(\mathcal{D}) \models f(\boldsymbol{q})$ and similarly for $\boldsymbol{q}'$. By Theorem 14, $\boldsymbol{q}$ and $\boldsymbol{q}'$ are separated by the corresponding example set with $(\mathfrak{p}_0)$–$(\mathfrak{p}_2)$ and $(\mathfrak{n}_0)$–$(\mathfrak{n}_2)$. Notice that the positive examples from $(\mathfrak{p}_0)$–$(\mathfrak{p}_2)$ are exactly the f-images of the examples $(\mathfrak{p}'_0)$–$(\mathfrak{p}'_2)$. So if $f(\boldsymbol{q})$ and $f(\boldsymbol{q}')$ are separated by some $\mathcal{D}$ from $(\mathfrak{p}_0)$–$(\mathfrak{p}_2)$, the corresponding member of $(\mathfrak{p}'_0)$–$(\mathfrak{p}'_2)$ separates $\boldsymbol{q}$ and $\boldsymbol{q}'$.

So suppose $f(\boldsymbol{q})$ and $f(\boldsymbol{q}')$ are separated by some $\mathcal{D}$ from $(\mathfrak{n}_0)$–$(\mathfrak{n}_2)$. If $\mathcal{D} = \Gamma^n$, then it means that the temporal depth of $f(\boldsymbol{q}')$ is less than the temporal depth of $f(\boldsymbol{q}')$, so $m < n$, and the queries $\boldsymbol{q}$ and $\boldsymbol{q}'$ are separated by $\mathcal{A}^n_{\Sigma}$.

Suppose $\mathcal{D} = \Gamma^{n-i}(\Gamma \setminus \{\boldsymbol{a}\})\Gamma^i$. Since $\mathcal{D} \not\models f(\boldsymbol{q})$, we have $f(\boldsymbol{r}_i) \not\subseteq \Gamma \setminus \{\boldsymbol{a}\}$, and so $\hat{\boldsymbol{r}}_i, a \models \boldsymbol{a}$ but $f(\boldsymbol{r}'_i) \subseteq \Gamma \setminus \{\boldsymbol{a}\}$. Then $\hat{\boldsymbol{r}}'_i, a \not\models \boldsymbol{a}$, and so $\boldsymbol{r}'_i \not\models \boldsymbol{r}_i$. Therefore, there is $(\mathcal{A}, a) \in \mathcal{S}(\{\boldsymbol{r}_i\})$ such that $\mathcal{A} \models \boldsymbol{r}'_i(a)$ and $\mathcal{A}^{n-i}_{\Sigma}\mathcal{A}\mathcal{A}^i_{\Sigma}$ separates $\boldsymbol{q}$ and $\boldsymbol{q}'$.

The cases when $\mathcal{D}$ is from $(\mathfrak{n}_1)$ or $(\mathfrak{n}_2)$ are treated in a similar manner.

If $\boldsymbol{q} \in \mathcal{P}^{\Sigma}[\mathsf{U}](\mathcal{ELI})$, the characterisation is exponential as the size of $(\mathfrak{n}_0)$–$(\mathfrak{n}_2)$ is exponential because the exponential size of constructed split partners for $\mathcal{ELI}$-queries.

F Proofs for Section 9

We provide further details of the proof of Theorem 33.

Theorem 33. *(i) The class of safe queries in $\mathcal{Q}_p[\bigcirc, \Diamond_r](\mathcal{ELI})$ is polynomial-time learnable with membership queries.*

(ii) The class $\mathcal{Q}_p[\bigcirc, \Diamond_r](\mathcal{ELI})$ is polynomial-time learnable with membership queries if the learner knows the size of the target query in advance.

(iii) The class $\mathcal{Q}_p[\bigcirc, \Diamond](\mathcal{ELI})$ is polynomially-time learnable with membership queries.

We start by completing the proof for the 1D case. For (i), it remains to consider **step 4**. At that point of the computation, the algorithm has identified all blocks of $\boldsymbol{q}$ but not the sequences of $\Diamond$ and $\Diamond_r$ between them. Suppose that $\mathcal{D} = \bar{\boldsymbol{q}}_0\emptyset^b \ldots \bar{\boldsymbol{q}}_i\emptyset^b\bar{\boldsymbol{q}}_{i+1} \ldots \emptyset^b\bar{\boldsymbol{q}}_n$. We construct $\boldsymbol{q}'$ from the blocks of $\mathcal{D}$ by selecting $\mathcal{R}_{i+1}$ as follows: If $\mathcal{D}_i \models \boldsymbol{q}$ for $\mathcal{D}_i = \bar{\boldsymbol{q}}_0\emptyset^b \ldots \bar{\boldsymbol{q}}_i \bowtie \bar{\boldsymbol{q}}_{i+1} \ldots \emptyset^b\bar{\boldsymbol{q}}_n$ then set $\mathcal{R}_{i+1}$ to be $\leq$. Otherwise, let n_{i+1} be the smallest number such that $\mathcal{D}_i \models \boldsymbol{q}$ for $\mathcal{D}_i = \bar{\boldsymbol{q}}_0\emptyset^b \ldots \bar{\boldsymbol{q}}_i\emptyset^{n_{i+1}}\bar{\boldsymbol{q}}_{i+1} \ldots \emptyset^b\bar{\boldsymbol{q}}_n$. Set $\mathcal{R}_{i+1}$ to be a sequence of $<$ of length n_{i+1}. It is now easy to see that $\boldsymbol{q}'$ fits the example set (E^+, E^-) and so $\boldsymbol{q}' \equiv \boldsymbol{q}$, as required.

For (ii) we have to replace **step 3** by a computation step that ensures that after applying (f) one does not obtain a data instance $\mathcal{D}$ with $\mathcal{D} \models \boldsymbol{q}$. Recall that for bound $n = |\boldsymbol{q}|$ and $\boldsymbol{q}_i = \rho(s)$ a lone conjunct in $\boldsymbol{q}$ with $\rho = \{A_1, \ldots, A_k\}$ the rule (f) is defined as follows:

(f) replace ρ with $(\rho \setminus \{A_1\}\emptyset^b \cdots \emptyset^b\rho \setminus \{A_k\})^n$.

is satisfied. Now, after having computed $\mathcal{D}$ in **step 2**, and a data instance $\mathcal{D}'$ is obtained from $\mathcal{D}$ by applying rule (f) such that $\mathcal{D}' \models \boldsymbol{q}$, then replace $\mathcal{D}$ with $\mathcal{D}'$ and return to **step 2**. If no such $\mathcal{D}'$ exists, proceed to **step 4**. To bound the number of applications of rule (f) notice that at the end of **step 2**, the number of time points in $\mathcal{D}$ other than $\emptyset$ does not exceed $|\boldsymbol{q}|$. Indeed, any time point in $\mathcal{D}$ not in the range of some $h : \boldsymbol{q} \to \mathcal{D}$ would be eliminated by rule (a). We obtain a polynomial bound on the number of applications by observing that each application of rule (f) removes a symbol from ρ.

We now move to the 2D case. The proof extends the argument given above for the 1D case and uses the example set $E = (E^+, E^-)$ defined in the proof of Theorem 27. Intuitively, whenever in the argument above we replace ρ by $\rho \setminus \{A\}$ for a set ρ of atoms, we now replace $\hat{\boldsymbol{r}}$ by an element of the frontier $\mathcal{F}(\hat{\boldsymbol{r}})$. There is only one difficulty: in the 1D case the algorithm starts with data instances $\sigma \cdots \sigma$ with σ the signature of the target query, whereas now we have to start with data instances $\mathcal{A}_\Sigma \cdots \mathcal{A}_\Sigma$ with Σ the signature of $\boldsymbol{q}$ and $\mathcal{A}_\Sigma = \{R(a,a), A(a) \mid R, A \in \Sigma\}$. The atemporal data instance $\mathcal{A}_\Sigma$, however, is not tree-shaped, and we have not yet discussed frontiers for data instances that are not tree-shaped. Indeed, in (ten Cate and Dalmau 2021), frontiers are not only computed for tree-shaped data instances but for a generalisation called *c-acyclic* data instances with cycles through the distinguished node. We could at this point introduce the relevant machinery from (ten Cate and Dalmau 2021) and work with frontiers for c-acyclic data instances. Instead, we show that one can use the machinery we have introduced already and work with frontiers of tree-shaped data instances. But we require a straightforward intermediate step that transfers the data instance $\mathcal{A}_\Sigma$ into a tree-shaped data instance $\mathcal{D}$ with $\mathcal{D} \models \boldsymbol{q}$ using membership queries.

We adjust **step 1** of the learning algorithm from the 1D case as follows. Assume $\boldsymbol{q}$ is the target query and let $\Sigma = sig(\boldsymbol{q})$. We aim to identify an initial temporal data instance $\mathcal{D}_0 = \mathcal{A}_0, \dots, \mathcal{A}_n$ with a designated individual a such that

- $\mathcal{D}_0, a, 0 \models \boldsymbol{q}$,
- if $\mathcal{D}'$ is obtained from $\mathcal{D}_0$ by removing an atom then $\mathcal{D}', a, 0 \not\models \boldsymbol{q}$, and
- all $\mathcal{A}_i$ are tree-shaped with distinguished node a.

By asking incrementally membership queries of the form '$\mathcal{A}^k_\Sigma, a, 0 \models \boldsymbol{q}$?', we identify the number of $\bigcirc$ and $\Diamond$ in $\boldsymbol{q}$. Let $b = \min\{k \mid (\mathcal{A}_\Sigma)^k, a, 0 \models \boldsymbol{q}\} + 1$.

Let $\mathcal{D}_0 = \mathcal{A}_0, \dots, \mathcal{A}_n$, where $n = b - 2$ and, initially, $A_i = \mathcal{A}_\Sigma$ for $i = 1, \dots, n$. Before progressing, we make $\mathcal{A}_i$ tree-shaped by applying the following unwind and minimise operations:

unwind Suppose that $\mathcal{A}_i$ contains an atom $S(c,c)$. Then introduce fresh individuals c_R, c_{R^-} for every binary predicate R with $R(c,c) \in \mathcal{A}_i$, remove all $R(c,c)$ from $\mathcal{A}_i$, and add instead $R(c,c_R)$, $R(c_{R^-},c)$, and $A(c_R)$, $A(c_{R^-})$ $R'(c_R,c_R)$, and $R'(c_{R^-},c_{R^-})$ for all $A, R' \in \Sigma$. Let $\mathcal{A}'_i$ denote the resulting data instance and set $\mathcal{D}' = \mathcal{A}_0, \dots, \mathcal{A}_{i-1}, \mathcal{A}'_i, \mathcal{A}_{i+1}, \dots, \mathcal{A}_n$.

minimise Remove exhaustively atoms from $\mathcal{A}'_i$ as long as $\mathcal{D}', a, 0 \models \boldsymbol{q}$.

Observe that if $\mathcal{D} \models \boldsymbol{q}$ and $\mathcal{D}'$ is obtained from $\mathcal{D}$ by an unwinding step, then $\mathcal{D}' \models \boldsymbol{q}$. After the minimise step, the size of $\mathcal{D}'$ does not exceed the size of $\boldsymbol{q}$. Therefore we can replace $\mathcal{D}_0$ with $\mathcal{D}'$. By applying the unwind-minimise steps exhaustively, we eventually eliminate all loops from $\mathcal{A}_i$. It remains to notice that the minimise step can be implemented by querying the membership oracle.

The remaining steps of the learning algorithm remain the same as before by replacing the removal of an atom by taking an element of the frontier of a tree-shaped data instance.

For (iii), notice that in part (ii) the learner actually only needs to know the temporal depth of the goal query, not the overall size. Hence (iii) also reduces to (ii).