



**TECHNISCHE
UNIVERSITÄT
DRESDEN**

**Technische Universität Dresden
Institute for Theoretical Computer Science
Chair for Automata Theory**

LTCS-Report

Ontology-Based Query Answering for Probabilistic Temporal Data (Extended Version)

Patrick Koopmann

LTCS-Report 18-13

This is an extended version of the article to appear in the proceedings of AAAI 2019.

Postal Address:
Lehrstuhl für Automatentheorie
Institut für Theoretische Informatik
TU Dresden
01062 Dresden

<http://lat.inf.tu-dresden.de>

Visiting Address:
Nöthnitzer Str. 46
Dresden

Contents

1	Introduction	2
2	Preliminaries	4
3	Temporal Probabilistic Knowledge Bases and Queries	5
4	Hardness of TPQ Entailment	9
5	Deciding TPQ Entailment	10
6	Positive TPQs	12
7	Conclusion	13
8	Acknowledgements	13
A	Proofs of Theorems and Lemmata	15
A.1	Lower Bounds	15
A.2	Correctness of the Decision Procedure	17
A.3	Deciding r-Satisfiability	21
A.4	Complexity Upper Bounds	26
A.5	Positive Temporal Probabilistic Queries	27

Ontology-Mediated Query Answering for Probabilistic Temporal Data (Extended Version)*

Patrick Koopmann

December 20, 2018

Abstract

We investigate ontology-based query answering for data that are both temporal and probabilistic, which might occur in contexts such as stream reasoning or situation recognition with uncertain data. We present a framework that allows to represent temporal probabilistic data, and introduce a query language with which complex temporal and probabilistic patterns can be described. Specifically, this language combines conjunctive queries with operators from linear time logic as well as probability operators. We analyse the complexities of evaluating queries in this language in various settings. While in some cases, combining the temporal and the probabilistic dimension in such a way comes at the cost of increased complexity, we also determine cases for which this increase can be avoided.

1 Introduction

The internet has become highly dynamic, with information being frequently added and changed, and new data being generated from a variety of sources. In addition, new technologies such as smart phones and the internet of things (IoT) frequently encounter a data environment that is constantly changing. To make use of these data, there has been an increasing interest in investigating semantic and reasoning techniques that process not only static data, but streams of data, such as in the semantic stream reasoning paradigm [29]. One application is that of *situation recognition*, where we want to recognise or query temporal patterns in a stream of data. As [29] illustrate, frequently, the data encountered in stream reasoning applications is not only temporal, but also probabilistic in nature. In *ontology-based query answering* (OBQA), queries are evaluated with respect to an *ontology*, which specifies background knowledge about the domain of interest. Using a reasoner, this allows to query also information that follows implicitly from the data. While OBQA was originally designed for querying static and precise data, there is good motivation also for semantic stream reasoning as well as for querying historical data, where data are temporal and probabilistic.

As an example, consider a health or fitness monitoring application, for which one may want to use concepts from a medical ontology such as SNOMED CT [18] to describe information about the health status of a patient. Specifically, such an application could be used on a smartphone in combination with a sensor that measures the diastolic blood pressure of the patient while he is exercising [25]. As the sensor might be imprecise in its measurements, it might report information about whether the blood pressure of the patient is high with an associated probability, and provide this information to the application in regular time intervals. If a too

*Supported by the DFG within the collaborative research center SFB 912 (HAEC).

high blood pressure was observed for several times during a short period, the application should give a warning to the patient, and advise him to take a break from his exercise.

We assume a representation of the data in form of a sequence of probabilistic data sets, which may have been obtained using further preprocessing and windowing operations. A typical query would then ask whether, with a high probability, the patient had at least twice a high blood pressure during the last 10 minutes. In order to properly take both the temporal and the probabilistic aspects of this question into account when querying the probabilistic stream, we propose a query language for OBQA that comes with both temporal and probabilistic operators. In this language, the query would be expressed as follows, where `HighBloodPressure` is a concept defined in the ontology.

$$P_{>.8}(\circ^{-10}\diamond(\text{HighBloodPressure}(x) \\ \wedge \circ\diamond\text{HighBloodPressure}(x)))$$

Our language is an extension of the well-investigated temporal query language introduced in [8, 3], which extends conjunctive queries with operators from linear temporal logic (LTL). Other authors considered using these operators also as part of the DL, either to describe temporal concepts [20], or to make the axioms of the ontology itself temporal [4]. Recently, this work has been extended also to metric temporal logics [2, 10]. Temporal reasoning for streams of data has also been considered in the context of Datalog [31]. A recent survey on temporal query answering with ontologies can be found in [1].

A major restriction of using temporal concepts in the DL is that we cannot keep relations between objects stable throughout the timeline (rigid) without making the DL undecidable. This limits their application for querying situations concerning more than one object, which in applications more involved than in our running example might be crucial. For this reason, we focus on extensions of the query language rather than the DL in this paper, though investigating an extension of this framework with temporal DLs might be interesting as future work.

In addition to the temporal dimension, we add a probabilistic dimension to our setting. An OBQA framework for *probabilistic* data was presented in [24], though a temporal dimension was not considered here yet. Since this publication, several authors investigated OBQA in similar settings [6, 5, 15]. In addition to settings based on probabilistic databases, there is also research on extending DLs with probability operators, such as in $\mathbf{P}\text{-SHLF}(\mathcal{D})/\mathbf{P}\text{-SHOIN}(\mathcal{D})$ [26] or $\text{Prob-}\mathcal{ALC}/\text{Prob-}\mathcal{EL}$ [23]. The probability operator used in our query language syntactically and semantically corresponds to the probability operator in $\text{Prob-}\mathcal{ALC}$ and $\text{Prob-}\mathcal{EL}$.

To our knowledge, the only work that combines both temporal and probabilistic query answering in the presence of description logic ontologies is [14]. Albeit, the authors consider a different setting, in which the flow of time is modeled by a Markov-process, and not by a sequence of observations as in our case. Moreover, they do not consider a rich query language like ours, but focus on computing the probability that some axiom is entailed in some given time range. [17] consider temporal probabilistic databases with temporal Datalog rules and constraints, and computing probabilities of conjunctive queries in these KBs. Both works do not allow for nested probabilities as part of the query language.

To handle scenarios like in our example, we propose a framework that combines the ideas from [8] for temporal knowledge bases with the framework for probabilistic knowledge bases introduced in [24]. To query data in the resulting temporal probabilistic knowledge bases, our language extends temporal queries with probabilistic operators, to allow to assign probability bounds to arbitrary parts of the query. We establish a more or less complete picture of the complexity of query entailment in this framework for various DLs (see Figure 1, explained in detail throughout the text), and also discuss a restricted variant of our query language without negation, which sometimes leads to a restricted complexity.

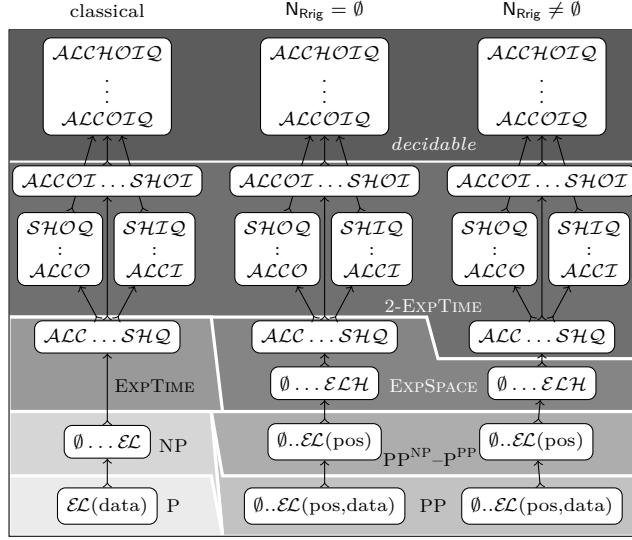


Figure 1: Complexity of TPQ Entailment vs. classical CQ entailment. Here, \emptyset corresponds to the case without TBox. Except for the P^{PP} and the decidability results, all complexity bounds are tight.

Detailed proofs can be found in the appendix.

2 Preliminaries

We recall the DLs studied in the paper, conjunctive query answering, and probabilistic complexity classes.

Description Logics. Let N_C , N_R and N_I be pair-wise countably infinite sets of respectively *concept names*, *role names* and *individual names*. A *role* is an expression of the forms r , r^- , where $r \in N_R$. *Concepts* are of the following forms, where $A \in N_C$, R is a role, C, D are concepts, $n \in \mathbb{N}$ and $a \in N_I$:

$$A \mid C \sqcap D \mid \exists R.C \mid \forall R.C \mid \geq nR.C \mid \{a\}.$$

A *TBox* is a set of *axioms* of the forms $C \sqsubseteq D$, $R \sqsubseteq S$ and $\text{trans}(R)$, where C, D are concepts and R, S roles, while an *ABox* is a set of assertions of the forms $A(a)$ and $r(a, b)$, $A \in N_C$, $r \in N_R$, $a, b \in N_I$. For a TBox \mathcal{T} , we define the relation $\prec_{\mathcal{T}}$ s.t. for two roles R, S , $S \prec_{\mathcal{T}} R$ holds if $S' \sqsubseteq R' \in \mathcal{T}$ with $S, S' \in \{s, s^-\}$ and $R, R' \in \{r, r^-\}$, $r, s \in N_R$. A role R is *complex wrt.* \mathcal{T} if $\text{trans}(S) \in \mathcal{T}$ for some role S s.t. $S \prec_{\mathcal{T}}^* R$. To ensure decidability, we require for every concept of the form $\geq nR.C$ in \mathcal{T} that R is not complex. Now a *knowledge base* (KB) is a tuple $\langle \mathcal{T}, \mathcal{A} \rangle$ of a TBox \mathcal{T} and an ABox \mathcal{A} . We differentiate different DLs based on the operators allowed: \mathcal{EL} only supports concepts of the form A , $C \sqcap D$ and $\exists R.C$ and axioms of the form $C \sqsubseteq D$, no roles of the form r^- , and no axioms of the forms $R \sqsubseteq S$ or $\text{trans}(R)$. \mathcal{ALC} extends \mathcal{EL} with concepts of the form $\neg C$, and \mathcal{S} extends \mathcal{ALC} with axioms of the form $\text{trans}(R)$. More expressive DLs are denoted by attaching a letter to the DL, where we use \mathcal{I} for support of roles r^- , \mathcal{O} for concepts of the form $\{a\}$, \mathcal{Q} for concepts of the form $\geq nR.C$, and \mathcal{H} for axioms of the form $R \sqsubseteq S$. For example, \mathcal{SHI} extends \mathcal{S} with axioms of the form $R \sqsubseteq S$ and roles of the form r^- , whereas \mathcal{ALCHOQ} extends \mathcal{ALC} with concepts of the form $\{a\}$ and $\geq nR.C$. Depending on the DL \mathcal{L} used, we speak of \mathcal{L} concepts, \mathcal{L} axioms, \mathcal{L} TBoxes and \mathcal{L} KBs.

The semantics of KBs is defined in terms of *interpretations* $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, where $\Delta^{\mathcal{I}}$ is a set of *domain elements* and $\cdot^{\mathcal{I}}$ maps each concept name $A \in \mathbf{N}_C$ to a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, each role name $r \in \mathbf{N}_R$ to a relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, each individual name $a \in \mathbf{N}_I$ to a domain element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, and each role r^- to $(r^-)^{\mathcal{I}} = (r^{\mathcal{I}})^-$. It is extended to concepts as follows.

$$\begin{aligned} (C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}}, (-C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}, \{a\}^{\mathcal{I}} = \{a^{\mathcal{I}}\}, \\ (\exists R.C)^{\mathcal{I}} &= \{d \in \mathcal{I} \mid \exists e \in \Delta^{\mathcal{I}} : \langle d, e \rangle \in R^{\mathcal{I}}, e \in C^{\mathcal{I}}\}, \\ (\geq nR.C)^{\mathcal{I}} &= \{d \in \mathcal{I} \mid \#\{e \in \Delta^{\mathcal{I}} \mid \\ &\quad \langle d, e \rangle \in R^{\mathcal{I}}, e \in C^{\mathcal{I}}\} \geq n\} \end{aligned}$$

We say that an interpretation \mathcal{I} *satisfies* an axiom/assertion α , in symbols $\mathcal{I} \models \alpha$, if $\alpha = C \sqsubseteq D$ and $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$; $\alpha = R \sqsubseteq S$ and $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$; $\alpha = \text{trans}(R)$ and $R^{\mathcal{I}} = (R^{\mathcal{I}})^+$; $\alpha = A(a)$ and $a^{\mathcal{I}} \in A^{\mathcal{I}}$; and $\alpha = r(a, b)$ and $\langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in r^{\mathcal{I}}$. \mathcal{I} is a *model* of a TBox/ABox/KB iff it satisfies all axioms in it. Finally, a TBox/ABox/KB \mathcal{K} *entails* an axiom/assertion α iff α is satisfied by every model of \mathcal{K} .

Conjunctive Queries. A *conjunctive query* (CQ) takes the form $q = \exists \vec{y}. \phi(\vec{x}, \vec{y})$, where \vec{x}, \vec{y} are vectors of variables and $\phi(\vec{x}, \vec{y})$ is a conjunction over atoms of the forms $A(t_1)$ and $r(t_1, t_2)$, where $A \in \mathbf{N}_C$ and $r \in \mathbf{N}_R$ is not complex, and t_1 and t_2 are *terms* taken from \mathbf{N}_I , \vec{x} and \vec{y} . \vec{x} are the *answer variables* of q . Given an interpretation \mathcal{I} and a CQ q with answer variables x_1, \dots, x_n , the vector $a_1 \dots a_n \subseteq \mathbf{N}_I^n$ is an *answer of q in \mathcal{I}* if there exists a mapping $\pi : \text{term}(q) \rightarrow \Delta^{\mathcal{I}}$ s.t. $\pi(x_i) = a_i$ for $i \in \llbracket 1, n \rrbracket$, $\pi(b) = b^{\mathcal{I}}$ for $b \in \mathbf{N}_I$, $\pi(t) \in A^{\mathcal{I}}$ for every $A(t)$ in q , and $\langle \pi(t_1), \pi(t_2) \rangle \in r^{\mathcal{I}}$ for every $r(t_1, t_2)$ in q . A vector $a_1 \dots a_n$ is a *certain answer* of q in a KB \mathcal{K} if it is an answer in every model of \mathcal{K} . If a query does not contain any answer variables, it is a *Boolean CQ*, and we say it is *entailed* by a KB \mathcal{K} (interpretation \mathcal{I}) if it has the empty vector as answer. The complexity of query entailment for KBs in various DLs is shown in the left-part of Figure 1.

Probabilistic Complexity Classes. The complexity class PP is defined using *probabilistic Turing machines*, which are like non-deterministic Turing machines, but with an alternative acceptance condition: namely, they accept an input iff at least half of the computation paths end in an accepting state. PP describes the class of all problems that can be decided by a probabilistic Turing machine in which all paths are polynomially bounded by the size of the input. By using oracles, we can obtain the complexity classes PP^{NP} and P^{PP} , for which we have the relations $\text{NP} \cup \text{coNP} \subseteq \text{PP} \subseteq \text{PP}^{\text{NP}} \subseteq \text{P}^{\text{PP}} \subseteq \text{PSPACE}$ [33]. Strongly related to the decision class PP is the function class $\#\text{P}$, which is the class of functions that can be computed by counting accepting paths in a non-deterministic polynomial time-bounded Turing machine.

3 Temporal Probabilistic Knowledge Bases and Queries

We introduce our framework for temporal probabilistic query answering.

Temporal Probabilistic Knowledge Bases. Regarding the probabilistic aspects, we follow the paradigm introduced in [24] for atemporal probabilistic KBs. To keep things simple, we focus on the easiest type of probabilistic fact bases presented there, the so-called *assertion-independent probabilistic ABoxes* (ipABoxes). Here, assertions are assigned probabilities, which are assumed to be statistically independent. They correspond to the tuple-independent probabilistic databases studied in [16].

$\Omega_{\mathcal{K}}$	\mathcal{A}'_1	\mathcal{A}'_3	\mathcal{A}'_4	$\mu_{\mathcal{K}}$
w_1	hBP(p, b), HBP(b)	HBP(b)	HBP(b)	0.378
w_2	hBP(p, b)	HBP(b)	HBP(b)	0.162
w_3	hBP(p, b), HBP(b)	\emptyset	HBP(b)	0.042
w_4	hBP(p, b)	\emptyset	HBP(b)	0.018
w_5	hBP(p, b), HBP(b)	HBP(b)	\emptyset	0.252
w_6	hBP(p, b)	HBP(b)	\emptyset	0.108
w_7	hBP(p, b), HBP(b)	\emptyset	\emptyset	0.028
w_8	hBP(p, b)	\emptyset	\emptyset	0.012

Table 1: Probability space of example TPKB.

An *ipABox* is a set of *probabilistic ABox assertions* of the form $\alpha : p$, where α is an ABox assertion and p is a probability value between 0 and 1. Intuitively, $\alpha : p$ expresses that the assertion α holds with a probability of at least p . Instead of $\alpha : 1$, we may just write α if the meaning is clear from the context. ipABoxes only specify a *lower bound* on the probability, to conform with the open-world semantics common in ontology-based representations. This means, we might infer using other information in the KB that the probability is in fact higher.¹ A temporal probabilistic KB (TPKB) is now a tuple $\langle \mathcal{T}, (\mathcal{A}_i)_{i \in [1, n]} \rangle$, where \mathcal{T} is a TBox and $(\mathcal{A}_i)_{i \in [1, n]}$ is a sequence of ipABoxes.

We define the semantics of TPKBs using the *possible worlds semantics*, as common to probabilistic logics and databases [16]. For a given a TPKB $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{i \in [1, n]} \rangle$, the set $\Omega_{\mathcal{K}}$ of *possible worlds of \mathcal{K}* contains all sequences $w = (\mathcal{A}'_i)_{i \in [1, n]}$ of classical ABoxes such that for every $i \in [1, n]$ and $\alpha \in \mathcal{A}'_i$, \mathcal{A}_i contains an axiom of the form $\alpha : p$. Each TPKB uniquely defines a probability space $\langle \Omega_{\mathcal{K}}, \mu_{\mathcal{K}} \rangle$, where the probability measure $\mu_{\mathcal{K}} : 2^{\Omega_{\mathcal{K}}} \rightarrow [0, 1]$ satisfies

$$\mu_{\mathcal{K}}(\{(\mathcal{A}'_i)_{i \in [1, n]}\}) = \prod_{\substack{i \in [1, n] \\ \alpha : p \in \mathcal{A}_i \\ \alpha \in \mathcal{A}'_i}} p \cdot \prod_{\substack{i \in [1, n] \\ \alpha : p \in \mathcal{A}_i \\ \alpha \notin \mathcal{A}'_i}} (1 - p)$$

and for $W \subseteq \Omega_{\mathcal{K}}$, $\mu_{\mathcal{K}}(W) = \sum_{w \in W} \mu(\{w\})$. Intuitively, $\mu_{\mathcal{K}}(W)$ gives the probability of being in one of the possible worlds in W , by summing up the probabilities of each possible world. The definition of $\mu_{\mathcal{K}}(W)$ reflects the assumption that all probabilities in the TPKB are statistically independent.

Example 1. We define the TPKB $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{i \in [1, 5]} \rangle$, where \mathcal{T} contains the axiom

$$\begin{aligned} \text{HighBloodPressurePatient} &\equiv \\ &\exists \text{hasBloodPressure.HighBloodPressure,} \end{aligned}$$

and the probabilistic ABoxes are

$$\begin{aligned} \mathcal{A}_1 &= \{\text{hasBloodPressure}(a, b), \\ &\quad \text{HighBloodPressure}(b):0.7\} \\ \mathcal{A}_3 &= \{\text{HighBloodPressure}(b): 0.9\} \\ \mathcal{A}_4 &= \{\text{HighBloodPressure}(b): 0.6\} \end{aligned}$$

and $\mathcal{A}_2 = \mathcal{A}_5 = \emptyset$. Every possible world $w = (\mathcal{A}'_i)_{i \in [1, 5]}$ with $\text{hasBloodPressure}(a, b) \notin \mathcal{A}'_1$ has probability $\mu_{\mathcal{K}}(w) = 0$. The remaining possible worlds, excluding time points 2 and 5, are shown in Figure 1, with the probability measure $\mu_{\mathcal{K}}$ shown in the last column, where hBP is short for hasBloodPressure and HBP is short for HighBloodPressure. \square

¹This is different to the open-world semantics for probabilistic databases suggested in [13], which assumes a fixed upper probability for facts absent in the data.

Remark 1. We follow the semantical idea of ipABoxes, in which all assertions are assumed to be statistically independent, mainly to keep the representation simple. Of course, for realistic applications, the assumption that all probabilistic facts are statistically independent is not always accurate, and already [24] specify a more general concept of probabilistic ABoxes. The complexity upper bounds established in this paper only rely on a fixed probability distribution over the possible worlds, which is why they can be easily extended to more refined settings. \square

Based on the probability measure, we define models by assigning to each possible world a sequence of classical interpretations. A *model* of a TPKB $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{i \in \llbracket 1, n \rrbracket} \rangle$ is a mapping ι from possible worlds $w = (\mathcal{A}'_i)_{i \in \llbracket 1, n \rrbracket} \in \Omega_{\mathcal{K}}$ to sequences $(\iota(w)_i)_{i > 0}$ of (classical) models of \mathcal{T} s.t. for all $i \in \llbracket 1, n \rrbracket$, $\iota(w)_i$ is a model of the classical knowledge base $\langle \mathcal{T}, \mathcal{A}'_i \rangle$, and all $\iota(w)_i$ have the same set Δ^ι of domain elements (constant domain assumption).

Rigid Names. As typical for temporal knowledge bases, we may assume in addition a set N_{rig} of *rigid names*, containing the set $N_{\text{Crig}} \subseteq N_{\text{C}}$ of *rigid concept names* and the set $N_{\text{Rrig}} \subseteq N_{\text{R}}$ of *rigid role names*. Rigid names denote names whose interpretation is independent of the flow of time. We say that a model ι of a TPKB $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{i \in \llbracket 1, n \rrbracket} \rangle$ *respects rigid names* iff for all $w \in \Omega_{\mathcal{K}}$, $i, j \in \llbracket 1, n \rrbracket$ and $X \in N_{\text{rig}}$, $X^{\iota(w)_i} = X^{\iota(w)_j}$. Allowing for rigid names often has a direct impact on complexity and decidability of common reasoning problems, which is why typically different cases based on whether $N_{\text{Crig}} = \emptyset$ or $N_{\text{Rrig}} = \emptyset$ are studied for complexity.

Example 2. In the above example, the relation `hasBP` is rigid, as its interpretation should be independent of time, while the concept `HighBP` is not rigid, as the blood pressure of a patient can change from high to not high. As a consequence, the individual p will be related to the blood pressure b at all time points, even though the assertion `hasBP(p, b)` only occurs in \mathcal{A}_1 . \square

Temporal Probabilistic Queries. To query temporal data in the OBDA framework, extensions of conjunctive queries with operators from linear temporal logic (LTL) have been considered and well-investigated as *temporal queries* (TQs) [8]. When applied on a temporal probabilistic KB, an assignment of the answer variables in a TQ becomes an answer with a certain probability, depending on the query, we might be interested only in answers that holds with a certain minimal or maximal probability. Rather than just assigning an overall probability threshold, we might want to mark parts of the query with different probability upper and lower bounds. For example, in the scenario sketched in the introduction, the smart-phone could be equipped with a motion sensor to detect the probability that the patient is currently exercising, and one might want to detect situations in which the probability of them exercising is low, while the probability of his blood pressure being above some threshold is high.

To be able to describe all this, we extend TQs with probability operators. A *temporal probabilistic query* (TPQ) is of one of the following forms, where q is a CQ, ϕ_1 and ϕ_2 are a TPQs, $p \in [0, 1]$ and $\sim \in \{<, \leq, =, \geq, >\}$.

$$q \mid \neg\phi_1 \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \circ\phi_1 \mid \diamond\phi_1 \mid \square\phi_1 \mid \phi_1 \mathcal{U}\phi_2 \\ \circ^-\phi_1 \mid \diamond^-\phi_1 \mid \square^-\phi_1 \mid \phi_1 \mathcal{S}\phi_2 \mid P_{\sim p}\phi_1$$

The propositional operators, \circ (next), \diamond (eventually), \mathcal{U} (until) are found in TQs, while $P_{\sim p}$ is the operator that we add to this language. Note that due the disjunction operator, we can also express *unions of conjunctive queries* (UCQs), which are simply disjunctions of CQs. The answer variables of a TPQ ϕ are the answer variables of the CQs in ϕ . A TPQ ϕ is *Boolean* if every variable in ϕ is bound by an existential quantifier.

In order to define the semantics of TPQs, we have to take into consideration the two dimensions in which queries refer to a temporal probabilistic model. While the temporal operators refer to

ϕ	$\iota, w, i \models \phi$ iff
$\exists \vec{y}.\psi(\vec{y})$	$\iota(w), i \models \exists \vec{y}.\psi(\vec{y})$
$\neg\phi_1$	$\iota, w, i \not\models \phi_1$
$\phi_1 \wedge \phi_2$	$\iota, w, i \models \phi_1$ and $\iota, w, i \models \phi_2$
$\phi_1 \vee \phi_2$	$\iota, w, i \models \phi_1$ or $\iota, w, i \models \phi_2$
$\bigcirc\phi_1$	$\iota, w, i + 1 \models \phi_1$
$\bigcirc^-\phi_1$	$\iota, w, i - 1 \models \phi_1$ and $i > 0$
$\diamond\phi_1$	$\iota, w, j \models \phi_1$ for some $j \geq i$
$\diamond^-\phi_1$	$\iota, w, j \models \phi_1$ for some $j \leq i$
$\square\phi_1$	$\iota, w, j \models \phi_1$ for all $j \geq i$
$\square^-\phi_1$	$\iota, w, j \models \phi_1$ for all $j \leq i$
$\phi_1 \mathcal{U} \phi_2$	$\iota, w, j \models \phi_2$ for some $j \geq i$, and $\iota, w, k \models \phi_1$ for all $k \in \llbracket i, j - 1 \rrbracket$
$\phi_1 \mathcal{S} \phi_2$	$\iota, w, j \models \phi_2$ for some $j \leq i$, and $\iota, w, k \models \phi_1$ for all $k \in \llbracket j + 1, i \rrbracket$
$P_{\sim p}\phi$	$\mu_{\mathcal{K}}(\{w' \in \Omega_{\mathcal{K}} \mid \iota, w', i \models \phi\}) \sim p$, where $\sim \in \{<, \leq, =, \geq, >\}$

Table 2: Entailment of Boolean TPQs under interpretation ι .

the time line in a single possible world, for the probability operators we have to aggregate the probabilities of the possible worlds in which a query is entailed.

Let \mathcal{K} be a TPKB, ι a model of \mathcal{K} , and ϕ a Boolean TPQ. For a single possible world $w \in \Omega_{\mathcal{K}}$ and a time point i , we say that ϕ is satisfied at w, i under ι , in symbols $\iota, w, i \models \phi$ iff the conditions in Table 2 are satisfied. Note that the temporal operators refer to the time line of a single possible world, for which they are defined as in [8]. In contrast, the probability operators refer to the current time point in multiple possible worlds, and are defined similar to the probabilistic concept constructor in Prob- \mathcal{ALC} [23]. A Boolean TPQ ϕ is *satisfied* in an interpretation ι at i , in symbols $\iota, i \models \phi$, iff $\iota, w, i \models \phi$ for all $w \in \Omega_{\mathcal{K}}$. It is *entailed* by the TPKB \mathcal{K} at i , in symbols $\mathcal{K}, i \models \phi$, iff $\iota, i \models \phi$ for all models ι of \mathcal{K} . ϕ is *satisfiable in \mathcal{K} at i* iff there exists a model ι of \mathcal{K} s.t. $\iota, i \models \phi$. Note that satisfiability is complementary to entailment: namely, ϕ is satisfiable in \mathcal{K} at i iff $\mathcal{K}, i \not\models \neg\phi$.

Now given a TPKB \mathcal{K} , a TPQ ϕ with answer variables \vec{x} , a time point $i > 0$, and a mapping $\sigma : \vec{x} \rightarrow \mathbb{N}_1$, σ is a *certain answer for ϕ in \mathcal{K} at i* iff $\mathcal{K}, i \models \phi'$, where ϕ' is the result of applying σ on ϕ . As common, since computing answers for TPQs can be seen as a search problem that uses Boolean TPQ entailment, we focus on the decision problem of query entailment, and may refer to Boolean TPQs simply as TPQs.

Example 3. We consider a slight variation of the query from the introduction.

$$P_{>.8}(\bigcirc^{-5}\diamond(\text{HighBPP}(x) \wedge \bigcirc\bigcirc\bigcirc\bigcirc\bigcirc\text{HighBPP}(x)))$$

For $x = p$ and time point 5, the query below the probability operator is entailed in every model of the possible worlds w_1, w_2, w_3 and w_5 , which together have a probability of 0.834. Consequently, p is an answer to the query at time point 5. Now consider the variation where the probability operators are moved inside.

$$\bigcirc^{-5}\diamond(P_{>.8}(\text{HighBPP}(x)) \wedge \bigcirc\bigcirc P_{>.8}(\text{HighBPP}(x)))$$

This corresponds to the situation where at least twice in the last 5 time units, the probability of having a high blood pressure was above 0.8. As this probability is only once above this bound, this query is not entailed.

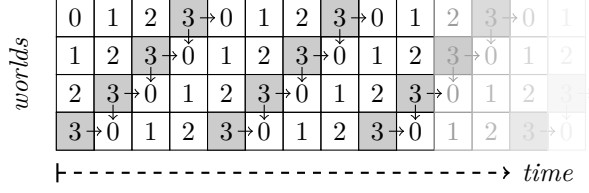


Figure 2: Illustration of how the counters are used in the possible worlds.

The complexity of TPQ entailment for various DLs is shown in Figure 1, where we compare against the complexity of classical query entailment (left column), and distinguish the cases based on whether $\mathbf{N}_{\text{Rrig}} = \emptyset$ (middle column) or $\mathbf{N}_{\text{Rrig}} \neq \emptyset$ (right column). All complexities remain tight independent on whether we admit rigid concept names ($\mathbf{N}_{\text{Crig}} \neq \emptyset$). Note that the EXPSPACE-result for \mathcal{ELH} remains tight even without any TBox. Results marked with (pos) regard *positive TPQs*, which we discuss towards the end of the paper.

4 Hardness of TPQ Entailment

We show that TPQ satisfiability, and thus entailment, is EXPSPACE-hard even if $\mathcal{T} = \emptyset$ and $\mathbf{N}_{\text{Crig}} = \mathbf{N}_{\text{Rrig}} = \emptyset$, by reduction of the exponential variant of the corridor tiling problem [34]. In this problem, we are given a set T of *tile types*, two special tile types $t_s, t_e \in T$, a natural number n , and two functions v and h of *compatibility constraints* $v : T \rightarrow 2^T$ (vertical) and $h : T \rightarrow 2^T$ (horizontal). The input is an *instance* of the exponential corridor tiling problem if there exists a number $m \in \mathbb{N}$ and a *tiling* $f : \llbracket 0, m \rrbracket \times \llbracket 0, 2^n - 1 \rrbracket \rightarrow T$ such that $f(0, 0) = t_s$, $f(m, 0) = t_e$, and for all $x \in \llbracket 0, m \rrbracket$ and $y \in \llbracket 0, 2^n - 1 \rrbracket$, if $x < m$, $f(x + 1, y) \in h(f(x, y))$ and if $y < 2^n - 1$, $f(x, y + 1) \in v(f(x, y))$.

We use n concept names A_i to mark the different possible worlds $w \in \Omega_{\mathcal{K}}$ with a counter, such that in interpretations ι that satisfy both the TPQ and the TPKB, $\iota, w, j \models A_i(a)$ iff the i th bit of the counter is 1 at time point j , and $\iota, w, j \not\models A_i(a)$ iff the i th bit is 0 at time point j . The ipABox $\mathcal{A}_1 = \{A_i(a) \sim 0.5 \mid i \in \llbracket 1, n \rrbracket\}$ assigns every possible world a different counter value. Our query makes sure that the counter values are increased for each time point. Figure 2 illustrates this idea. Each possible world corresponds to a row in the tiling, with its counter value at time point 1 denoting the row number.

At each time point, two possible worlds can be recognised by simple queries: the one whose counter value is 0 (which satisfies $\bigwedge_{1 \leq i \leq n} \neg A_i(a)$), and the one whose counter value is $2^n - 1$ (which satisfies $\bigwedge_{1 \leq i \leq n} A_i(a)$). Unless the latter one represents the last row, these worlds correspond to neighbours in the tiling, which means that for these worlds, we can enforce the vertical tiling conditions with the following query, where $L(a)$ is an assertion that marks the last row, and for a tile type $t \in T$, $B_t(a)$ expresses that the current cell has a tile of type t .

$$\begin{aligned} & \square \bigwedge_{t_1 \in T} \left(\left(B_{t_1}(a) \wedge \bigwedge_{i \in \llbracket 1, n \rrbracket} A_i(a) \wedge \neg L(a) \right) \right. \\ & \quad \left. \rightarrow \bigvee_{t_2 \in v(t_1)} \text{P}_{=1} \left(\left(\bigwedge_{i \in \llbracket 1, n \rrbracket} \neg A_i(a) \right) \rightarrow B_{t_2}(a) \right) \right) \end{aligned}$$

As we can only check the vertical tiling conditions for one pair of rows at a time, we represent each cell by up to 2^n succeeding time points in each possible world, switching to the next tile only when the counter reaches $2^n - 1$. The remaining details of the reduction can be found in

the appendix. The hardness for \mathcal{ALC} with rigid roles follows from the non-probabilistic case [3].

Theorem 4. *The lower bounds regarding general TPQs in Figure 1 hold.*

5 Deciding TPQ Entailment

We show the complexity upper bounds for general queries shown in Figure 1, where we again focus on the complementary problem of query satisfiability.

The main idea is to define appropriate abstractions of models of the TPKB $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{i \in \llbracket 1, n \rrbracket} \rangle$ which we call *quasi-models*, and then show how an abstraction that witnesses the satisfiability can be guessed and verified within the targeted complexity bound. We first define the structure to represent single time points, which we call *quasi-states*. We can assume without loss of generality that ϕ contains only the operators $\wedge, \neg, \mathcal{U}, \mathcal{S}$ and $\text{P}_{\sim p}$, since the remaining operators can be linearly encoded using known equivalences. Denote by $\text{sub}(\phi)$ the sub-queries of ϕ and set $T(\phi) = \{\psi, \neg\psi \mid \psi \in \text{sub}(\phi)\}$. A quasi-state is now a mapping $Q : \Omega_{\mathcal{K}} \rightarrow T(\phi)$ that satisfies the following conditions:

- S1** $\neg\psi \in Q(w)$ iff $\psi \notin Q(w)$,
- S2** for all $\psi_1 \wedge \psi_2 \in T(\phi)$: $\psi_1 \wedge \psi_2 \in Q(w)$ iff $\psi_1 \in Q(w)$ and $\psi_2 \in Q(w)$, and
- S3** for all $\text{P}_{\sim p}(\psi) \in T(\phi)$: $\text{P}_{\sim p}(\psi) \in Q(w)$ iff $\mu_{\mathcal{K}}(\{w \mid \psi \in Q(w)\}) \sim p$.

The quasi-state abstracts probabilistic interpretations at a single time point by assigning queries to each possible world according to the semantics of the atemporal operators in our query language. To incorporate the temporal dimension, we consider unbounded sequences of quasi-states $(Q_i)_{i \geq 1}$, which we call *quasi-models for \mathcal{K}* , and which have to satisfy the following conditions for $i \geq 1$ and $w = (\mathcal{A}'_i)_{i \in \llbracket 1, n \rrbracket} \in \Omega_{\mathcal{K}}$:

- Q1** if $i \in \llbracket 1, n \rrbracket$, then $\langle \mathcal{T}, \mathcal{A}'_i \rangle \not\models \neg \left(\bigwedge_{\psi \in X} \psi \right)$, where $X = \{\psi \in Q_i(w) \mid \psi \text{ is a CQ or a negated CQ}\}$.
- Q2** for all $\circ\psi \in T(\phi)$, $\circ\psi \in Q_i(w)$ iff $\psi \in Q_{i+1}(w)$,
- Q3** for all $\circ^-\psi \in T(\phi)$, $\circ^-\psi \in Q_{i+1}(w)$ iff $\psi \in Q_i(w)$,
- Q4** for all $\psi_1 \mathcal{U} \psi_2 \in T(\phi)$, $\psi_1 \mathcal{U} \psi_2 \in Q_i$ iff there exists $j \geq i$ s.t. $\psi_2 \in Q_j(w)$ and for all $k \in \llbracket i, j-1 \rrbracket$, $\psi_1 \in Q_k(w)$, and
- Q5** for all $\psi_1 \mathcal{S} \psi_2 \in T(\phi)$, $\psi_1 \mathcal{S} \psi_2 \in Q_i$ iff there exists $j \leq i$ s.t. $\psi_2 \in Q_j(w)$ and for all $k \in \llbracket j-1, i \rrbracket$, $\psi_1 \in Q_k(w)$.

Again, the intuition behind these conditions is given directly by the semantics of the temporal operators.

To handle rigid names, we need an additional structure to make sure that the queries assigned to different time points in a possible world correspond to a sequence of interpretations that respects rigid names. Let $\{q_1, \dots, q_n\}$ be the CQs that occur in the query ϕ . For each $w \in \Omega_{\mathcal{K}}$, we guess the set $\mathcal{S}(w) \subseteq 2^{\{q_1, \dots, q_m\}}$ of *sets of queries* that are allowed to be satisfied together at a time point in w , and thus obtain a mapping $\mathcal{S} : \Omega_{\mathcal{K}} \rightarrow 2^{2^{\{q_1, \dots, q_m\}}}$. To be consistent with the rigid names, $\mathcal{S}(w)$ has to correspond to a set of interpretations that agree on the rigid names, where each set of queries corresponds to one interpretation. To also take into account the ABoxes, we use

a second mapping $a : \Omega_{\mathcal{K}} \times \llbracket 1, n \rrbracket \rightarrow 2^{\{q_1, \dots, q_n\}}$, which for each $w \in \Omega_{\mathcal{K}}$ assigns elements from $\mathcal{S}(w)$ to the ABoxes in w . Given such mappings \mathcal{S} and a , we say that a quasi-model $(Q_i)_{i \geq 1}$ is *compatible to \mathcal{S} and a* if for every $i \geq 0$ and $w \in \Omega_{\mathcal{K}}$:

Q6 $Q_i(w) \cap \{q_1, \dots, q_m\} \in \mathcal{S}(w)$, and

Q7 if $i \in \llbracket 1, n \rrbracket$, $Q_i(w) \cap \{q_1, \dots, q_m\} = a(w, i)$.

The following definition captures when \mathcal{S} and a correspond to a model of \mathcal{K} that respects rigid names.

Definition 5. Let $w = (\mathcal{A}'_i)_{i \in \llbracket 1, n \rrbracket} \in \Omega_{\mathcal{K}}$, $\mathcal{S} : \Omega_{\mathcal{K}} \rightarrow 2^{\{q_1, \dots, q_m\}}$ and $a : \Omega_{\mathcal{K}} \times \llbracket 1, n \rrbracket \rightarrow 2^{\{q_1, \dots, q_m\}}$, where $\mathcal{S}(w) = \{X_1, \dots, X_k\}$. Then, \mathcal{S} is called *r-satisfiable wrt. w and a* iff there exist (classical) interpretations $\mathcal{J}_1, \dots, \mathcal{J}_k, \mathcal{I}_1, \dots, \mathcal{I}_n$ such that

R1 the interpretations are models of \mathcal{T} ,

R2 for any two interpretations $\mathcal{I}, \mathcal{I}' \in \{\mathcal{J}_1, \dots, \mathcal{J}_k, \mathcal{I}_1, \dots, \mathcal{I}_n\}$, we have $\Delta^{\mathcal{I}} = \Delta^{\mathcal{I}'}$ and $X^{\mathcal{I}} = X^{\mathcal{I}'}$ for all $X \in \mathbf{N}_{\text{rig}}$,

R3 for all $i \in \llbracket 1, k \rrbracket$, $\mathcal{J}_i \models \bigwedge_{q \in X_i} q \wedge \bigwedge_{q \notin X_i} \neg q$, and

R4 for all $i \in \llbracket 1, n \rrbracket$, $\mathcal{I}_i \models \bigwedge_{q \in a(w, i)} q \wedge \bigwedge_{q \notin a(w, i)} \neg q$ and $\mathcal{I}_i \models \mathcal{A}'_i$.

\mathcal{S} is r-satisfiable wrt. a , if for all $w \in \Omega_{\mathcal{K}}$, \mathcal{S} is r-satisfiable wrt. w and a .

Note that the interpretations $\mathcal{J}_1, \dots, \mathcal{J}_k$ in the interpretation correspond to the elements $\{X_1, \dots, X_k\} = \mathcal{S}(w)$, so that Condition **R2** ensures that we can find sequences of interpretation that respect rigid names.

Lemma 6. *Wrt. the size of \mathcal{K} and ϕ , r-satisfiability for \mathcal{L} -TPKBs can be decided in*

1. NEXPTIME for $\mathcal{L} = \mathcal{ELH}$,
2. NEXPTIME for $\mathcal{L} = \mathcal{SHQ}$ if $\mathbf{N}_{\text{Rig}} = \emptyset$,
3. 2-EXPTIME for $\mathcal{L} \in \{\mathcal{SHIQ}, \mathcal{SHOQ}, \mathcal{SHOI}\}$, and
4. it is decidable for $\mathcal{L} = \mathcal{ALCHOIQ}$.

Proof(Sketch). We define a classical KB based on the mappings a and a world $w \in \Omega_{\mathcal{K}}$ which encodes the interpretation of non-rigid names $Y \in (\mathbf{N}_{\mathcal{C}} \cup \mathbf{N}_{\mathcal{R}}) \setminus \mathbf{N}_{\text{rig}}$ for different elements $X_i \in \mathcal{S}_i(w)$ using fresh names Y^i . A similar translation is applied to the CQs $q \in X_i$. We can then reduce the properties in Definition 5 to a query entailment problem, where the KB and the query are of exponential size with respect to the input. While query entailment for \mathcal{ALCI} and \mathcal{ALCO} is 2-EXPTIME-hard [27, 30], we obtain by inspection of the procedures in [22, 21, 11] that this particular query entailment test can be performed in 2-EXPTIME. For \mathcal{SHQ} and $\mathbf{N}_{\text{Rig}} = \emptyset$, the complexity follows from results in [3]. \square

Quasi-models are indeed sufficient to witness the satisfiability of a TPQ. If the quasi-model is additionally compatible to mappings \mathcal{S} and a s.t. \mathcal{S} is r-satisfiable wrt. a , then they witness the satisfiability of a TPQ from TKBs with rigid names. Crucially for our complexity result, it is further sufficient to focus on quasi-models that have a regular shape.

Lemma 7. ϕ is satisfiable in \mathcal{K} at time point i iff there exists mappings $\mathcal{S} : \Omega_{\mathcal{K}} \rightarrow 2^{2^{\{q_1, \dots, q_m\}}}$ and $a : \Omega_{\mathcal{K}} \times \llbracket 1, n \rrbracket \rightarrow 2^{\{q_1, \dots, q_m\}}$ and a quasi-model $(Q_j)_{j \geq 1}$ for \mathcal{K} such that

1. $\phi \in Q_i(w)$ for all $w \in \Omega_{\mathcal{K}}$,
2. $(Q_j)_{j \geq 1}$ is compatible with \mathcal{S} and a ,
3. \mathcal{S} is r -satisfiable wrt. to a , and
4. $(Q_j)_{j \geq 1}$ is of the form $Q_1, \dots, Q_m(Q_{m+1}, \dots, Q_{m+o})^\omega$, where m and o are both double exponentially bounded in the size of \mathcal{K} .

Theorem 8. The complexity upper bounds for general TPQs in Figure 1 hold.

Proof (Sketch). We first guess the numbers m and o from Lemma 7. If $\mathbf{N}_{\text{rig}} \neq \emptyset$, we additionally guess the mappings \mathcal{S} and a and verify that \mathcal{S} is r -satisfiable wrt. a . We now guess the quasi-states Q_1, \dots, Q_{m+o} one after the other, where we carefully make sure that all the conditions in the definition of quasi-states and quasi-models are satisfied, and verify that Q_{m+o} is compatible to Q_{m+1} . This procedure runs in exponential space if r -satisfiability can be decided in exponential space, and in double exponential time if deciding r -satisfiability requires double exponential time. \square

6 Positive TPQs

It turns out that for \mathcal{EL} , we can obtain better complexity bounds if we restrict ourselves to *positive TPQs*, which are TPQs that do not use the operators \neg , $P_{<p}$ and $P_{=p}$. The probability operators $P_{<p}$ and $P_{=p}$ can be seen as implicit negation operators, as they express the non-entailment of a query ϕ in some possible worlds, whereas $P_{>p}\phi$ only expresses the positive entailment of ϕ in some possible worlds. The examples used in this paper all use only positive TPQs.

Definition 9. A TPQ is *positive* iff it does not use the operators \neg , $P_{<p}$, $P_{\leq p}$ and $P_{=p}$.

For DLs that have negation, our reduction used to show EXPSPACE-hardness can be adapted to query entailment for positive TPQs. As we reduced the corridor tiling problem to query satisfiability, the corresponding query entailment problem is of the form $\mathcal{K} \models \neg\phi$, where ϕ is the defined query. By pushing negations inside, we obtain a query in which every probability operator is of the form $P_{>0}$ or $P_{\geq 1}$, and negation only occurs in front of concept names. Therefore, for any DL extending \mathcal{ALC} , the complexity bounds established in the last sections remain tight even for positive TPQs. In contrast to \mathcal{ALC} , \mathcal{EL} has the canonical model property, which makes it possible to test for entailment in different possible worlds independently. This allows for a strategy in which the TPQ is evaluated “inside out”, by first evaluating the most nested probability operators, and then proceeding on the next level. Due to the known closure properties of the complexity class PP, we obtain a PP^{NP} complexity upper bound if the nesting depth of the probability operators is bound, which we show to be tight, and otherwise a P^{NP} upper bound. This approach further allows us to establish tight complexity for *data complexity*, where the size of the query is assumed to be fix, marked in Figure 1 with (pos,dat).

Theorem 10. The complexity results regarding positive TPQs in Figure 1 hold.

7 Conclusion

We introduced a framework for representing and querying temporal probabilistic data within the ontology-based query answering paradigm, and established tight complexity bounds for most common description logics. While for expressive DLs starting from \mathcal{ALCI} and \mathcal{ALCO} , adding both the temporal and the probabilistic dimension comes at no additional cost compared to classical query answering, for \mathcal{ALC} and below, reasoning becomes harder both in comparison to purely temporal and purely probabilistic query answering. For instance, probabilistic query answering is EXPTIME-complete for \mathcal{ALC} and PP^{NP} -complete for \mathcal{EL} , and for $\text{N}_{\text{Rig}} = \emptyset$, it is EXPTIME-complete for \mathcal{ALC} and PSPACE-complete for \mathcal{EL} , which contrasts with our EXPSPACE-hardness that occurs already without a TBox. For \mathcal{EL} , this situation can be improved if we forbid negation in the query language, in which case temporal probabilistic query answering is not harder as in the atemporal case. We believe that our technique for showing the upper bound here could also be used for practical implementations. We are currently looking at how query rewriting techniques for simpler DLs such as *DL-Lite* could be used for this in connection with existing probabilistic database systems.

8 Acknowledgements

This work is supported by the German Research Foundation (DFG) within the Collaborative Research Center SFB 912 HAEC.

References

- [1] Alessandro Artale, Roman Kontchakov, Alisa Kovtunova, Vladislav Ryzhikov, Frank Wolter, and Michael Zakharyashev. Ontology-mediated query answering over temporal data: A survey (invited talk). In *Proc. TIME 2017*, pages 1:1–1:37, 2017.
- [2] Franz Baader, Stefan Borgwardt, Patrick Koopmann, Ana Ozaki, and Veronika Thost. Metric temporal description logics with interval-rigid names. In *Proc. FroCoS 2017*, pages 60–76. Springer International, 2017.
- [3] Franz Baader, Stefan Borgwardt, and Marcel Lippmann. Temporal query entailment in the description logic \mathcal{SHQ} . *J. Web Sem.*, 33:71–93, 2015.
- [4] Franz Baader, Silvio Ghilardi, and Carsten Lutz. LTL over description logic axioms. *ACM Trans. Comput. Log.*, 13(3):21:1–21:32, 2012.
- [5] Franz Baader, Patrick Koopmann, and Anni-Yasmin Turhan. Using ontologies to query probabilistic numerical data. In *Proc. FroCoS 2017*, pages 77–94. Springer International, 2017.
- [6] Stefan Borgwardt, İsmail İlkan Ceylan, and Thomas Lukasiewicz. Ontology-mediated queries for probabilistic databases. In *Proc. AAAI 2017*, pages 1063–1069. AAAI Press, 2017.
- [7] Stefan Borgwardt, Marcel Lippmann, and Veronika Thost. Temporalizing rewritable query languages over knowledge bases. *J. Web Sem.*, 33:50–70, 2015.
- [8] Stefan Borgwardt and Veronika Thost. Temporal query answering in the description logic \mathcal{EL} . In *Proc. IJCAI 2015*, pages 2819–2825. AAAI Press, 2015.

- [9] Camille Bourgaux, Patrick Koopmann, and Anni-Yasmin Turhan. Ontology-mediated query answering over temporal and inconsistent data. *Semantic Web Journal*, 2018. To appear.
- [10] Sebastian Brandt, Elem Güzel Kalayci, Roman Kontchakov, Vladislav Ryzhikov, Guohui Xiao, and Michael Zakharyashev. Ontology-based data access with a Horn fragment of metric temporal logic. In *Proc. AAI 2017*, pages 1070–1076. AAAI Press, 2017.
- [11] Diego Calvanese, Thomas Eiter, and Magdalena Ortiz. Regular path queries in expressive description logics with nominals. In *Proc. IJCAI 2009*, pages 714–720. AAAI Press, 2009.
- [12] Diego Calvanese, Thomas Eiter, and Magdalena Ortiz. Answering regular path queries in expressive description logics via alternating tree-automata. *Inf. Comput.*, 237:12–55, 2014.
- [13] İsmail İlkan Ceylan, Adnan Darwiche, and Guy van den Broeck. Open-world probabilistic databases: An abridged report. In *Proc. IJCAI 2017*, pages 4796–4800. AAAI Press, 2017.
- [14] İsmail İlkan Ceylan and Rafael Peñaloza. Dynamic Bayesian description logics. In *Proc. DL 2015*. CEUR-WS, 2015.
- [15] İsmail İlkan Ceylan and Rafael Peñaloza. The Bayesian ontology language *BEL*. *J. Autom. Reasoning*, 58(1):67–95, 2017.
- [16] Nilesh N. Dalvi and Dan Suciu. Efficient query evaluation on probabilistic databases. *VLDB J.*, 16(4):523–544, 2007.
- [17] Maximilian Dylla, Iris Miliaraki, and Martin Theobald. A temporal-probabilistic database model for information extraction. *Proceedings of the VLDB Endowment*, 6(14):1810–1821, 2013.
- [18] Peter L Elkin, Steven H Brown, Casey S Husser, Brent A Bauer, Dietlind Wahner-Roedler, S Trent Rosenbloom, and Ted Speroff. Evaluation of the content coverage of SNOMED CT: ability of SNOMED clinical terms to represent clinical problem lists. *Mayo Clin. Proc.*, 81(6):741–748, 2006.
- [19] Lance Fortnow and Nick Reingold. PP is closed under truth-table reductions. In *Proceedings of the Sixth Annual Structure in Complexity Theory Conference*, pages 13–15, 1991.
- [20] Dov M Gabbay, Agi Kurucz, Frank Wolter, and Michael Zakharyashev. *Many-dimensional modal logics: Theory and applications*. Elsevier, 2003.
- [21] Birte Glimm, Ian Horrocks, and Ulrike Sattler. Unions of conjunctive queries in *SHOQ*. In *Proc. KR 2008*, pages 252–262, 2008.
- [22] Birte Glimm, Carsten Lutz, Ian Horrocks, and Ulrike Sattler. Conjunctive query answering for the description logic *SHIQ*. *J. Artif. Intell. Res.*, 31:157–204, 2008.
- [23] Víctor Gutiérrez-Basulto, Jean Christoph Jung, Carsten Lutz, and Lutz Schröder. Probabilistic description logics for subjective uncertainty. *J. Artif. Intell. Res.*, 58:1–66, 2017.
- [24] Jean Christoph Jung and Carsten Lutz. Ontology-based access to probabilistic data with OWL QL. In *The Semantic Web - ISWC 2012*, volume 7649 of *LNCS*, pages 182–197. Springer, 2012.
- [25] Nilay Kumar, Monica Khunger, Arjun Gupta, and Neetika Garg. A content analysis of smartphone-based applications for hypertension management. *Journal of the American Society of Hypertension*, 9(2):130–136, 2015.
- [26] Thomas Lukasiewicz. Expressive probabilistic description logics. *Artif. Intell.*, 172(6-7):852–883, 2008.

- [27] Carsten Lutz. Inverse roles make conjunctive queries hard. In *Proc. DL 2007*, volume 250 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.
- [28] Carsten Lutz. Two upper bounds for conjunctive query answering in *SHIQ*. In *Proc. DL 2008*, volume 353 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.
- [29] Alessandro Margara, Jacopo Urbani, Frank van Harmelen, and Henri Bal. Streaming the web: Reasoning over dynamic data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 25:24 – 44, 2014.
- [30] Nhung Ngo, Magdalena Ortiz, and Mantas Simkus. Closed predicates in description logics: Results on combined complexity. In *Proc. KR 2016*, pages 237–246. AAAI Press, 2016.
- [31] Alessandro Ronca, Mark Kaminski, Bernardo Cuenca Grau, Boris Motik, and Ian Horrocks. Stream reasoning in temporal datalog. In *Proc. AAAI 2018*. AAAI Press, 2018.
- [32] Riccardo Rosati. On conjunctive query answering in \mathcal{EL} . In *Proc. DL 2007*, volume 250 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.
- [33] Seinosuke Toda. PP is as hard as the polynomial-time hierarchy. *SIAM J. Comput.*, 20(5):865–877, 1991.
- [34] Peter van Emde Boas. The convenience of tilings. *Lecture Notes in Pure and Applied Mathematics*, pages 331–363, 1997.
- [35] Klaus W. Wagner. The complexity of combinatorial problems with succinct input representation. *Acta Informatica*, 23(3):325–356, Jun 1986.

A Proofs of Theorems and Lemmata

We provide proofs for the theorems and lemmata in the paper. Note that this document has an extended bibliography section at the end.

A.1 Lower Bounds

Theorem 4. *The lower bounds regarding general TPQs in Figure 1 hold.*

Proof. For DLs extending \mathcal{ALCI} or \mathcal{ALCO} , 2-EXPTIME lower bounds follow from the corresponding results in classical query entailment [27, 30].

For the remaining DLs, we provide a reduction of the EXPSPACE-complete 2^n corridor tiling problem as specified in the main text. We provide an encoding of this problem using a single ipABox \mathcal{A} and a TPCQ ϕ with negations. \mathcal{A} and ϕ are constructed in such a way that there is a correspondence between solutions f to the tiling problem and models ι of $\mathcal{K} = \langle \emptyset, (\mathcal{A}) \rangle$ and ϕ . The (bounded) vertical dimension of the corridor is represented across the 2^n possible worlds, while the (unbounded) horizontal dimension is represented along the time line. Specifically, the correspondence from ι to a tiling f is specified via two mappings:

1. $y : \llbracket 0, 2^n - 1 \rrbracket \rightarrow \Omega_{\mathcal{K}}$ maps each y -coordinate of the tiling to a possible world
2. $c : \mathbb{N} \times \Omega_{\mathcal{K}} \rightarrow T$ maps each pair of x -coordinate and possible world to the tile type.

The query will then be constructed in such a way that, in case of satisfiability, we can obtain the tiling by $f(i, j) = c(2^n \cdot j + i, y(j))$.

We use a concept name B_t for every tile type $t \in T$, and use the assertion $B_t(a)$ to denote that the cell corresponding to a possible world/time point pair has a tile of type t . To make sure that every pair of a possible world and a time point represents exactly one $t \in T$, we use the query

$$\square \bigwedge_{t_1 \in T} \left(B_{t_1}(a) \leftrightarrow \bigwedge_{t_2 \in T, t_1 \neq t_2} \neg B_{t_2}(a) \right).$$

We can already define the first mapping c from pairs of possible worlds and time points to their tile type: given a model ι of the final query, we define $c(i, w) = t$, where $\iota, w, i \models B_t(a)$.

To provide for the mapping $y : [0, 2^n - 1] \rightarrow \Omega_{\mathcal{K}}$, which assigns row numbers to possible worlds, we set

$$\mathcal{A}_1 = \{A_i(a) : 0.5 \mid i \in \llbracket 1, n \rrbracket\},$$

where A_1, \dots, A_n are concept names that correspond to bits in a binary counter. Since all probabilities are statistically independent, each counter value is represented by some possible world. For $i \in [0, 2^n - 1]$, the value of $y(i)$ is then simply the possible world in which the counter has the value i at the first time point. We mark the possible world which represents the last row with the assertion $L(a)$ using the following queries.

$$\begin{aligned} & \left(\bigwedge_{i \in \llbracket 1, n \rrbracket} A_i(a) \right) \leftrightarrow L(a) \\ & \square (L(a) \leftrightarrow \circ L(a)) \end{aligned}$$

The previous queries ensure that the mappings y and c are well-defined. The following query ensures the tiling conditions regarding the special tiles t_s and t_e , which have to occur in the first row of respectively the first and the last column of the tiling solution.

$$\left(\bigwedge_{i \in \llbracket 1, n \rrbracket} \neg A_i(a) \right) \rightarrow (B_{t_s}(a) \wedge \diamond B_{t_e}(a))$$

It remains to provide queries that enforce the tiling conditions. The following queries ensure that the counters in each possible world get incremented across the time line.

$$\begin{aligned} & \square \bigwedge_{i \in \llbracket 1, n \rrbracket} \left(\left(\neg A_i(a) \wedge \bigwedge_{j < i} A_j(a) \right) \rightarrow \circ \left(A_i(a) \wedge \bigwedge_{j < i} \neg A_j(a) \right) \right) \\ & \square \bigwedge_{i \in \llbracket 1, n \rrbracket} \left(\left(\neg A_i(a) \wedge \bigvee_{j < i} \neg A_j(a) \right) \rightarrow \circ \neg A_i(a) \right) \\ & \square \bigwedge_{i \in \llbracket 1, n \rrbracket} \left(\left(A_i(a) \wedge \bigvee_{j < i} \neg A_j(a) \right) \rightarrow \circ A_i(a) \right) \\ & \square \left(\left(\bigwedge_{i \in \llbracket 1, n \rrbracket} A_i(a) \right) \rightarrow \circ \bigwedge_{i \in \llbracket 1, n \rrbracket} \neg A_i(a) \right) \end{aligned}$$

In each possible world, the current tile type is transported to the next time point until until the counter reaches $2^n - 1$.

$$\square \bigwedge_{t \in T} \left(\left(B_t(a) \wedge \bigvee_{i \in [1, n]} \neg A_i(a) \right) \rightarrow \circ B_t(a) \right)$$

Note that we only have to check the tiling conditions until the special tile t_e has been placed in the first row: afterwards, we do not care. We use the assertion $B_{end}(a)$ to mark the last column of the tiling:

$$\left(\bigwedge_{i \in [1, n]} \neg A_i(a) \right) \rightarrow (\mathsf{P}_{=1}(\neg B_{end}(a)) \mathcal{U}(B_{end}(a) \wedge t_e(a)))$$

This query ensures that no possible world satisfies $B_{end}(a)$ until the last tile has been placed in the first row.

As explained in the main text, if the counter in a possible world reaches $2^n - 1$, we can identify the world that corresponds to the next row easily, as its counter then has the value 0. Note that we only have to enforce the tiling conditions until the last tile has been placed, which is the case as soon as in some possible world the assertion $B_{end}(a)$ is satisfied. We can thus enforce the vertical tiling conditions using the following query.

$$\begin{aligned} & \bigwedge_{t_1 \in T} \left(\left(\left(B_{t_1}(a) \wedge \neg L(a) \wedge \bigwedge_{i \in [1, n]} A_i(a) \right) \right. \right. \\ & \left. \left. \rightarrow \bigvee_{t_2 \in v(t_1)} \mathsf{P}_{=1} \left(\left(\bigwedge_{i \in [1, n]} \neg A_i(a) \right) \rightarrow B_{t_2}(a) \right) \right) \right) \mathcal{U} \mathsf{P}_{>0}(B_{end}(a)) \end{aligned}$$

To enforce the horizontal constraints, we only have to identify the next time point when the counter is $2^n - 1$.

$$\bigwedge_{t_1 \in T} \left(\left(\left(B_{t_1}(a) \wedge \bigwedge_{i \in [1, n]} A_i(a) \right) \rightarrow \circ \bigvee_{t_2 \in h(t_1)} B_{t_2}(a) \right) \right) \mathcal{U} \mathsf{P}_{>0}(B_{end}(a))$$

The final query ϕ is the conjunction of all queries. It is now standard to verify that the tiling problem has a solution f iff ϕ is satisfiable in \mathcal{A} , and that this solution is then obtained by $f(i, j) = c(2^n \cdot j + i, y(j))$. \square

A.2 Correctness of the Decision Procedure

We first show that our notion of quasi-models compatible to the mappings \mathcal{S} and a indeed captures query satisfiability.

Lemma 11. *ϕ is satisfiable in \mathcal{K} at i iff there exist mappings $\mathcal{S} : \Omega_{\mathcal{K}} \rightarrow 2^{2^{\{q_1, \dots, q_m\}}}$ and $a : \Omega_{\mathcal{K}} \times [1, n] \rightarrow 2^{\{q_1, \dots, q_m\}}$ s.t.*

1. \mathcal{S} is r -satisfiable wrt. a , and

2. there exists a quasi-model $(Q_j)_{j \geq 1}$ for ϕ in \mathcal{K} compatible with \mathcal{S} and a s.t. $\phi \in Q_i(w)$ for all $w \in \Omega_{\mathcal{K}}$.

We show both directions of the lemma separately.

Lemma 12. *If ϕ is satisfiable in \mathcal{K} at i , then there exist mappings $\mathcal{S} : \Omega_{\mathcal{K}} \rightarrow 2^{2^{\{q_1, \dots, q_m\}}}$ and $a : \Omega_{\mathcal{K}} \times \llbracket 1, n \rrbracket \rightarrow 2^{\{q_1, \dots, q_m\}}$ s.t.*

1. for every $w \in \Omega_{\mathcal{K}}$, \mathcal{S} is r -satisfiable wrt. a and w , and
2. there exists a quasi-model $(Q_j)_{j \geq 1}$ for ϕ in \mathcal{K} compatible with \mathcal{S} and a s.t. $\phi \in Q_i(w)$ for all $w \in \Omega_{\mathcal{K}}$.

Proof. Assume ϕ is satisfiable in \mathcal{K} at i . There then exists a temporal probabilistic model ι of \mathcal{K} s.t. for all $w \in \Omega_{\mathcal{K}}$, $\iota, w, i \models \phi$. For $w \in \Omega_{\mathcal{K}}$ and $j \geq 1$, set $X_j(w) = \{q \text{ is a CQ in } \phi \mid \iota(w)_i \models q_i\}$. \mathcal{S} and a are now defined by setting for all $w \in \Omega_{\mathcal{K}}$ and $i \in \llbracket 1, n \rrbracket$:

$$\begin{aligned} \mathcal{S}(w) &= \{X_j(w) \mid j \geq 1\} \\ a(w, i) &= X_i(w). \end{aligned}$$

It follows by construction that for each $w \in \mathcal{S}$, \mathcal{S} is r -satisfiable wrt. a and w . Indeed, the interpretations $\mathcal{I}_1, \dots, \mathcal{I}_n, \mathcal{J}_1, \dots, \mathcal{J}_k$ that witness this can be directly taken from ι : for $i \in \llbracket 1, n \rrbracket$, we have $\mathcal{I}_i = \iota(w)_i$, and for each $X_i \in \mathcal{S}(w)$, there exists some $j \geq 1$ s.t. $X_j(w) = X_i$, for which we set $\mathcal{J}_i = \iota(w)_j$. Conditions **R1–R4** are now readily checked.

The quasi-model $(Q_j)_{j \geq 1}$ is now defined by setting

$$Q_j(w) = \{\psi \in T(\phi) \mid \iota, w, j \models \psi\}$$

for all $w \in \Omega_{\mathcal{K}}$ and $i > 1$.

We show that $(Q_j)_{j \geq 1}$ is indeed a quasi-model compatible with \mathcal{S} and a .

For each $j \geq 1$, it follows by direct correspondence between the semantics of the atemporal query operators defined in Table 2 to Conditions **S1–S3** that Q_j is a quasi-state. Specifically, for Condition **S3**, we have $\iota, w, j \models P_{sim}(\psi)$ iff

$$\mu_{\mathcal{K}}(\{w' \in \Omega_{\mathcal{K}} \mid \iota, w', j \models \psi\}) \sim p,$$

and consequently, by induction, $P_{\sim p}(\psi) \in Q_j(w)$ iff

$$\mu_{\mathcal{K}}(\{w' \in \Omega_{\mathcal{K}} \mid \psi \in Q_j(w)\}) \sim p.$$

Similarly, $(Q_j)_{j \geq 1}$ satisfies Conditions **Q2–Q5** by direct correspondence to the semantics of the temporal query operators defined in Table 2. For Condition **Q1**, we have to show that for $w = (\mathcal{A}'_i)_{i \in \llbracket i \rrbracket} \in \Omega_{\mathcal{K}}$ and $i \in \llbracket 1, n \rrbracket$,

$$\langle \mathcal{T}, \mathcal{A}'_i \rangle \not\models \neg \left(\bigwedge_{\psi \in X} \psi \right),$$

where

$$X = \{\psi \in Q_i(w) \mid \psi \text{ is a CQ or a negated CQ}\}.$$

Let $X = \{\psi \in Q_i(w) \mid \psi \text{ is a CQ or a negated CQ}\}$. By construction, $\iota, w, i \models \bigwedge_{\psi \in X} \psi$. Furthermore, since ι is a model of \mathcal{K} , $\iota(w)_i$ is a model of $\langle \mathcal{T}, \mathcal{A}'_i \rangle$. As a consequence, we cannot

have $\langle \mathcal{T}, \mathcal{A}'_i \rangle \models \neg \left(\bigwedge_{\psi \in X} \psi \right)$, and therefore $\langle \mathcal{T}, \mathcal{A}'_i \rangle \not\models \neg \left(\bigwedge_{\psi \in X} \psi \right)$ and $(Q_j)_{j \geq 1}$ satisfies Condition **Q1**. We obtain that $(Q_j)_{j \geq 1}$ satisfies Conditions **Q1–Q5**, and thus that $(Q_j)_{j \geq 1}$ is a quasi-model.

It remains to show that $(Q_j)_{j \geq 1}$ is compatible with \mathcal{S} and a , that is, that it satisfies Conditions **Q6** and **Q7**. Let $j \geq 1$ and $w \in \Omega_{\mathcal{K}}$.

Q6 We have to show that $Q_j(w) \cap \{q_1, \dots, q_m\} \in \mathcal{S}(w)$. Let $X = Q_j(w) \cap \{q_1, \dots, q_m\}$. By construction of $Q_j(w)$, X contains exactly the CQs from $\{q_1, \dots, q_m\}$ that are entailed by $\iota(w)_j$. Consequently, $X = X_j(w)$. We have $X_j(w) \in \mathcal{S}(w)$ by construction of $\mathcal{S}(w)$, and therefore, $X \in \mathcal{S}(w)$.

Q7 We have to show that if $j \in \llbracket 1, n \rrbracket$, then $Q_j(w) \cap \{q_1, \dots, q_m\} = a(w, j)$. Let $j \in \llbracket 1, n \rrbracket$ and $X = Q_j(w) \cap \{q_1, \dots, q_m\}$. By construction of $Q_j(w)$, X contains exactly the CQs from $\{q_1, \dots, q_m\}$ that are entailed by $\iota(w)_j$, and therefore, $X = X_j(w)$. By definition of a , we have $a(w, j) = X_j(w)$, and therefore $a(w, j) = X$.

We established that \mathcal{S} is r -satisfiable wrt. a , and that $(Q_j)_{j \geq 1}$ is a quasi-model compatible to \mathcal{S} and a . Furthermore, we have $\phi \in Q_i(w)$ for all $w \in \Omega_{\mathcal{K}}$, since by assumption, $\iota, w, i \models \phi$ for all $w \in \Omega_{\mathcal{K}}$. It follows that Conditions 1 and 2 from the Lemma are satisfied. \square

Lemma 13. *Let $\mathcal{S} : \Omega_{\mathcal{K}} \rightarrow 2^{2^{\{q_1, \dots, q_m\}}}$ and $a : \Omega_{\mathcal{K}} \times \llbracket 1, n \rrbracket \rightarrow 2^{\{q_1, \dots, q_m\}}$ be such that*

1. *for every $w \in \Omega_{\mathcal{K}}$, \mathcal{S} is r -satisfiable wrt. a and w , and*
2. *there exists a quasi-model $(Q_j)_{j \geq 1}$ for ϕ in \mathcal{K} compatible with \mathcal{S} and a s.t. $\phi \in Q_i(w)$ for all $w \in \Omega_{\mathcal{K}}$.*

Then, ϕ is satisfiable in \mathcal{K} at i .

Proof. Assume there exist the mappings \mathcal{S} and a , as well as a quasi-model $(Q_j)_{j \geq 1}$, as in the lemma. We construct a temporal probabilistic model ι of \mathcal{K} s.t. for all $w \in \Omega_{\mathcal{K}}$, $\iota, w, 1 \models \phi$. For all $w \in \Omega_{\mathcal{K}}$, assume $\mathcal{S}(w) = \{X_1^w, \dots, X_{k_w}^w\}$. By Definition 5 of r -satisfiability, for every $w \in \Omega_{\mathcal{K}}$, there exists interpretations $\mathcal{J}_1^w, \dots, \mathcal{J}_{k_w}^w, \mathcal{I}_1^w, \dots, \mathcal{I}_n^w$ such that

- R1** the interpretations are models of \mathcal{T} ,
- R2** for any two interpretations $\mathcal{I}', \mathcal{I}'' \in \{\mathcal{J}_1^w, \dots, \mathcal{J}_{k_w}^w, \mathcal{I}_1^w, \dots, \mathcal{I}_n^w\}$, we have $\Delta^{\mathcal{I}'} = \Delta^{\mathcal{I}''}$ and $X^{\mathcal{I}'} = X^{\mathcal{I}''}$ for all $X \in \mathbf{N}_{rig}$,
- R3** for all $i \in \llbracket 1, k \rrbracket$, $\mathcal{J}_i^w \models \bigwedge_{q \in X_i^w} q \wedge \bigwedge_{q \notin X_i^w} \neg q$, and
- R4** for all $i \in \llbracket 1, n \rrbracket$, $\mathcal{I}_i^w \models \bigwedge_{q \in a(w, i)} q \wedge \bigwedge_{q \notin a(w, i)} \neg q$ and $\mathcal{I}_i^w \models \mathcal{A}'_i$.

ι is now defined by setting for all $w \in \Omega_{\mathcal{K}}$ and $j \geq 1$:

1. if $j \in \llbracket 1, n \rrbracket$: $\iota(w)_i = \mathcal{I}_i^w$, and
2. for all $j > n$: $\iota(w)_i = \mathcal{J}_j^w$, where j is such that $Q_i(w) \cap \{q_1, \dots, q_m\} = X_j^w$ (by Condition **Q6**).

By Condition **R1**, ι respects rigid names. Furthermore, for every $w = (\mathcal{A}'_j)_{j \in \llbracket 1, n \rrbracket} \in \Omega_{\mathcal{K}}$ and $j \geq 1$, $\iota(w)_j \models \mathcal{T}$, and if $j \in \llbracket 1, n \rrbracket$, $\iota(w)_j \models \mathcal{A}'_j$. It follows that ι is a model of \mathcal{K} . It remains to show that ι satisfies ϕ at i . We do this by structural induction on ϕ and show for every $\psi \in T$, $w \in \Omega_{\mathcal{K}}$ and $j \geq 1$, $\iota(w)_j \models \psi$ iff $\psi \in Q_j(w)$. Since $\phi \in Q_i(w)$ for all $w \in \Omega_{\mathcal{K}}$, this proves that $\iota, i \models \phi$. We distinguish the cases based on the syntactical shape of ψ .

1. ψ is a CQ. We have two cases. Either i) $j \in \llbracket 1, n \rrbracket$ and $\iota(w)_j = \mathcal{I}_j^w$, or ii) $j > n$ and $\iota(w) = \mathcal{J}_{j'}^w$ for some $j' \in \llbracket 1, k \rrbracket$. In the first case, $\psi \in a(w, j)$ iff $\iota(w)_j \models \psi$ by Condition **R4**, and $\psi \in a(w, j)$ iff $\psi \in Q_j(w)$ by Condition **Q7**. Consequently, $\psi \in Q_j(w)$ iff $\iota(w)_j \models \psi$ and $\iota, w, j \models \psi$.
In the second case, by Condition **R3**, we have $\psi \in X_{j'}^w$ iff $\iota(w)_j \models \psi$. Furthermore, we have $Q_j(w) \cap \{q_1, \dots, q_m\} = X_{j'}^w$ by construction, and consequently $\psi \in Q_j(w)$ iff $\psi \in X_{j'}^w$. It follows that $\psi \in Q_j(w)$ iff $\iota(w)_j \models \psi$ and $\iota, w, j \models \psi$.
2. $\psi = \neg\psi'$. By inductive hypothesis, $\iota, w, j \models \psi'$ iff $\psi' \in Q_j(w)$. Consequently, by the semantics of negation and Condition **S1** $\iota, w, j \models \neg\psi'$ iff $\neg\psi' \in Q_j(w)$.
3. $\psi = \psi_1 \wedge \psi_2$. By inductive hypothesis, $\iota, w, j \models \psi_1$ iff $\psi_1 \in Q_j(w)$, and $\iota, w, j \models \psi_2$ iff $\psi_2 \in Q_j(w)$. By the semantics of conjunction and Condition **S3**, we obtain that $\iota, w, j \models \psi_1 \wedge \psi_2$ iff $\psi_1 \wedge \psi_2 \in Q_j(w)$.
4. $\psi = \circ\psi'$. By Condition **Q2**, $\circ\psi' \in Q_j(w)$ iff $\psi' \in Q_{j+1}(w)$. By inductive hypothesis, $\psi' \in Q_{j+1}(w)$ iff $\iota, w, j+1 \models \psi'$. By the semantics of the TPQs, $\iota, w, j+1 \models \psi'$ iff $\iota, w, j \models \circ\psi'$. Consequently, $\circ\psi' \in Q_j(w)$ iff $\iota, w, j \models \circ\psi'$. The case where $\psi = \circ^-\psi'$ is similar.
5. $\psi = \psi_1 \mathcal{U}\psi_2$. Assume $\psi_1 \mathcal{U}\psi_2 \in Q_j(w)$. By Condition **Q4**, there then exists $j' \geq j$ s.t. $\psi_2 \in Q_{j'}(w)$ and $\psi_1 \in Q_{k'}$ for all $k' \in \llbracket j, j' \rrbracket$. By the inductive hypothesis, we then also have $\iota, w, j' \models \psi_2$ and $\iota, w, k' \models \psi_2$ for all $k' \in \llbracket j, j' \rrbracket$, which, by the semantics of the until-operator, implies $\iota, w, j \models \psi_1 \mathcal{U}\psi_2$. For the other direction, assume $\iota, w, j \models \psi_1 \mathcal{U}\psi_2$. By the semantics of the until-operator, there then exists some $j' \geq j$ s.t. $\iota, w, j' \models \psi_2$ and $\iota, w, k' \models \psi_1$ for all $k' \in \llbracket j, j' \rrbracket$, which by Condition **Q4** implies that $\psi_1 \mathcal{U}\psi_2 \in Q_j(w)$. The case where $\psi = \psi_1 \mathcal{S}\psi_2$ is similar.

We obtain that for every $\psi \in T(\phi)$, $w \in \Omega$ and $j \geq 1$, $\iota, w, j \models \psi$ iff $\psi \in Q_j(w)$. Since $\phi \in Q_i(w)$ for all $w \in \Omega_{\mathcal{K}}$, this implies $\iota, i \models \phi$. Since ι is a model of \mathcal{K} , we obtain that ϕ is satisfiable in \mathcal{K} at i . \square

We can now prove Lemma 7 from the main text.

Lemma 7. *ϕ is satisfiable in \mathcal{K} at time point i iff there exists mappings $\mathcal{S} : \Omega_{\mathcal{K}} \rightarrow 2^{\{q_1, \dots, q_m\}}$ and $a : \Omega_{\mathcal{K}} \times \llbracket 1, n \rrbracket \rightarrow 2^{\{q_1, \dots, q_m\}}$ and a quasi-model $(Q_j)_{j \geq 1}$ for \mathcal{K} such that*

1. $\phi \in Q_i(w)$ for all $w \in \Omega_{\mathcal{K}}$,
2. $(Q_j)_{j \geq 1}$ is compatible with \mathcal{S} and a ,
3. \mathcal{S} is r -satisfiable wrt. to a , and
4. $(Q_j)_{j \geq 1}$ is of the form $Q_1, \dots, Q_m(Q_{m+1}, \dots, Q_{m+o})^\omega$, where m and o are both double exponentially bounded in the size of \mathcal{K} .

Proof. The direction (\Leftarrow) follows directly from Lemma 13. We therefore only have to show the other direction.

Let $\mathcal{S} : \Omega_{\mathcal{K}} \rightarrow 2^{\{q_1, \dots, q_m\}}$ and $a : \Omega_{\mathcal{K}} \times \llbracket 1, n \rrbracket \rightarrow 2^{\{q_1, \dots, q_m\}}$ be mappings and $(Q_j)_{j \geq 1}$ be a quasi-model for ϕ compatible with \mathcal{S} and a . First note that there can be at most double-exponentially many distinguishable quasi-states in $(Q_j)_{j \geq 1}$: $\Omega_{\mathcal{K}}$ contains at most $2^{|\mathcal{K}|}$ many elements, and for each $w \in \Omega_{\mathcal{K}}$, $Q_i(w)$ contains at most $|\phi|$ elements. We obtain that there are at most $2^{2^{|\mathcal{K}|} \cdot |\phi|}$ many different combinations. For indices j, k s.t. $Q_j = Q_k$, we define an operation *merging of Q_i and Q_j* in $(Q_i)_{i \geq 1}$, which replaces the quasi-model with $Q_1, \dots, Q_i, Q_{j+1}, \dots$. One can verify that the result of merging in a quasi-model is again a quasi-model: Condition **Q1** 1) Conditions **Q1–Q3** only consider at most two-subsequent states, and 2) Conditions **Q4** and **Q5** are still satisfied in the new quasi-model. Furthermore, since we only remove quasi-states, the resulting quasi-model also still satisfies Conditions **Q6** and **Q7**, and is thus compatible to \mathcal{S} and a .

Now let $(Q_j)_{j \geq 1}$ be any quasi-model, and let m, o be two indices s.t. $Q_m = Q_{m+o}$ and the following condition is satisfied:

(*) for every $w \in \Omega_{\mathcal{K}}$ and $\psi_1 \mathcal{U} \psi_2 \in Q_m(w)$, there exists $k < m + o$ s.t. $\psi_2 \in Q_k(w)$.

Such a quasi-model always exists by Condition **Q4**. From (*), it already follows that $Q_1, \dots, Q_m (Q_{m+1}, \dots, Q_{m+o})^\omega$ is also a quasi-model. However, m and o might not be double-exponentially bounded in the size of \mathcal{K} and ϕ . By the above observation, we may assume that no quasi-state occurs twice before Q_m , since we can always merge any quasi-states that occur more than once, so that $m \leq 2^{2^{|\mathcal{K}|} \cdot |\phi|}$. To reduce the index of Q_{m+o} , we exhaustively merge any two quasi-states that occur between Q_m and Q_{m+o} for which merging does not break Condition (*). The resulting quasi-state can now be represented as

$$Q'_1, \dots, Q'_m (Q'_{m+1}, \dots, Q'_{m+o'})^\omega.$$

We give a bound on o' . For every $i, j \in \llbracket n+1, n+o' \rrbracket$ s.t. $Q'_i = Q'_j$, there must be some $w \in \Omega_{\mathcal{K}}$, $\psi_1 \mathcal{U} \psi_2 \in Q'_n(w)$, and $k \in \llbracket i, j \rrbracket$ s.t. $\psi_2 \in Q'_k(w)$ and $\psi_2 \notin Q'_l(w)$ for all $l \in \llbracket n, k-1 \rrbracket$, since otherwise Q'_i and Q'_j would have been merged. It follows that every quasi-state is repeated at most $2^{|\mathcal{K}|} \cdot |\phi|$ times, because there are at most $2^{|\mathcal{K}|}$ possible worlds in $\Omega_{\mathcal{K}}$ and for each $w \in \Omega_{\mathcal{K}}$, at most $|\phi|$ queries of the form $\psi_1 \mathcal{U} \psi_2$ in $Q'_n(w)$. Because the number of distinct quasi-states is bounded by $2^{2^{|\mathcal{K}|} \cdot |\phi|}$, we obtain $o' \leq 2^{|\mathcal{K}|} \cdot |\phi| \cdot 2^{2^{|\mathcal{K}|} \cdot |\phi|}$, that is, o' is double-exponentially bounded in the size of \mathcal{K} and ϕ . It follows that we can transform any quasi-model into a quasi-model of the required form, and thus that a quasi-model exists iff there exists a regular quasi-model which is of the form as in the lemma. \square

A.3 Deciding r-Satisfiability

In order to decide r-satisfiability, we construct a classical KB $\mathcal{K}_{a,w} = \langle \mathcal{A}_{a,w}, \mathcal{T}_{a,w} \rangle$ based on $w = (\mathcal{A}'_i)_{i \in \llbracket 1, n \rrbracket}$, $\mathcal{S} = \{X_1, \dots, X_k\}$ and a , where for each non-rigid name $Y \in (\mathbf{N}_{\mathbf{C}} \cup \mathbf{N}_{\mathbf{R}}) \setminus \mathbf{N}_{\mathbf{rig}}$ and $X_i \in \mathcal{S}(w)$, we use a fresh name Y^i . For every $A(a)/r(a, b) \in \mathcal{A}'_i$, $\mathcal{A}_{a,w}$ contains the assertion $A^i(a)/r^i(a, b)$ if $A \notin \mathbf{N}_{\mathbf{Crig}}/r \notin \mathbf{N}_{\mathbf{Rrig}}$, and otherwise the assertion $A(a)/r(a, b)$. The TBox $\mathcal{T}_{a,w}$ contains for every axiom α and every $X_i \in \mathcal{S}(w)$ the axiom α^i , which is obtained from α by replacing every non rigid name Y by Y^i . We do a similar transformation for the queries $q \in X_i$ to obtain queries q^i .

Lemma 14. \mathcal{S} is r-satisfiable wrt. a and w iff the following entailment holds.

$$\mathcal{K}_{a,w} \not\models \neg \left(\bigwedge_{X_i \in \mathcal{S}(w)} \left(\bigwedge_{q \in X_i} q^i \wedge \bigwedge_{q \notin X_i} \neg q^i \right) \right).$$

Proof. (\Leftarrow) Assume the entailment holds, and let \mathcal{I} be a model of $\mathcal{K}_{a,w}$ s.t.

$$\mathcal{I} \models \left(\bigwedge_{X_i \in \mathcal{S}(w)} \left(\bigwedge_{q \in X_i} q^i \wedge \bigwedge_{q \notin X_i} \neg q^i \right) \right).$$

Based on \mathcal{I} , we construct the models $\{\mathcal{I}_1, \dots, \mathcal{I}_n, \mathcal{J}_1, \dots, \mathcal{J}_k\}$ as required by Definition 5. For $X_j \in \mathcal{S}(w)$, we define the interpretation \mathcal{J}_j

1. $\Delta^{\mathcal{J}_i} = \Delta^{\mathcal{I}}$,
2. for all $Y \in \mathbf{N}_{\text{rig}}$: $X^{\mathcal{J}_i} = Y^{\mathcal{I}}$, and
3. for all $Y \in (\mathbf{N}_{\mathbf{C}} \cup \mathbf{N}_{\mathbf{R}}) \setminus \mathbf{N}_{\text{rig}}$: $Y^{\mathcal{J}_i} = (Y^j)^{\mathcal{J}}$.

For $i \in \llbracket 1, n \rrbracket$ and $a(w, i) = X_j$, we define $\mathcal{I}_i = \mathcal{J}_j$. It is now standard to verify the conditions in Definition 5:

R1: by the construction of the $\mathcal{T}_{a,w}$, every \mathcal{J}_i is a model of \mathcal{T} ,

R2: rigid names are interpreted the same for all interpretations,

R3: for all $i \in \llbracket 1, k \rrbracket$, $\mathcal{I} \models \bigwedge_{q \in X_i} q^i \wedge \bigwedge_{q \notin X_i} \neg q^i$, and consequently, $\mathcal{J}_i \models \bigwedge_{q \in X_i} q \wedge \bigwedge_{q \notin X_i} \neg q$,

R4: for $i \in \llbracket 1, n \rrbracket$, $\mathcal{I}_i \models \bigwedge_{q \in X_i} q \wedge \bigwedge_{q \notin X_i} \neg q$ for the same reason as in the last case, and since $\mathcal{I} \models \alpha^i$ for all $\alpha \in \mathcal{A}'_i$, $\mathcal{I}_i \models \mathcal{A}'_i$.

We obtain that \mathcal{S} is r-satisfiable wrt. a and w .

(\Rightarrow) Now for the other direction, assume \mathcal{S} is r-satisfiable wrt. a and w , and let $\mathcal{I}_1, \dots, \mathcal{I}_n, \mathcal{J}_1, \dots, \mathcal{J}_k$ be the interpretations from Definition 5. Note that, since $a(w, i) \in \mathcal{S}(w)$ for all $i \in \llbracket 1, n \rrbracket$, we may assume without loss of generality that $\{\mathcal{I}_1, \dots, \mathcal{I}_n\} \subseteq \{\mathcal{J}_1, \dots, \mathcal{J}_k\}$: if there exists interpretations $\mathcal{I}_1, \dots, \mathcal{I}_n, \mathcal{J}_1, \dots, \mathcal{J}_k$ satisfying the conditions in Definition 5, then there also exist such interpretation so that $\{\mathcal{I}_1, \dots, \mathcal{I}_n\} \subseteq \{\mathcal{J}_1, \dots, \mathcal{J}_k\}$. We may further assume without loss of generality that for $i \in \llbracket 1, n \rrbracket$, $\mathcal{I}_i = \mathcal{J}_i$.

We construct an interpretation \mathcal{I} as follows.

1. $\Delta^{\mathcal{I}} = \Delta^{\mathcal{J}_1}$,
2. for all $Y \in \mathbf{N}_{\text{rig}}$, $Y^{\mathcal{I}} = X^{\mathcal{J}_1}$, and
3. for all $Y \in (\mathbf{N}_{\mathbf{C}} \cup \mathbf{N}_{\mathbf{R}}) \setminus \mathbf{N}_{\text{rig}}$ and $i \in \llbracket 1, k \rrbracket$, $(Y^i)^{\mathcal{I}} = Y^{\mathcal{J}_i}$.

Note that by Condition **R2**, the specific choice of an interpretation in Step 1 and 2 does not affect the result. We show that \mathcal{I} is a model of $\mathcal{K}_{a,w}$ s.t.

$$\mathcal{I} \models \left(\bigwedge_{X_i \in \mathcal{S}(w)} \left(\bigwedge_{q \in X_i} q^i \wedge \bigwedge_{q \notin X_i} \neg q^i \right) \right).$$

By Condition **R1**, for every $j \in \llbracket 1, k \rrbracket$ and every $\alpha \in \mathcal{T}$, $\mathcal{J}_j \models \alpha$, and consequently, by construction of \mathcal{I} , one easily sees that $\mathcal{I} \models \alpha^j$. It follows that \mathcal{I} is a model of $\mathcal{T}_{a,w}$. By Condition **R4**, for every $i \in \llbracket 1, n \rrbracket$ and $\alpha \in \mathcal{A}'_i$, we have $\mathcal{J}_i = \mathcal{I}_i \models \alpha$, which similarly by construction implies $\mathcal{I} \models \alpha^i$. It follows that \mathcal{I} is a model of $\mathcal{K}_{a,w}$. It remains to show that for the queries $q \in \{q_1, \dots, q_m\}$, and for every $i \in \llbracket 1, k \rrbracket$, $\mathcal{I} \models q^i$ iff $q \in X_i$. Again this directly follows from construction, where we now consider Condition **R3**. \square

We next show that the entailment in Lemma 14 can be decided within the desired complexity bounds. By pushing negations inside and reordering, we obtain that the non-entailment in Lemma 14 is equivalent to the following.

$$\mathcal{K}_{a,w} \not\models \bigvee_{X_i \in \mathcal{S}(w), q \in X_i} \neg q^i \vee \bigvee_{X_i \in \mathcal{S}(w), q \notin X_i} q^i$$

For a query q , denote by \mathcal{A}_q the classical ABox obtained from q by replacing every variable by a fresh individual name. Clearly, our non-entailment holds if the following non-entailment is satisfied.

$$\langle \mathcal{T}_{a,w}, \mathcal{A}_{a,w} \cup \bigcup_{X_i \in \mathcal{S}(w), q \in X_i} \mathcal{A}_{q^i} \rangle \not\models \bigvee_{X_i \in \mathcal{S}(w), q \notin X_i} q^i$$

We obtain that, in order to use this to decide r-satisfiability, we have to decide the entailment of an exponentially-sized union of linearly sized CQs from an exponentially sized KB. For \mathcal{ELH} and \mathcal{SHQ} with $\mathbb{N}_{\text{Rig}} \neq \emptyset$, this already gives us the desired complexity bounds, since query entailment is in NP for \mathcal{ELH} [32] and in EXPTIME for \mathcal{SHQ} [28], thus giving us NEXPTIME and 2-EXPTIME upper bounds immediately. On the other hand, for DLs extending \mathcal{ALCI} and \mathcal{ALCO} , query entailment is 2-EXPTIME-hard [27, 30], so that we would obtain at least a 3-EXPTIME upper bound if we just apply these complexity bounds directly. To show that the above entailment can still be decided in 2-EXPTIME wrt. \mathcal{K} and ϕ , we examine the decision procedures presented in [22, 21, 11] to show that query entailment is in 2-EXPTIME for \mathcal{SHIQ} , \mathcal{SHOQ} and \mathcal{SHOI} , and show that they run in 2-EXPTIME wrt. \mathcal{K} and ϕ for the above entailment. The crucial property here is that, even though the KB and the UCQ are exponential in size, each CQ is only linear in the size of the input. We therefore first give some auxiliary lemmata that refine the complexity results from [22, 21] by close inspection of the construction used there. The lemmata are all of the same form and only differ in the DL under consideration.

Lemma 15. *For a given natural number n , entailment of disjunctions over CQs from \mathcal{SHIQ} -KBs can be decided in time double exponential in n if*

- *the size of the KB is at most exponential in n ,*
- *the number of CQs in the disjunction is at most exponential in n , and*
- *the size of each CQ is polynomial in n .*

Proof. The result follows from a close analysis of the 2-EXPTIME decision procedure for entailment of UCQs from \mathcal{SHIQ} KBs presented in [22]. The authors show the result for the slightly more expressive DL \mathcal{SHIQ}^\square , which additionally supports conjunctions of roles. The decision procedure makes use of several satisfiability tests of \mathcal{SHIQ}^\square KBs constructed based on the input. We show that this procedure runs within the bounds provided by the lemma by direct reference to the results from this publication. The main result is given in Theorem 32 in that paper. The main construction underlying the decision procedure is provided in Definition 21.

We first sketch the idea to decide query entailment for CQs q from classical KBs $\mathcal{K} = (\mathcal{T}, \mathcal{A})$. For a given CQ q , the authors define the set $\text{trees}_{\mathcal{K}}(q)$ which contain rewritings of q in form of concepts corresponding to all possible tree-shaped matches of q in models of \mathcal{K} , and the set $\text{ground}_{\mathcal{K}}(q)$, which contains ground CQs (that is, CQs that only contain individual names as terms, which may however use complex concepts) corresponding to *forest-shaped* matches in models of \mathcal{K} in which every root of the forest is matched to an individual in \mathcal{K} . The authors define the \mathcal{SHIQ}^\square -TBox $\mathcal{T}_q = \{\top \sqsubseteq \neg C \mid C \in \text{trees}_{\mathcal{K}}(q)\}$, which only has models in which there is no purely-tree shaped match of q . In addition, they define a series of classical \mathcal{SHIQ}^\square -ABoxes \mathcal{A}_q , which contain from each ground query $q' \in \text{ground}_{\mathcal{K}}(q)$ at least one assertion $\neg\alpha$, where $\alpha \in q'$.

If for any such ABox \mathcal{A}_q , $\langle \mathcal{T} \cup \mathcal{T}_q, \mathcal{A} \cup \mathcal{A}_q \rangle$ has a model, then this model is a counter-example for the entailment of q . Note that both \mathcal{T}_q and \mathcal{A}_q are linear in the size of $\text{trees}_{\mathcal{K}}(q)$ and $\text{ground}_{\mathcal{K}}(q)$, and that the number of possible choices for \mathcal{A}_q is exponentially bounded by the size of $\text{ground}_{\mathcal{K}}(q)$. To decide entailment of a query, we therefore iterate over all possible choices of \mathcal{A}_q and decide the satisfiability of $\langle \mathcal{T} \cup \mathcal{T}_q, \mathcal{A} \cup \mathcal{A}_q \rangle$.

The details about the construction of the sets $\text{trees}_{\mathcal{K}}(q)$ and $\text{ground}_{\mathcal{K}}(q)$ go beyond the scope of the paper. Relevant for us are the bounds the authors give on their size: by [22][Lemma 20], the size of $\text{trees}_{\mathcal{K}}(q)$ is polynomially bounded in the size of \mathcal{K} and exponentially bounded in the size of q , and so is the size of $\text{ground}_{\mathcal{K}}(q)$.

Now to handle UCQs $q_1 \vee \dots \vee q_n$, the same procedure is used based on the unions over all sets $\text{trees}_{\mathcal{K}}(q_i)$ and $\text{ground}_{\mathcal{K}}(q_i)$, $i \in \llbracket 1, n \rrbracket$ [22][Definition 21]. Clearly, if the number of CQs in the UCQ is exponentially bounded by a natural number n , and the size of each CQ is polynomially bounded in n , then the size of these unions are exponentially bounded in n , and there are at most double exponentially many KBs that have to be tested for satisfiability. As a consequence, if the KB and the UCQ are of the shape as in the Lemma, then entailment of the UCQ can be decided in time double exponential in n . \square

Lemma 16. *For a given natural number n , entailment of disjunctions over CQs from SHOQ-KBs can be decided in time double exponential in n if*

- *the size of the KB is at most exponential in n ,*
- *the number of CQs in the disjunction is at most exponential in n , and*
- *the size of each CQ is polynomial in n .*

Proof. The result follows from a close analysis of the 2-EXPTIME decision procedure for entailment of UCQs from SHOQ KBs presented in [21]. The decision procedure follows a similar idea than the one presented in [22], and also apply to the extension SHOQ^\square of SHOQ with role conjunctions. Specifically, to decide entailment of a CQ, a set $\text{con}_{\mathcal{K}}(q)$ of CQs of the form $C_1(x_1) \vee \dots \vee C_n(x_n)$ is defined, where each C_i , $i \in \llbracket 1, n \rrbracket$ is a SHOQ^\square -concept and $x_i \neq x_j$ for $i \neq j \in \llbracket 1, n \rrbracket$. For UCQs q , $\text{con}_{\mathcal{K}}(q)$ is defined as the union of all sets $\text{con}_{\mathcal{K}}(q')$ where $q' \in q$. Intuitively, each $q' \in \text{con}_{\mathcal{K}}(q)$ corresponds to a forest-shaped match in some model of \mathcal{K} , and every such match is represented by some CQ in $\text{con}_{\mathcal{K}}(q)$. A consequence of this is that $\mathcal{K} \models \bigvee_{q_i \in \text{con}_{\mathcal{K}}(q)} q_i$ iff $\mathcal{K} \models q$ [21, Theorem 6]. The size bounds on $\text{con}_{\mathcal{K}}(q)$ are given in [21, Lemma 7]: its size is at most i) polynomial in the size of \mathcal{K} and ii) exponential in the size of q . For a UCQ whose number of CQs is exponentially bounded in n and the size of each CQ is polynomially bounded in n , from the fact that $\text{con}_{\mathcal{K}}(q)$ corresponds to the union of all sets $\text{con}_{\mathcal{K}}(q')$ where $q' \in q$, it follows that $\text{con}_{\mathcal{K}}(q)$ is exponential in both n and the size of \mathcal{K} .

Now to decide $\mathcal{K} \models \bigvee_{q_i \in \text{con}_{\mathcal{K}}(q)} q_i$, we use the fact that the atoms in each query in $\text{con}_{\mathcal{K}}(q)$ are variable-disjoint, and built a sequence of *reduction KBs*. Each reduction KB is obtained from selecting from each CQ in $q_i \text{con}_{\mathcal{K}}(q)$ one atom $C_i(x_i)$, and adding $\top \sqsubseteq \neg C_i$ to \mathcal{K} . If one of them is unsatisfiable, $\mathcal{K} \not\models \bigvee_{q_i \in \text{con}_{\mathcal{K}}(q)} q_i$ and $\mathcal{K} \not\models q$.

The size of each KB is linear in $\text{con}_{\mathcal{K}}(q)$, and there are exponentially many possible choices wrt. the size of $\text{con}_{\mathcal{K}}(q)$. If q is shaped as in the lemma, this amounts to a number of satisfiability tests that is double exponential in n , where each KB is exponential in the size of n . As shown in the remainder of [21], each such satisfiability test can be performed in time exponential to the size of the KB, so that we obtain that the overall decision procedure runs in time double exponential in n . \square

Lemma 17. *For a given natural number n , entailment of disjunctions over CQs from SHOI-KBs can be decided in time double exponential in n if*

- the size of the KB is at most exponential in n ,
- the number of CQs in the disjunction is at most exponential in n , and
- the size of each CQ is polynomial in n .

Proof. where we inspect the procedure presented in [11] deciding entailment of *regular path queries* (RPQs) in *ZOI*-KBs. Regular path queries are a generalisation and UCQs, and *ZOI* is an extension of *SHOI*, so that this procedure can also be used to decide UCQ entailment for *SHOI*-KBs. The authors in [11] reduce UCQ-entailment to the emptiness problem for *one-way non-deterministic parity tree automata* (1NPAs). Specifically, given a *ZOI* KB \mathcal{K} and a RPQ q , they construct a 1NPA $\mathbf{A}_{\mathcal{K} \not\models q}$ which accepts the empty language iff $\mathcal{K} \models q$. We first give an overview over the main ideas, before we argue why this approach can decide bounded CQs as in the Lemma in double-exponential time. Specifically, the authors exploit the fact that *ZOI* has the forest model property for query-entailment, that is, query entailment from *ZOI*-KBs \mathcal{K} can be completely characterized by restricting to models of \mathcal{K} that can be mapped to a labeled forest where every node represents a domain element, edges correspond to role-connections and the roots of the forest to the named individuals, and every role-connection that does not go to a named individual has a corresponding edge in the forest. To characterise these forest models by means of automata, they represent forest-interpretations directly in labeled trees of fixed branching degree, where the branching degree is bounded by the KB and nodes are labeled with sets of individual, concept and role names. Here, the role names refer to the incoming edge, and individual names only occur on the direct successors of the root [11, Definition 3.7 and beginning of Section 4]. They then construct a *fully enriched automaton* (FEA) with a polynomial number of states and a constant index, which they step-wise translate into an 1NPA of $\mathbf{A}_{\mathcal{K}}$ accepting exactly those labeled trees that correspond to a forest-shaped model of \mathcal{K} . $\mathbf{A}_{\mathcal{K}}$ has a double-exponential number of states and a constant index².

To capture query entailment, they define another 1NPA $\mathbf{A}_{\neg q}$ which accepts exactly those labelled trees that correspond to a forest-shaped interpretation in which the query q is not entailed, and build the intersection of the automata $\mathbf{A}_{\mathcal{K}}$ and $\mathbf{A}_{\neg q}$, which is the final 1NPA $\mathbf{A}_{\mathcal{K} \not\models q}$. If $\mathbf{A}_{\mathcal{K} \not\models q}$ is empty, that is, the language of accepted trees is empty, there cannot be a forest-shaped model of \mathcal{K} in which q is not entailed, and correspondingly, $\mathcal{K} \models q$.

While [11] only briefly sketch the automaton $\mathbf{A}_{\neg q}$, a construction of an automaton accepting the same language is described in detail in [12]: this automaton has an exponential number of states while its index is exponential. The relevant construction in [12, Section 5] can be easily adapted so that number of states and index depend only on the size of the query, as the only relevant factors are the variables, concept and role names occurring in it.

It is not hard to see that for a UCQ $q = q_1 \vee \dots \vee q_m$ the automaton $\mathcal{A}_{\neg q}$ is equivalent to the intersection of the automata $\mathcal{A}_{\neg q_1}, \dots, \mathcal{A}_{\neg q_m}$, as the disjuncts can be tested independently on a given interpretation. Now the number of states and the index of the intersection of two 1NPAs is determined as follows [12, Proposition 2.15]: Let $Q(\mathbf{A})$ denote the states of \mathbf{A} and $\text{ind}(\mathbf{A})$ its index, then

$$\text{ind}(\mathbf{A}_1 \cap \mathbf{A}_2) = O(f(\mathbf{A}_1, \mathbf{A}_2))$$

and

$$|Q(\mathbf{A}_1 \cap \mathbf{A}_2)| \leq 2^{O(f(\mathbf{A}_1, \mathbf{A}_2)^2)} \cdot f(\mathbf{A}_1, \mathbf{A}_2) \cdot |Q(\mathbf{A}_1)| \cdot |Q(\mathbf{A}_2)|,$$

²The authors only explicitly spell out the size of the two-way alternating parity tree (2APA) which they construct in the second-last step of their transformation and then translate to the 1NPA. The number of states of the 2APA is polynomially bounded in the size of \mathcal{K} and has a constant index. However, according to [12, Proposition 2.12], the transformation comes with an exponential blow-up in the number of states, while it keeps the index. Therefore, the final 1NPA $\mathbf{A}_{\mathcal{K}}$ constructed here has a double-exponential number of states and a constant index.

where $f(\mathbf{A}_1, \mathbf{A}_2) = \mathbf{A}_1 + \mathbf{A}_2 + 1$. Let $\mathbf{A}'_{-q} \equiv \mathbf{A}_{-q}$ denote the intersection of $\mathbf{A}_{q_1}, \dots, \mathbf{A}_{q_n}$. We obtain that

$$\text{ind}(\mathbf{A}'_{-q}) = O\left(\sum_{1 \leq i \leq m} \text{ind}(\mathbf{A}_{-q_i}) + 1\right),$$

and

$$|Q(\mathbf{A}'_{-q})| \leq 2^{O\left(\left(\sum_{1 \leq i \leq n} \text{ind}(\mathbf{A}_{-q_i}) + 1\right)^{2m}\right)} \cdot \left(\sum_{1 \leq i \leq n} \text{ind}(\mathbf{A}_{-q_i}) + 1\right) \cdot \prod_{1 \leq i \leq n} |Q(\mathbf{A}_{-q_i})|$$

Since for each \mathbf{A}_{-q_i} , the $\text{ind}(\mathbf{A}_{-q_i})$ is single exponential in the size of q_i , and $|Q(\mathbf{A}_{-q_i})|$ is double exponential in the size of q_i , we obtain that, provided that q is shaped as in the lemma, $\text{ind}(\mathbf{A}'_{-q})$ is single exponential in n and $|Q(\mathbf{A}'_{-q})|$ is double exponential in n , and the same holds for $\mathbf{A}'_{\mathcal{K} \neq q} = \mathbf{A}_{\mathcal{K}} \cap \mathbf{A}'_{-q}$. Emptiness of 1NDAs \mathbf{A} can be decided in time $O(|Q(\mathbf{A})|^{\text{ind}(\mathbf{A})})$, which is double exponential in n for $\mathbf{A}'_{\mathcal{K} \neq q}$. We obtain that for $\mathcal{S}\mathcal{H}\mathcal{O}\mathcal{I}$ -KBs \mathcal{K} , entailment of UCQs q can be decided in time double exponential in n , provided that the size of \mathcal{K} and the number of CQs in q is exponentially bounded in n and the size of each CQ is polynomially bounded in n . \square

In connection with Lemmata 14, Lemmata 15, 16 and 17 now allow to prove almost all cases in Lemma 6. For the only remaining case where $\mathcal{L} = \mathcal{S}\mathcal{H}\mathcal{Q}$ and $\mathbf{N}_{\text{Rrig}} = \emptyset$, we refer to [3, Section 6], where r -satisfiability is considered in the context of TQ-satisfiability in non-probabilistic, temporal KBs.

Lemma 6. *Wrt. the size of \mathcal{K} and ϕ , r -satisfiability for \mathcal{L} -TPKBs can be decided in*

1. NEXPTIME for $\mathcal{L} = \mathcal{E}\mathcal{L}\mathcal{H}$,
2. NEXPTIME for $\mathcal{L} = \mathcal{S}\mathcal{H}\mathcal{Q}$ if $\mathbf{N}_{\text{Rrig}} = \emptyset$,
3. 2-EXPTIME for $\mathcal{L} \in \{\mathcal{S}\mathcal{H}\mathcal{I}\mathcal{Q}, \mathcal{S}\mathcal{H}\mathcal{O}\mathcal{Q}, \mathcal{S}\mathcal{H}\mathcal{O}\mathcal{I}\}$, and
4. it is decidable for $\mathcal{L} = \mathcal{A}\mathcal{L}\mathcal{C}\mathcal{H}\mathcal{O}\mathcal{I}\mathcal{Q}$.

A.4 Complexity Upper Bounds

We are now ready to prove the complexity upper bound.

Lemma 18. *Given the mappings $\mathcal{S} : \Omega_{\mathcal{K}} \rightarrow 2^{\{q_1, \dots, q_m\}}$ and $a : \Omega_{\mathcal{K}} \times \llbracket 1, n \rrbracket \rightarrow 2^{\{q_1, \dots, q_m\}}$ s.t. \mathcal{S} is r -satisfiable wrt. a , it can be decided in EXPSpace whether there exists a quasi-model for ϕ in \mathcal{K} wrt. \mathcal{S} and a .*

Proof. By Lemma 7, there exists a quasi-model Q_1, \dots as in the lemma iff there exists a periodic quasi-model of the form

$$Q_1, \dots, Q_m(Q_{m+1}, \dots, Q_{m+o})^\omega,$$

where both m and o are double-exponentially bounded.

To verify the existence of such a quasi-model in (non-deterministic) exponential space, we proceed as follows. We first guess the numbers m and o , which both require at most exponentially many bits in binary representation. We then guess the quasi-states Q_i one after the other, keeping always two proceeding quasi-states in memory, and verify that they satisfy the Conditions **Q1**–**Q3**, and additionally that $\psi_1 \mathcal{U} \psi_2 \in Q_i(w)$ iff $\psi_2 \in Q_i(w)$ or $\psi_1 \mathcal{U} \psi_2 \in Q_{i+1}(w)$, and similarly for queries of the form $\psi_1 \mathcal{S} \psi_2$. To verify that each \mathcal{S}/\mathcal{U} -formula is eventually satisfied, we keep

a set of those queries for each possible world that have not been satisfied yet, which we update at each time point. In the same manner, we check whether the negation of a \mathcal{S}/\mathcal{U} -formula is satisfied. After we guessed the quasi-state Q_{m+1} , we store this quasi-state in memory, as well as all \mathcal{S}/\mathcal{U} -queries that still have to be satisfied at this point. We then proceed until Q_{m+o+1} , and verify that all \mathcal{S}/\mathcal{U} -queries from Q_{m+1} have been satisfied in the meanwhile, and that $Q_{m+o+1} = Q_{m+1}$. Since $\text{NEXPSPACE} = \text{EXPSPACE}$, the above procedure decides existence of a quasi-model in exponential space. \square

Theorem 8. *The complexity upper bounds for general TPQs in Figure 1 hold.*

Proof. Entailment of a query ϕ corresponds to non-satisfiability of the query $\neg\phi$, and both complexity classes EXPSPACE and 2-EXPTIME are closed under complement. To decide satisfiability of a TPQ ϕ with negations in a TPKB \mathcal{K} , we proceed as follows. We iterate over the (double exponentially many) different possible mappings \mathcal{S} and a , each of which require only exponential space. For each pair \mathcal{S} , a , we first verify whether \mathcal{S} is r -satisfiable wrt. a , and then whether there exists a quasi-model that is compatible with \mathcal{S} and a . By Lemma 18, the last test takes exponential space in each iteration. The only step that differs depending on the DL in question and which rigid names we allow is verifying r -satisfiability, which by Lemma 6, can be decided in NEXPTIME (and therefore in EXPSPACE) for $\mathcal{L} = \mathcal{EL}$, as well as for $\mathcal{L} = \mathcal{SHQ}$ provided that $\text{N}_{\text{Rig}} = \emptyset$. For $\mathcal{L} \in \{\mathcal{SHIQ}, \mathcal{SHOQ}, \mathcal{SHOI}\}$, it can be decided in 2-EXPTIME . We thus obtain the required complexity bounds. For $\mathcal{L} = \mathcal{ALCHOIQ}$, it suffices to use that r -satisfiability is decidable for this logic. \square

A.5 Positive Temporal Probabilistic Queries

We first establish the complexity bounds for the non-probabilistic case, that is, entailment of *temporal queries* (TQs), which are TPQs without probability operators, from *temporal knowledge bases* (TKBs), which are TPKBs without probabilities different from 1.

Lemma 19. *Entailment of positive TQs is in P wrt. data complexity and NP complete wrt. combined complexity, even if $\text{N}_{\text{Crig}} \neq \emptyset$ and $\text{N}_{\text{Rig}} \neq \emptyset$.*

Proof. [7] show that query entailment is in P data complexity, even if $\text{N}_{\text{Rig}} \neq \emptyset$. The only remaining case is the combined complexity for \mathcal{EL} TKBs.

To establish the combined complexity, we describe an NP procedure for a given positive TQ ϕ and \mathcal{EL} TKB $\langle \mathcal{T}, (\mathcal{A}_i)_{i \in \llbracket 1, n \rrbracket} \rangle$. For $X \in \text{N}_{\text{C}} \cup \text{N}_{\text{R}}$, denote by $X^{(i)}$ the name X if $X \in \text{N}_{\text{Rig}}$, and a fresh name X^i if $X \notin \text{N}_{\text{Rig}}$, and for a given axiom/assertion/query α , denote by $\alpha^{(i)}$ the result of replacing every name X in α by $X^{(i)}$. Define an atemporal KB $\mathcal{K}' = \{\mathcal{T}', \mathcal{A}'\}$ based on the TKB $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{i \in \llbracket 1, n \rrbracket} \rangle$ by $\mathcal{T}' = \{\alpha^{(i)} \mid \alpha \in \mathcal{T}, i \in \llbracket 1, n+1 \rrbracket\}$ and $\mathcal{A}' = \{\alpha^{(i)} \mid \alpha \in \mathcal{A}_i, i \in \llbracket 1, n+1 \rrbracket\}$. \mathcal{K}' is polynomial in \mathcal{K} , and one can show that for any axiom/assertion/CQ α and $i \in \llbracket 1, n+1 \rrbracket$, we have $\mathcal{K}, i \models \alpha$ iff $\mathcal{K}' \models \alpha^{(i)}$ [8].

In order to decide entailment of a TQ ϕ , we guess a certificate that assigns to each pair (i, ψ) of a time point $i \in \llbracket 1, n+n_t \rrbracket$ and a CQ ψ occurring in ϕ a truth value, and, in case **true** is assigned to such a pair (i, ψ) , a certificate for the entailment of ψ at i (such a certificate exists since entailment of CQs is in NP wrt. combined complexity). For any time point after n , the entailment of a CQ solely depends on the rigid names. Therefore, for every CQ q in ϕ , if $\mathcal{K}, n+1 \models q$, then $\mathcal{K}, n+i \models q$ for all $i > 1$. Based on the guessed truth-assignment of CQs, we can now evaluate the entailment of ϕ as in the propositional case, which for LTL-formulae without negation symbols can be done in P [9]. As this certificate can be guessed and verified in non-deterministic polynomial time, we obtain an NP-upper bound. \square

The proof of Theorem 10 further depends on the following lemma, which limits the time points we have to consider explicitly.

Lemma 20. *Let ϕ be a TPQ, $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{i \in \llbracket 1, n \rrbracket} \rangle$ a TKB and n_t be the maximal nesting depth of temporal operators in ϕ . Then, for every $i > n + n_t$, $\mathcal{K}, i \models \phi$ iff $\mathcal{K}, n + n_t + 1 \models \phi$.*

Proof. We do the proof by structural induction on ϕ , and distinguish the cases based on the structure of ϕ .

1. If ϕ is a CQ, note that the only way in which \mathcal{K} restricts its models for time points after n is via its rigid names. Therefore, we have for all $i > n$, $\mathcal{K}, i \models \phi$ iff $\mathcal{K}, n + 1 \models \phi$.
2. If ϕ is of one of the forms $\psi_1 \wedge \psi_2$ and $\psi_1 \vee \psi_2$, the hypothesis follows by direct application of the inductive hypothesis.
3. If ϕ is of the form $\bigcirc^- \psi$, we have by inductive hypothesis that for all $i > n + n_t - 1$, $\mathcal{K}, n + n_t + 1 \models \psi$ iff $\mathcal{K}, i \models \psi$ iff $\mathcal{K}, i + 1 \models \bigcirc^- \psi$ iff $\mathcal{K}, j \models \phi$ for all $j > n + n_t$.
4. If ϕ is of one of the forms $\bigcirc \psi$, $\diamond \psi$, $\diamond^- \psi$, $\square \psi$, $\square^- \psi$, $\psi_1 \mathcal{U} \psi_2$ or $\psi_1 \mathcal{S} \psi_2$, we note that by inductive hypothesis, for all $i > n + n_t - 1$ $\mathcal{K}, i \models \psi$ (ψ_1, ψ_2) iff $\mathcal{K}, n + n_t \models \psi$ (ψ_1, ψ_2), which implies $\mathcal{K}, i \models \phi$ iff $\mathcal{K}, n + n_t + 1 \models \psi$ for all $i > n + n_t - 1$, and consequently also $\mathcal{K}, i \models \phi$ iff $\mathcal{K}, n + n_t + 1 \models \phi$ for all $i > n + n_t$.

□

We can now provide the upper bounds stated in Theorem 10. A central technique used for this is to flatten TPQs using an abstraction of the probability expressions $\mathbb{P}_{\geq p}(\psi)$ occurring in the query. We identify each such expression with the assertion $A_{p,\psi}(a)$, where $A_{p,\psi}$ is fresh, which we add to the ipABox \mathcal{A}_i once we established that $\mathbb{P}_{\geq p}(\psi)$ is entailed at i . To capture this abstraction in a given TPQ ψ , we denote by ψ_f the result of replacing every outermost sub-query in ψ of the form $\mathbb{P}_{\geq p}(\psi)$ with $\exists x. A_{p,\psi}(x)$.

Lemma 21. *Entailment of TPQs from \mathcal{EL} - and DL-Lite-TPKBs is in PP wrt. data complexity, even if $\mathbb{N}_{\text{Rig}} \neq \emptyset$. It is in PP^{NP} wrt. combined complexity if the nesting depth of probability-operators in the query is bounded, and otherwise in P^{PP} .*

Proof. Before we consider nested probability operators, we consider the basic case of simple TPQs of the form $\mathbb{P}_{\geq p}(\phi)$, where ϕ does not contain any probability operators. Entailment of such a TPQ can be decided by checking for which possible world $w \in \Omega_{\mathcal{K}}$, $w, i \models \phi$, and then summing the probabilities of these worlds. This can be implemented by a probabilistic Turing machine (which uses an NP-oracle in the case of the combined complexities), which constructs a single possible world $w = (\mathcal{A}'_i)_{i \in \llbracket 1, n \rrbracket}$ on each branch, while taking care that the probabilities of the possible worlds are reflected by the probabilities in the Turing machine. For each $i \in \llbracket 1, n \rrbracket$ and $\alpha: p \in \mathcal{A}_i$, the machine adds α to \mathcal{A}'_i on b_1 succeeding branches, and does not add α to \mathcal{A}'_i on b_2 succeeding branches, where $\frac{b_1}{b_1 + b_2} = p$. After all axioms are processed, accept if $\langle \mathcal{T}, (\mathcal{A}'_i)_{i \in \llbracket 1, n \rrbracket} \rangle, i \models \phi$, which can be decided in P data complexity and NP combined complexity. By adding further dummy states to the Turing machine, we can ensure that the machine accepts at least half of its computation paths iff $\mathcal{K} \models \mathbb{P}_{\geq p}(\phi)$, so that entailment of the simple TPQ ϕ is decided in PP data complexity and PP^{NP} combined complexity.

To decide entailment of TPQs that contain several probability operators, we proceed in k rounds, where k is the maximal nesting depth of probability operators in ϕ , and test in each round for the entailment of probabilistic sub-queries at different time points. Let n_t denote the maximal nesting depth of temporal operators in ϕ . It can be shown that we have to consider only the

first $n + n_t$ time points. In each round $r \in \llbracket 1, k \rrbracket$, we iterate over all subformulae in ϕ that are of the form $\mathsf{P}_{\geq p}(\psi)$, where ψ contains at most $r - 1$ nestings of probability operators, and over all time points $i \in \llbracket 1, n + n_t + 1 \rrbracket$, and decide whether $\mathcal{K}, i \models \mathsf{P}_{\geq p}(\psi_f)$. If $\mathcal{K}, i \models \mathsf{P}_{\geq p}(\psi_f)$, we add $A_{p,\psi}(a)$ to \mathcal{A}_i . In the last round, we processed all probability operators, and decide whether $\mathcal{K} \models \mathsf{P}_{\geq 1}(\phi_k)$. Provided the nesting depth of probability operators is bounded, (as is always the case for data complexity), we can now use the fact that PP (and therefore also PP^{NP}) is closed under k -round polynomial truth table reductions [19]. These are defined as a sequence of k sets of polynomially many polynomial truth-table reductions, where k is a constant, and each truth-table reduction only depends on the input and the results of previous rounds. If the nesting-depth of probability operators is bounded, the above procedure can be described by such a reduction, and we obtain the PP and PP^{NP} upper bounds. Regarding the combined complexity with unbounded nesting of probability operators, we note that the above procedure can be implemented by a polynomial Turing machine that decides entailment of simple TPQs using a PP^{NP} oracle, so that we obtain a $\text{P}^{\text{PP}^{\text{NP}}}$ upper bound. Now, using Toda's result that $\text{PP}^{\text{PH}} \subseteq \text{P}^{\text{PP}}$ [33], we can internalise all calls to the PP^{NP} oracle in a P^{PP} machine, so that we obtain a P^{PP} upper bound for the combined complexity without bound on the nesting-depth of probability operators. \square

For the data complexity, our upper bound is matched by PP-hardness of the atemporal case [24]. We could not find a lower bound for the combined complexity in the literature for our precise setting (ipABoxes or tuple-independent databases). We therefore provide a proof for it here.

Lemma 22. *Entailment of TPQs from TPKBs is PP^{NP} -hard.*

Proof. We only need to provide a lower bound for the combined complexity. We do the proof by reduction of the PP^{NP} complete problem $\text{M}\exists\text{CNF3}$: given a QBF-formula of the form $\phi = \exists x_1, \dots, x_n. \phi'$, where ϕ' is a CNF3-formula over the variables $\{x_1, \dots, x_n, y_1, \dots, y_m\}$ with clauses $\{c_1, \dots, c_o\}$, decide whether at least half of assignments of truth values to the variables y_1, \dots, y_m make ϕ true [35]. As it turns out, we only need a single ipABox for this. The ipABox contains for every variable $x_i, i \in \llbracket 1, n \rrbracket$ the assertions $\mathsf{B}(x_i^+)$ and $\mathsf{B}(x_i^-)$, and for every variable $y_i, i \in \llbracket 1, m \rrbracket$ the assertions $\mathsf{B}_i^+(y_i^+): 0.5$ and $\mathsf{B}_i^-(y_i^-): 0.5$. Intuitively, $\mathsf{B}_i^+(y_i^+)$ is entailed in a possible world that corresponds to an assignment of true to the variable y_i , while $\mathsf{B}_i^-(y_i^-)$ is entailed in a possible world that corresponds to an assignment of false to the variable y_i . Since all probabilities are independent, we will have worlds that correspond to “invalid variable assignments”, in the sense that they either do not assign a truth value to every variable, or multiple truth values. We will take care of this later. We use the TBox axioms $B_i^+ \sqsubseteq B_i$, $B_i^- \sqsubseteq B_i$, $B_i^+ \sqsubseteq B^+$, $B_i^- \sqsubseteq B^-$, $B^+ \sqsubseteq B$ and $B^- \sqsubseteq B$ to abstract away from the specific assignment if needed.

For every literal l , denote by $v(l)$ the variable in l . For every clause $c_j = l_1 \vee l_2 \vee l_3$, $c \in \llbracket 1, o \rrbracket$, and truth valuation π that makes c_j true, add the assertions

$$M(c_j, \pi), M(\pi, l'_1), M(\pi, l'_2), M(\pi, l'_3),$$

where for $i \in \llbracket 1, 3 \rrbracket$, $l'_i = v(l_i)$ if $\pi(l_i) = \mathbf{true}$, and $l'_i = \neg v(l_i)$ if $\pi(l_i) = \mathbf{false}$. As last assertion, we add $\mathsf{H}(a): 0.5$, which only serves the purpose of being satisfied in at least half of the possible worlds.

Our CQ is now composed of three queries q_1, q_2 and q_3 defined next. The query

$$q_1 = \exists y_1, \dots, y_m : B_1(y_1) \dots B_m(y_m)$$

is entailed in every possible world which assigns a truth value to each variable $y_i, i \in \llbracket 1, m \rrbracket$. The query

$$q_2 = \exists y : B^+(y) \wedge B^-(y)$$

is entailed in the possible worlds that assign two truth values to some variable y . Finally, the query

$$q_3 = \exists x_1, \dots, x_n, y_1, \dots, y_m, z_1, \dots, z_o : \bigwedge_{i \in \llbracket 1, o \rrbracket} \tau(c_i),$$

where for $c_i = l_1 \vee l_2 \vee l_3$,

$$\begin{aligned} \tau(c_i) = & M(c_i, z_i), M(z_i, v(l_1)), M(z_i, v(l_2)), M(z_i, v(l_3)) \\ & \wedge \bigwedge_{i \in \llbracket 1, 3 \rrbracket} B(v(l_i)), \end{aligned}$$

is satisfied in all possible worlds that correspond to an assignment that make ϕ true. q_3 can only be entailed in a possible world in which q_1 is also entailed (otherwise, we lack variables for some of the clauses). The query $(q_1 \wedge q_2) \vee (H(c) \wedge q_2)$ is entailed in (1) all possible worlds that correspond to an assignment that is complete but assigns to at least one variable two values and (2) half of the possible worlds that correspond to assignments that are both incomplete and assign two values to a variable. Due to symmetry, this query is thus entailed in exactly half of those possible worlds that do not correspond to a valid variable assignment. Consequently, the query

$$q = (q_1 \wedge q_2) \vee (H(c) \wedge q_2) \vee q_3$$

is entailed in more than half of all possible worlds iff ϕ is satisfied for more than half of its valid assignments, so that $P_{\geq 0.5}(q)$ is entailed iff ϕ is satisfied by at least half of the assignments. Note furthermore that both the TPKB and the TPQ are polynomial in the input, so that we obtain the PP^{NP} lower bound. \square