Ontology-mediated query answering over temporal and inconsistent data
Camille Bourgaux, Patrick Koopmann, Anni-Yasmin Turhan

To cite this version:
Camille Bourgaux, Patrick Koopmann, Anni-Yasmin Turhan. Ontology-mediated query answering over temporal and inconsistent data. Semantic Web – Interoperability, Usability, Applicability, 2019, 10.3233/SW-180337. hal-02111428

HAL Id: hal-02111428
https://hal.science/hal-02111428
Submitted on 26 Apr 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Ontology-mediated query answering over temporal and inconsistent data

Camille Bourgaux a,* , Patrick Koopmann b and Anni-Yasmin Turhan b

a Laboratoire Traitement et Communication de l’Information, Télécom ParisTech, France
E-mail: camille.bourgaux@telecom-paristech.fr
b Institute of Theoretical Computer Science, Technische Universität Dresden, Germany
E-mails: patrick.koopmann@tu-dresden.de, anni-yasmin.turhan@tu-dresden.de

Abstract. Stream-based reasoning systems process data stemming from different sources that are received over time. In this kind of applications, reasoning needs to cope with the temporal dimension and should be resilient against inconsistencies in the data. Motivated by such settings, this paper addresses the problem of handling inconsistent data in a temporal version of ontology-mediated query answering. We consider a recently proposed temporal query language that combines conjunctive queries with operators of propositional linear temporal logic (LTL), and consider these under three inconsistency-tolerant semantics that have been introduced for querying inconsistent description logic knowledge bases. We investigate their complexity for temporal $\mathcal{EL} \perp$ and DL-Lite$^R$ knowledge bases. In particular, we consider two different cases, depending on the presence of negations in the query. Furthermore, we complete the complexity picture for the consistent case. We also provide two approaches toward practical algorithms for inconsistency-tolerant temporal query answering.

Keywords: temporal query answering, inconsistency-tolerant reasoning, description logics, computational complexity

1. Introduction

For applications that rely on sensor data, such as context-aware applications, ontologies can enrich and abstract the (numerical) stream data by means of background knowledge. This richer view on the data often results in more query results than over the data alone. Since the collected data usually provides an incomplete description of the observed system, the closed world assumption employed by database systems, where facts not present are assumed to be false, is not appropriate. Most applications that rely on sensor streams observe some kind of running system over time. In order to be able to react to the behaviour of the observed system, they need to employ some representation of temporal information and a query mechanism that can reference this temporal information. If the sources of the collected data are not reliable, as it might be in case of faulty sensors, the internal representation of the observations may contain inconsistencies. In such cases, query mechanisms that rely on logical reasoning are effectively useless, as everything would follow from an inconsistent knowledge base. As a counter measure to this effect, inconsistency-tolerant semantics for answering ontology-mediated queries have been devised. In this paper, we investigate combinations of inconsistency-tolerant and temporal query answering w.r.t. ontologies. This setting is fairly abstract and deliberately neglects implementation details.
as it provides foundations on the combination of three aspects vital to most stream reasoning applications. First, stream reasoning requires background knowledge either on the streams used or on the context in which the information provided by the streams is to be handled. Both kinds of background knowledge can be represented by ontologies and be exploited by ontology reasoning systems. Second, stream reasoning inherently has to cope with the temporal dimension. This dimension is given by the sequence of observations that individual items in the streams provide. Thus, traditional reasoning services that are oblivious of the temporal dimension are hardly of use to stream reasoning applications. Naive combinations of such temporal reasoning services can compute some forms of temporal consequences, but these combinations need not be complete and thus could miss consequences. To stream reasoning applications relying on such ad-hoc systems, it is not discernible when the naive reasoner does or does not provide a reliable answer. Third, and most importantly, the different streams or data sources need not be free of errors due to the methods by which the data is gathered or provided to the system. Moreover, an automatic combination of open stream sources to which a stream reasoning system can subscribe simply cannot guarantee a consistent combination of such sources. Thus, in general, the combination of the data used by a stream reasoning system can simply contain contradictory information. The combination of these three aspects is so far not investigated in the literature. This is why our setting abstracts from many application specific details and variations of stream reasoning systems, in order to concentrate on the combination of the three fundamental conditions inherent to most such systems. Our goal is to provide an initial orientation for selecting between the many design choices by which each of these factors can be addressed in a stream reasoning system, such as expressivity of the ontology and query language, treatment of temporal information, and different inconsistency-tolerant semantics. To this end we map out the landscape of the computational complexity for many combinations of the three design factors for stream reasoning systems in this paper. In preparation for this, we examine the three design aspects in more detail.

In many stream reasoning systems, the collected data is transformed into an abstract logical representation, and situation recognition is performed by some kind of logical inference over the abstract logical representation. There are stream reasoning approaches based on rules, such as answer set programming [1–3], (datalog) rules and approaches based on ontology languages [4–7]. The ontology-based approaches mostly employ the framework of ontology-mediated queries, where forms of conjunctive queries are answered over data that is enriched by an ontology, to perform situation recognition. The ontology languages that are investigated for situation recognition are usually those where reasoning is of lower computational complexity in order to obtain systems with low execution times.

In this paper, we investigate the lightweight description logics (DLs) $\cal{EL}_\perp$ and DL-Lite$_R$, for which answering conjunctive queries is tractable (respectively in $P$ and $AC^0$ w.r.t. the size of the data). The low complexity for query answering in DL-Lite$_R$ made it the choice for the OWL 2 QL profile [8] in the latest version of OWL [9], the W3C-standardized ontology language for the Semantic Web. For similar reasons, the logic $\mathcal{EL}$ was picked as the core of the OWL 2 EL profile. Both DL-Lite$_R$ and $\mathcal{EL}_\perp$ aim to use database systems to answer conjunctive queries and are thus good candidates for implementing ontology-based stream reasoning. In DL-Lite$_R$, the query can be rewritten using the information from the ontology such that the resulting query can be evaluated directly on the data, i.e. stored in a database [10]. For query answering in $\mathcal{EL}$, there are several approaches. In the combined approach [11] the data is augmented in a query-independent way to build a canonical model by adding information form the terminology. The query is then evaluated over this model and unsound answers are filtered out. Alternatively, a translation to datalog variants can be used [12] for query answering in $\mathcal{EL}$ dialects.

In stream reasoning approaches in general, the temporal information is often represented by associating data with the time point at which it was collected. Regarding the language in which queries can be formulated, many variations that capture the temporal aspect have been studied in recent research [2–4, 13]. Window-based approaches admit to concentrate on recent continuous substreams when answering queries over the data, and are the most prominent in implemented systems [2–4]. Ontology-based approaches mostly cover classical temporal logics such as linear temporal logic (LTL) [14] (see [5, 15–18]) or metric variants of temporal logics [19, 20] to enrich the query language. For a recent overview on temporal ontology-mediated querying see [20, 21]. Note that window-based and ontology-based approaches can be
combined. For example, it might often be convenient to first extract a time window of interest and then apply ontology-based query answering to its content. Or alternatively, it is possible to refer to subsequences of the time line using LTL operators, though with a semantics that differs from a standard window operator (see Section 2 for an example.)

Ontology-based approaches for stream reasoning often admit the use of temporal operators only in the query language and use classical ontologies without any temporal operators together with sequences of datasets. Each dataset in such a sequence contains data collected at the same time point. The ontology together with the sequence of datasets constitute the temporal knowledge base. Queries can then refer to the different time points by means of temporal operators. This kind of setting has been intensively investigated for temporal conjunctive queries, that is, queries with temporal operators from LTL appearing in front of Boolean combinations of atoms, for expressive DLs in [5, 22], for DL-LiteR in [6], and for EL in [23]. We base our study in this paper on this general setting.

For stream reasoning systems, erroneous data can be a severe problem, as for instance pointed out in [24]. If inconsistencies arise in the knowledge base, the logical reasoning mechanisms are rendered useless. There are several directions of research to cope with this problem. While some employ non-monotonic reasoning techniques [25, 26], others try to resolve the inconsistencies directly [27] or perform reasoning with respect to inconsistency-tolerant semantics (see [28] for a recent overview). We follow the latter road, since the techniques developed there are tailored to ontology-mediated queries and often of lower complexity than the other approaches for resolving inconsistencies.

A prominent approach for inconsistency-tolerant reasoning is to consider repairs of the knowledge base, i.e., maximal consistent subsets of the data, and then to perform query answering with respect to these subsets. Arguably the most natural and well-known inconsistency-tolerant semantics is the AR semantics [29, 30], inspired by consistent query answering in the database setting [31], which considers the queries that hold in every repair. However, AR query answering is intractable even for very simple ontologies [32], which leads the authors of [29, 30] to propose an approximation of AR tractable for DL-LiteR, namely the IAR semantics, which queries the intersection of the repairs. Beside its better computational properties, this semantics is more cautious, since it provides answers supported by facts that are not involved in any contradictions. It may therefore be interesting in our setting when the observed system should change its behaviour only if some situation has been recognized with a very high confidence. Finally, the brave semantics [33] returns every answer that holds in some repair, so is supported by some consistent set of facts. This less cautious semantics may be relevant for context recognition, when critical situations must imperatively be handled.

For the two DLs to be investigated in this paper, answering of (atemporal) conjunctive queries under these inconsistency-tolerant semantics has already been investigated for DL-LiteR in [29, 30, 33, 34] and for EL in [28, 35]. Attention has then turned to the problem of designing algorithms and implementing these alternative semantics. Most work has focused on the IAR semantics and dialects of DL-Lite, due to the aforementioned tractability result [30, 36, 37]. A notable exception is the COAPri system, which implements all three mentioned semantics—AR, IAR and brave—for DL-LiteR knowledge bases [38, 39].

So far, inconsistency-tolerant semantics have not been investigated in combination with temporal reasoning. In this paper, we lift inconsistency-tolerant semantics to the case of answering temporal conjunctive queries over lightweight DL temporal knowledge bases.

1.1. Contributions

This article extends the conference paper [40] on temporal query answering in DL-LiteR over inconsistent data, where the complexity of answering queries with LTL operators, but without negation, over DL-LiteR temporal knowledge bases was investigated. The considered ontologies admit the use of rigid predicates, which are predicates that do not change their interpretation over time. The initial results were obtained for the three inconsistency-tolerant semantics AR, IAR and brave and with respect to three cases of rigid predicates: no rigid predicates, rigid concepts together with rigid roles.

Compared to the conference version, the present article includes new complexity results (all results for \( \mathcal{EL}_\bot \), as well as some results for DL-LiteR). It also extends the set of temporal operators, distinguishing bounded and unbounded variants of the future LTL operators, in order to cover the two different settings that have been investigated for temporal query answering
in the literature, where temporal knowledge bases are interpreted w.r.t. finite or infinite sequences of interpretations. Furthermore, we investigate both cases: the one where negation is admitted in the query language and the one where it is not.

The complexity upper bounds are often obtained by non-deterministic procedures that require, for instance, to guess the right repairs which may not be feasibly computed in practice. Thus algorithms that lend themselves to implementation are still to be devised. We make two contributions toward practical algorithms for temporal inconsistency-tolerant query answering. The first is a polynomial reduction of reasoning in the presence of rigid predicates to reasoning without such predicates by propagating the rigid facts in the sequence of datasets. The second is to identify cases where in the absence of rigid predicates the well-known methods for classical temporal query answering and (atemporal) inconsistency-tolerant query answering can straightforwardly be combined. We show that for the IAR semantics, this yields a sound and complete algorithm. For the AR semantics, such a combination of the algorithms always yields a sound approximation, and additionally yields a sound and complete procedure if the query contains only a restricted set of operators.

This paper is structured as follows. In the next section, we introduce the basic notions of DLs, query answering, inconsistency-tolerant semantics for atemporal knowledge bases and the temporal setting. Furthermore, we discuss known complexity results for both inconsistency-tolerant and temporal query answering in the logics considered. In Section 3, we lift the introduced inconsistency-tolerant semantics to temporal query answering over inconsistent data. Section 4 gives an overview over the complexity results obtained. General versions of algorithms for testing (non-)entailment of temporal conjunctive queries under the different semantics are described in Section 5 in preparation of the complexity analysis. Section 6 shows data and combined complexity of inconsistency-tolerant temporal query answering for DL-LiteR and EL⊥, for the case where the query language admits negation. In Section 7, we complete the complexity picture of temporal query answering under classical semantics by investigating the case where the query does not contain negation. We then built on these results to provide the complexity of inconsistency-tolerant temporal query answering for queries without negation in Section 8. Finally, Section 9 investigates two approaches for practical implementations that allow to employ well-known methods. The article finishes with a section on conclusions and future work.

To improve readability, some of the proofs have been moved to the appendix and are only sketched in the main text.

2. Preliminaries

We briefly recall the syntax and semantics of DLs and the three inconsistency-tolerant semantics that we consider, and then we introduce our setting of temporal query answering.

Syntax. A DL knowledge base (KB) $\mathcal{K}$ consists of an ABox $\mathcal{A}$ and a TBox $\mathcal{R}$, both constructed from three countably infinite sets: a set $\mathcal{N}_C$ of concept names (unary predicates), a set $\mathcal{N}_R$ of role names (binary predicates), and a set $\mathcal{N}_I$ of individual names (constants). The ABox (dataset) is a finite set of concept assertions $A(a)$ and role assertions $R(a, b)$, where $A \in \mathcal{N}_C$, $R \in \mathcal{N}_R$, $a, b \in \mathcal{N}_I$. The TBox (ontology) is a finite set of axioms whose form depends on the particular DL.

In DL-LiteR, TBox axioms are either concept inclusions $B \subseteq C$ or role inclusions $P \subseteq S$, built according to the following syntax, where $A \in \mathcal{N}_C$ and $R \in \mathcal{N}_R$:

$$B := A | \exists P C := B \mid \neg B$$

$$P := R | R^\top S := P \mid \neg P.$$  

Inclusions of the form $B_1 \subseteq B_2$ or $P_1 \subseteq P_2$ are called positive inclusions (PI), those of the form $B_1 \subseteq \neg B_2$ or $P_1 \subseteq \neg P_2$ are called negative inclusions (NI).

In $\mathcal{EL}_{\perp}$, the TBox contains concept inclusions of the form $C_1 \subseteq C_2$, where $C_1$ and $C_2$ are built according to the following syntax, where $A \in \mathcal{N}_C$ and $R \in \mathcal{N}_R$:

$$C := \top | \bot | A | \exists R.C | C \cap C.$$  

An $\mathcal{EL}_{\perp}$ inclusion of the form $C_1 \cap C_2 \subseteq \bot$ can also be written in the form of a negative inclusion $C_1 \subseteq \neg C_2$.

Semantics. An interpretation is a tuple of the form $\mathcal{I} = (\Delta^\mathcal{I}, \mathcal{I})$, where $\Delta^\mathcal{I}$ is a non-empty set and $\mathcal{I}$ maps each $a \in \mathcal{N}_I$ to $a^\mathcal{I} \in \Delta^\mathcal{I}$, each $A \in \mathcal{N}_C$ to $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$, and each $R \in \mathcal{N}_R$ to $R^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$. We adopt the unique name assumption, i.e., for all $a, b \in \mathcal{N}_I$, we require $a^\mathcal{I} \neq b^\mathcal{I}$ if $a \neq b$. The function $\mathcal{I}$ is straightforwardly extended to general concepts and roles, e.g.,

$$\top^\mathcal{I} = \Delta^\mathcal{I}, \bot^\mathcal{I} = \emptyset, (R^\top)^\mathcal{I} = \{(d, e) \mid (e, d) \in R^\mathcal{I}\}.$$
A consistent \( \langle T, A \rangle \) under the different semantics for DL-Lite \( – I \) no free variables (i.e., variables in \( \vec{x} \)). We call \( T \) a model of a KB \( \mathcal{K} = \langle T, A \rangle \) if \( I \) satisfies all axioms in \( T \) and all assertions in \( A \). A KB is consistent if it has a model, and we say that an ABox \( A \) is \( T \)-consistent (or simply consistent if \( T \) is clear from the context), if the KB \( \langle T, A \rangle \) is consistent.

Queries. A conjunctive query (CQ) takes the form \( q = \exists \vec{y} \psi(\vec{x}, \vec{y}) \), where \( \psi \) is a conjunction of atoms of the form \( A(t) \) or \( R(t, t') \), with \( t, t' \) individual names or variables from \( \vec{x}, \vec{y} \). We call the variables in \( \vec{x} \) the free variables in \( q \). A CQ is called Boolean (BCQ) if it has no free variables (i.e., \( \vec{x} = \emptyset \)). A BCQ \( q \) is satisfied by an interpretation \( I = (\Delta^I, \cdot^I) \), written \( I \models q \), if there is a homomorphism \( \pi \) mapping the variables and individual names of \( q \) into \( \Delta^I \) such that: \( \pi(a) = a^I \) for every \( a \in N \), \( \pi(t) \in A^I \) for every concept atom \( A(t) \) in \( \psi \), and \( \pi(t), \pi(t') \) \( \in \Delta^I \) for every role atom \( R(t, t') \) in \( \psi \). A BCQ \( q \) is entailed from \( K \), written \( K \models q \), iff \( q \) is satisfied by every model of \( K \). Given a CQ \( q \) with free variables \( \vec{x} = (x_1, \ldots, x_l) \) and a tuple of individuals \( \vec{a} = (a_1, \ldots, a_k) \), \( \vec{a} \) is a certain answer to \( q \) over \( K \) if \( K \models q(\vec{a}) \), where \( q(\vec{a}) \) is the BCQ resulting from replacing each \( x_i \) in \( q \) by \( a_i \).

Inconsistency-tolerant semantics. A repair of \( K = \langle T, A \rangle \) is an inclusion-maximal subset of \( A \) that is \( T \)-consistent. We consider three semantics based on repairs that have been previously introduced in the literature [29, 30, 33]. A tuple \( \vec{a} \) is an answer to \( q \) over \( K \) under

- AR semantics, written \( K \models_{AR} q(\vec{a}) \), iff \( \langle T, A' \rangle \models q(\vec{a}) \) for every repair \( A' \) of \( K \);
- IAR semantics, written \( K \models_{IAR} q(\vec{a}) \), iff \( \langle T, A'' \rangle \models q(\vec{a}) \) where \( A'' \) is the intersection of all repairs of \( K \);
- brave semantics, written \( K \models_{brave} q(\vec{a}) \), iff \( \langle T, A' \rangle \models q(\vec{a}) \) for some repair \( A' \) of \( K \).

Figure 1 summarizes the complexity of BCQ entailment under the different semantics for DL-Lite\( _2 \) and \( \mathcal{EL} \). Data complexity is measured in the size of the ABox only, while combined complexity is measured in the size of the whole KB and the query. When complexity is measured w.r.t. the size of the KB (ABox and TBox), it is called KB complexity. For DL-Lite\( _2 \) and \( \mathcal{EL} \), CQ answering under the semantics is in \( P \) w.r.t. KB complexity. We refer to Section 4 for a reminder on the definitions of the different complexity classes that appear in this work.

Temporal query answering. We now present our temporal framework inspired from [5] and [18].

Definition 2.1 (TKB). A temporal knowledge base (TKB) \( \mathcal{K} = (\langle T, (\mathcal{A}_i)_{0 \leq i \leq n} \rangle) \) consists of a TBox \( T \) and a finite sequence of ABoxes \( (\mathcal{A}_i)_{0 \leq i \leq n} \). An infinite sequence \( \mathcal{J} = (\langle I_i \rangle)_{i \geq 0} \) of interpretations \( I_i = (\Delta_i, \cdot_i) \) over a fixed non-empty domain \( \Delta \) (constant domain assumption) is a model of \( \mathcal{K} \) iff for every \( i \in [0, n] \), \( I_i \) is a model of \( T \), and for every \( a \in N_0 \) and all \( i, j \geq 0 \), \( a^{\Delta_i} = a^{\Delta_j} \) (rigidity of individual names). Rigid predicates are elements from the set \( N_{\mathcal{R}} \subseteq N_0 \) of rigid concepts and the set \( N_{\mathcal{R}} \subseteq N_0 \) of rigid roles. A sequence of interpretations \( \mathcal{J} = (\langle I_i \rangle)_{i \geq 0} \) respects rigid predicates iff for every \( X \in N_{\mathcal{R}} \cap N_{\mathcal{R}} \) and all \( i \geq 0 \), \( X^{\Delta_i} = X^{\Delta_j} \). A TKB is consistent if it has a model that respects rigid predicates. A sequence of ABoxes \( (\mathcal{A}_i)_{0 \leq i \leq n} \) is \( T \)-consistent, or simply consistent, if the TKB \( \langle T, (\mathcal{A}_i)_{0 \leq i \leq n} \rangle \) is consistent.

Example 2.2. The following \( \mathcal{EL} \) TKB gives information useful to monitor a cluster of servers. The TBox \( T \) defines that a server is overloaded when it executes a process with an increasing workload while its CPU is already maximally utilized. The sequence of ABoxes \( (\mathcal{A}_i)_{0 \leq i = 2} \) records information on a server \( s \). The set of rigid roles is \( N_{\mathcal{R}} = \{ \text{hasCPU} \} \), which models the fact that the CPU of a server does not change over time.

\[
T = \{ \exists \text{executes.} \text{IncreasingWorkload} \sqsubseteq \text{LoadRise}, \\
\text{hasCPU} \sqsubseteq \text{MaxUtilized} \sqsubseteq \text{MaxLoaded}, \\
\text{LoadRise} \sqcap \text{MaxLoaded} \sqsubseteq \text{Overloaded} \}
\]

\[
A_0 = \{ \text{hasCPU}(s, c), \text{MaxUtilized}(c), \\
\text{executes}(s, p_1), \text{IncreasingWorkload}(p_1) \}
\]

\[
A_1 = \{ \text{MaxUtilized}(c), \text{executes}(s, p_1), \\
\text{executes}(s, p_2), \text{IncreasingWorkload}(p_2) \}
\]

\[
A_2 = \{ \text{executes}(s, p_2) \}
\]

It is sometimes convenient to represent a sequence of ABoxes as a set of assertions associated with timestamps, which we call timed assertions: \( (\mathcal{A}_i)_{0 \leq i \leq n} \) then becomes \( \{ (a, i) \mid a \in \mathcal{A}_i, 0 \leq i \leq n \} \).
A rigid concept is a concept which only uses rigid predicates. A rigid assertion is of the form $A(a)$ with $A \in N_{RC}$ or $R(a, b)$ with $R \in N_{RR}$. We distinguish three cases depending on which predicates can be rigid: in the first case, we have no rigid predicates ($N_{RC} = \emptyset$ and $N_{RR} = \emptyset$), in the second case, we do allow only for rigid concepts ($N_{RC} \neq \emptyset$ and $N_{RR} = \emptyset$), and in the last case, we have both, rigid concepts and rigid roles ($N_{RC} \neq \emptyset$ and $N_{RR} \neq \emptyset$). Because rigid concepts can be simulated with rigid roles using a pair of concept inclusions of the form $A \subseteq \exists R, \exists R \subseteq A$, these three cases cover all interesting combinations.

In the remainder of the paper, we always use $n$ to denote the maximal timestamp of a TKB.

**Definition 2.3 (TCQ).** Temporal conjunctive queries (TCQs) are built from CQs as follows: each CQ is a TCQ, and if $\phi_1$ and $\phi_2$ are TCQs, then so are $\neg \phi_1$ (negation), $\phi_1 \land \phi_2$ (conjunction), $\phi_1 \lor \phi_2$ (disjunction), $\diamond \phi_1$ (next), $\bullet \phi_1$ (bounded next), $\circ \phi_1$ (strong previous), $\lozenge \phi_1$ (weak previous), $\Box \phi_1$ (always), $\lozenge \phi_1$ (bounded always), $\square \phi_1$ (always in the past), $\lozenge \phi_1$ (eventually), $\lozenge \phi_1$ (bounded eventually), $\diamond \phi_1$ (some time in the past), $\phi_1 \cup \phi_2$ (until), $\phi_1 \cup^b \phi_2$ (bounded until), and $\phi_1 \cup^s \phi_2$ (since). We further may use $\psi_1 \rightarrow \psi_2$ as a shortcut for $\neg \psi_1 \lor \psi_2$.

We impose the constraint that the past operators $\circ \neg$, $\lozenge \neg$, $\lozenge \neg$ and $S$ cannot be nested under the unbounded future operators $\circ$, $\square$, $\lozenge$ and $U$ (in second position) (see discussion later in this section).

**Example 2.4 (Example 2.2 cont’d).** The following TCQ returns the list of servers which have been overloaded at least twice.

$$\phi = \lozenge \neg (\text{Overloaded}(x) \land \circ \neg \lozenge \neg \text{Overloaded}(x))$$

Note that we can modify our query to speak about a window of 10 time units as follows:

$$\bigcirc^{-10} \lozenge (\text{Overloaded}(x) \land \bigcirc \neg \text{Overloaded}(x))$$

where $\bigcirc^{-10}$ abbreviates a sequence of ten $\bigcirc$ operators. Differently to window operators found in other settings however, subexpressions within such a window can refer to time points outside this window.

Given a TCQ $\phi$, we refer to the TCQs that occur in $\phi$ as subformulas of $\phi$. We sometimes use the notion of the propositional abstraction of a Boolean TCQ $\phi$, which is the propositional LTL formula obtained by replacing each BCQ in $\phi$ by a propositional variable (e.g., the propositional abstraction of $\lozenge \exists x.A(x) \lor \exists x.B(x)$ is the propositional LTL formula $\lozenge x_1 \lor x_2$).

**Remark 2.5 (Choice of operators).** The additional LTL operators $W$ (weak until), $W^-$ (weak since), $R$ (release), and $R^-$ (past release) can be expressed by our set of operators as follows:

\[
\begin{align*}
\phi_1 W \phi_2 & \equiv (\phi_1 U \phi_2) \lor (\Box \phi_1), \\
\phi_1 W^- \phi_2 & \equiv (\phi_1 S \phi_2) \lor (\neg \Box \phi_1), \\
\phi_1 R \phi_2 & \equiv (\phi_1 W (\phi_2 \land \phi_1)), \\
\phi_1 R^- \phi_2 & \equiv (\phi_1 W^-(\phi_2 \land \phi_1)).
\end{align*}
\]

We will consider in Sections 7, 8 and 9 a special setting where TCQs do not contain negation symbols, which sometimes leads to a lower computational complexity. For this reason, we did not introduce $U^b \phi$ as a shortcut for $\neg \diamond \neg \phi$, as it is often done in the literature, but instead treat the operators $\Box$ and $\lozenge$ as native members of our query language. Similarly, since the top and bottom concepts $\top$ and $\bot$ are not allowed in every DL, $\diamond$ (resp. $\lozenge$) cannot be defined using $U$ (resp. $U^b$) as usual in LTL ($\diamond \phi_1 \equiv \text{true} U \phi_1$), unless we allow for negation (where we can express true using $\exists x.A(x) \lor \neg \exists x.A(x)$). We thus keep all these operators in the set we consider.

Note also that since disjunctions are allowed, TCQs could be defined with unions of conjunctive queries (UCQs) instead of CQs. We use CQs for simplicity.

**Definition 2.6 (TCQ answering).** Given a TCQ $\phi$ with free variables $\vec{x} = (x_1, \ldots, x_k)$ and a tuple of individ-
Table 1

<table>
<thead>
<tr>
<th>Satisfaction of BTCQs by a sequence of interpretations</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi$</td>
</tr>
<tr>
<td>$\exists \psi(y)$</td>
</tr>
<tr>
<td>$\neg \phi_1$</td>
</tr>
<tr>
<td>$\phi_1 \land \phi_2$</td>
</tr>
<tr>
<td>$\phi_1 \lor \phi_2$</td>
</tr>
<tr>
<td>$\bigcirc \phi_1$</td>
</tr>
<tr>
<td>$\bigtriangledown \phi_1$</td>
</tr>
<tr>
<td>$\square \phi_1$</td>
</tr>
<tr>
<td>$\neg \square \phi_1$</td>
</tr>
<tr>
<td>$\forall \phi_1$</td>
</tr>
<tr>
<td>$\exists \phi_1$</td>
</tr>
<tr>
<td>$\phi_1 \lor \phi_2$</td>
</tr>
<tr>
<td>$\phi_1 \land \phi_2$</td>
</tr>
</tbody>
</table>

Since $a$ denotes the Boolean TCQ (BTCQ) resulting from replacing each $x_i$ by $a_i$. A tuple in a sequence of interpretations $J = (I_i)_{i \geq 0}$ at time point $p$ iff $J, p \models \phi(a)$. The satisfaction of a BTCQ $\phi$ by a sequence of interpretations $J$ is defined by induction on its structure as shown in Table 1. A tuple $a$ is a certain answer to $\phi$ over $K$ at time point $p$, written $K, p \models \phi(a)$, iff $J, p \models \phi(a)$ for every model $J$ of $K$ that respects rigid predicates.

Example 2.7 (Example 2.2 cont’d). It follows from the semantics of TKBs and TCQs that $s$ is a certain answer to the TCQ $a$ of Example 2.4 over the TKB of Example 2.2 at time point 2 and at time point 1, but not at time point 0. Note that the rigidity of the role hasCPU is crucial to get this answer.

In addition to the standard LTL past and unbounded future operators, we introduce four bounded future operators that mimic the semantics based on finite sequences of interpretations used in [18] and similar to that of LTL on finite traces (see e.g., [41]). Indeed, while the standard way of interpreting TKBs is based on infinite sequences of interpretations, it can be relevant to limit the scope of querying to the known time points, especially in the context of data streams. For instance, a user may want to ask whether a server has been running some process since it started ($\bigtriangledown \neg \text{Start}(s) \land \square \exists \text{execute}(s, p)$), rather than whether it will continue to run this process forever. Moreover, we will see in Section 9.1 that using the bounded semantic can be of practical interest, since it allows us to reduce TCQ answering in the presence of rigid predicates to TCQ answering without rigid predicates. We choose to keep both unbounded and bounded versions of future operators to cover the two settings that have been previously studied for TCQ answering.

The constraint that TCQs should not contain past operators in the scope of unbounded future operators allows us to take advantage of the fact that a TKB entails the same BCQs for every time point $i > n$ to get a lower complexity in the case where there is no negation in the query. Indeed, in this case, the unknown future ($i > n$) can be entirely summarized in one time point $n + 1$. This will also be useful to obtain the data complexity upper bound of brave semantics in the case where there are neither rigid predicates present nor negation in the queries. Moreover, it turns out that for some cases in our analysis, this restriction has no impact on our results. Indeed, [18] shows that Gabbay’s separation theorem [42] can be used to rewrite an LTL formula $\phi$ containing bounded operators into a logically equivalent LTL formula $\phi'$ that is a Boolean combination of pure-past and pure-future formulas, although with an exponential blow-up. It follows that the restriction we impose does not have any influence over the data complexity of TCQ entailment, as long as negation is allowed in the query. Moreover, since past operators can still appear in the scope of bounded future operators, all time points that refer to observations having been made so far ($i \leq n$) can be referenced by the query language. This query language can thus still express most situations that might be meaningful to detect.

It follows from the definition of certain answers that TCQ answering can be straightforwardly reduced to BTCQ entailment (using polynomially many tests w.r.t. data complexity and exponentially many tests w.r.t. combined complexity). For this reason, we focus on the latter problem.

Figure 2 summarizes the complexity of BTCQ entailment for DL-Lite$\mathcal{R}$ and $\mathcal{EL}^-_$. In the different cases depending on which kind of predicates are rigid. Our setting is slightly different from those of [6] and [23], because we have additional bounded operators and the restriction that past operators cannot be nested under unbounded future operators. However, in the results shown in those papers apply to our setting. Indeed, the proofs for the lower bounds do not use past operators nested under future operators, and for the upper bounds, we argue that it is possible to reduce the en-
3. Temporal query answering over inconsistent data

We extend the three inconsistency-tolerant semantics to temporal query answering. The main difference to the atemporal case is that in the presence of rigid concepts or roles, a TKB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}_0 \rangle_{0 \leq i \leq n}$ may be inconsistent even if each KB $\langle \mathcal{T}, \mathcal{A}_i \rangle$ is consistent. Indeed, in this case, there may not be a sequence of interpretations $\mathcal{I} = (\mathcal{I}_i)_{i\geq 0}$ such that $\mathcal{I}_i$ is a model of $\langle \mathcal{T}, \mathcal{A}_i \rangle$ for every $i \in [0, n]$ and which also respects rigid predicates. That is why we have to consider as repairs the $\mathcal{T}$-consistent sequences of subsets of the initial ABoxes that are component-wise maximal.

**Definition 3.1** (Repair of a TKB). A repair of a TKB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}_0 \rangle_{0 \leq i \leq n}$ is a sequence of ABoxes $(\mathcal{A}'_i)_{0 \leq i \leq n}$ such that $\{(a, i) \mid a \in \mathcal{A}'_i, 0 \leq i \leq n\}$ is a maximal $\mathcal{T}$-consistent subset of $\{(a, i) \mid a \in \mathcal{A}_i, 0 \leq i \leq n\}$. We denote the set of repairs of $\mathcal{K}$ by $\text{Rep}(\mathcal{K})$.

The next example illustrates the impact which rigid predicates can have on repairs.

**Example 3.2.** Consider the TKB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}_0 \rangle_{0 \leq i \leq 1}$. The TBox expresses that web servers and application servers are two distinct kinds of servers, and the ABoxes provide information about a server $a$ that executes two processes $b$ and $c$.

$\mathcal{T} = \{\text{WebServer} \sqsubseteq \text{Server}, \text{AppServer} \sqsubseteq \text{Server}, \text{WebServer} \sqsubseteq \neg \text{AppServer}\}$

$\mathcal{A}_0 = \{\text{WebServer}(a), \text{executes}(a, b)\}$

$\mathcal{A}_1 = \{\text{AppServer}(a), \text{WebServer}(a), \text{executes}(a, c)\}$

Assume that no predicate is rigid. The TKB $\mathcal{K}$ is inconsistent because the timed assertions $\text{AppServer}(a, 1)$ and $\text{WebServer}(a, 1)$ violate the negative inclusion in $\mathcal{T}$. Specifically, $\text{AppServer}(a)$ and $\text{WebServer}(a)$ cannot both be true at time point 1. It follows that $\mathcal{K}$ has two repairs that correspond to the two different ways of restoring consistency: $\langle \mathcal{A}'_0 \rangle_{0 \leq i \leq 1}$ and $\langle \mathcal{A}'_1 \rangle_{0 \leq i \leq 1}$, where

$\mathcal{A}'_0 = \mathcal{A}'_1 = \mathcal{A}_0$

$\mathcal{A}'_1 = \{\text{AppServer}(a), \text{executes}(a, c)\}$

$\mathcal{A}'_0 = \{\text{WebServer}(a), \text{executes}(a, c)\}$

Now assume that AppServer is rigid. There is then a new reason for $\mathcal{K}$ being inconsistent: the timed assertions $\langle \text{WebServer}(a, 0) \rangle$ and $\langle \text{AppServer}(a, 1) \rangle$ violate the negative inclusion of $\mathcal{T}$ due to the rigidity of AppServer, which implies that AppServer$(a)$ and WebServer$(a)$ should be both entailed at time point 0. Therefore, $\mathcal{K}$ has now the two repairs $\langle \mathcal{A}'_0 \rangle_{0 \leq i \leq 1}$ and $\langle \mathcal{A}'_1 \rangle_{0 \leq i \leq 1}$, where

$\mathcal{A}'_0 = \{\text{executes}(a, b)\}$

$\mathcal{A}'_1 = \{\text{AppServer}(a), \text{executes}(a, c)\}$

$\mathcal{A}'_0 = \mathcal{A}_0$

$\mathcal{A}'_1 = \{\text{WebServer}(a), \text{executes}(a, c)\}$

Note that even though $\langle \mathcal{A}'_0 \rangle_{0 \leq i \leq 1}$ is maximal (adding WebServer$(a)$ to $\mathcal{A}_0$ renders the TKB inconsistent), $\mathcal{A}'_0$ is not a repair of $\langle \mathcal{T}, \mathcal{A}_0 \rangle$, because it is not maximal.

<table>
<thead>
<tr>
<th>Rigid predicates</th>
<th>Data complexity</th>
<th>Combined complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{N}<em>\mathcal{R} = \mathcal{N}</em>{\mathcal{R}} = \emptyset$</td>
<td>ALOGTIME-complete</td>
<td>PS-complete</td>
</tr>
<tr>
<td>$\mathcal{N}<em>\mathcal{R} \neq \emptyset, \mathcal{N}</em>{\mathcal{R}} = \emptyset$</td>
<td>ALOGTIME-complete</td>
<td>PSPACE-complete</td>
</tr>
<tr>
<td>$\mathcal{N}<em>\mathcal{R} \neq \emptyset, \mathcal{N}</em>{\mathcal{R}} \neq \emptyset$</td>
<td>ALOGTIME-complete</td>
<td>PSPACE-complete</td>
</tr>
</tbody>
</table>

Figure 2. Complexity of BTCQ entailment in DL-Lite$_R$ [6] and $\mathcal{E}_{\mathcal{L}}$ [23]
We extend the semantics AR, IAR, and brave to the temporal case in the natural way by regarding sequences of ABoxes.

Definition 3.3 (AR, IAR, brave semantics for TCQs). A tuple \( \bar{d} \) is an answer to a TCQ \( \phi \) over a TKB \( K = \langle T, \mathcal{A} \rangle_{0 \leq i \leq n} \) at time point \( p \) under

- AR semantics, written \( K, p \models_{AR} \phi(\bar{d}) \), iff \( \langle T, (\mathcal{A})_{0 \leq i \leq n}, p \rangle \models \phi(\bar{d}) \) for every repair \( (\mathcal{A}_i)_{0 \leq i \leq n} \) of \( K \);
- IAR semantics, written \( K, p \models_{IAR} \phi(\bar{d}) \), iff \( \langle T, (\mathcal{A}_i)_{0 \leq i \leq n}, p \rangle \models \phi(\bar{d}) \), with \( \mathcal{A}_i = \bigcap_{i \in \mathbb{N}} (\mathcal{A}_i)_{0 \leq i \leq n} \) for all \( i \in [0, n] \);
- brave semantics, written \( K, p \models_{brave} \phi(\bar{d}) \), iff \( \langle T, (\mathcal{A}_i)_{0 \leq i \leq n}, p \rangle \models \phi(\bar{d}) \) for some repair \( (\mathcal{A}_i)_{0 \leq i \leq n} \) of \( K \).

The following relationships between the semantics, which already hold in the atemporal case, are implied by their definition:

\[ K, p \models_{IAR} \phi(\bar{d}) \Rightarrow K, p \models_{AR} \phi(\bar{d}) \Rightarrow K, p \models_{brave} \phi(\bar{d}) \]

In the following example, we illustrate the effect of the different semantics in the temporal case.

Example 3.4 (Example 3.2 cont’d). Consider the following three temporal conjunctive queries.

\[ \phi_1 = \Box^p (\exists y. \text{executes}(x, y)) \]
\[ \phi_2 = \Box^p (\exists y. \text{Server}(x) \land \text{executes}(x, y)) \]
\[ \phi_3 = \Box^p (\exists y. \text{AppServer}(x) \land \text{executes}(x, y)) \]

If there are no rigid predicates, the intersection of the repairs is \( (\mathcal{A}_i)_{0 \leq i \leq 1} \), with \( \mathcal{A}_0 = \mathcal{A}_0 \) and \( \mathcal{A}_1 = \{ \text{executes}(a, c) \} \). We have \( K, 0 \models_{IAR} \phi_1(a) \), because in every model of the intersection of the repairs \( a \) executes \( b \) at time point 0 and at time point 1. For \( \phi_2 \), \( K, 0 \models_{AR} \phi_2(a) \), since every model of every repair assigns \( a \) to \( \text{Server} \) at time point 0 and to either \( \text{AppServer} \) (in models of \( (\mathcal{A}_i)_{0 \leq i \leq 1} \)) or \( \text{WebServer} \) (in models of \( (\mathcal{A}_i')_{0 \leq i \leq 1} \)) at time point 1. However, \( K, 0 \not\models_{IAR} \phi_2(a) \). Finally, \( K, 0 \not\models_{brave} \phi_3(a) \), because no repair entails \( \text{AppServer}(a) \) at time point 0.

If \( \text{AppServer} \) is rigid, the intersection of the repairs is \( (\mathcal{A}_i)_{0 \leq i \leq 1} \) with \( \mathcal{A}_0 = \{ \text{executes}(a, b) \} \) and \( \mathcal{A}_1 = \{ \text{executes}(a, c) \} \). So, still \( K, 0 \models_{IAR} \phi_1(a) \). Since every model of every repair assigns \( a \) to \( \text{Server} \) at time points 0 and 1 (either because \( a \) is a web server or because \( a \) is an application server), \( K, 0 \models_{AR} \phi_2(a) \), but \( K, 0 \not\models_{IAR} \phi_2(a) \). Finally, \( K, 0 \models_{brave} \phi_3(a) \), because every model of \( (\mathcal{A}_i)_{0 \leq i \leq 1} \) assigns \( a \) to \( \text{AppServer} \) at any time point by rigidity of \( \text{AppServer} \), but \( K, 0 \not\models_{AR} \phi_3(a) \).

We conclude this section by pointing out some characteristics of the case without rigid predicates that will be useful later. If there are no rigid predicates, the interpretations \( I_i \) of a model \( \mathcal{J} = (I_i)_{i \geq 0} \) of \( K \) that respects rigid predicates are independent, besides the interpretation of the individual names. We thus obtain the following proposition.

Proposition 3.5. If \( N_{RG} = N_{RR} = \emptyset \), then a TKB \( K = \langle T, (\mathcal{A})_{0 \leq i \leq n} \rangle \) is consistent iff every \( \langle T, \mathcal{A} \rangle \) is consistent. Moreover, if \( K \) is consistent, for every \( p \in [0, n] \), \( I_p \) is a model of \( \langle T, \mathcal{A} \rangle \) iff there exists a model \( \mathcal{J} = (I_i)_{i \geq 0} \) of \( K \) such that \( I_p = I_p^{\mathcal{J}} \).

Proof. If \( N_{RG} = N_{RR} = \emptyset \), a sequence of interpretations \( \mathcal{J} = (I_i)_{i \geq 0} \) is a model of \( K \) that respects rigid predicates iff it is a model of \( K \). It follows that \( K \) is consistent iff there exists \( \mathcal{J} = (I_i)_{i \geq 0} \) such that for every \( i \in [0, n] \), \( I_i \) is a model of \( \langle T, \mathcal{A} \rangle \), for every \( i > n \) \( I_i \) is a model of \( T \), and for every \( a \in N_t \) and all \( i, j \geq 0 \), \( a^{I_i} = a^{I_j} \). This already establishes one direction of the proposition. We show that also the other direction holds.

Assume that every \( \langle T, \mathcal{A} \rangle \) is consistent, i.e., every \( \langle T, \mathcal{A} \rangle \) has a model. Let \( p \in [0, n] \) be any fixed index and \( I_p^{\mathcal{J}} = (\Delta_1, 2, \Delta_2) \) be a model of \( \langle T, \mathcal{A} \rangle \). Since we adopt the unique name assumption and \( N_t \) is unbounded, \( \Delta_2 \) is an infinite set. Moreover, for every \( i \in [0, n] \), since \( \mathcal{A} \) is finite, by the Löwenheim-Skolem theorem, there exists a model \( I_p'' = (\Delta_1', 2, \Delta_2') \) of \( \langle T, \mathcal{A} \rangle \) such that \( \Delta_2'' \) is countable. Thus \( \Delta_2'' \) can be mapped to \( \Delta_2 \) by an injective function \( f \). For every \( i \), we can then build a model \( I_p'' = (\Delta_1', 2, \Delta_2') \) of \( \langle T, \mathcal{A} \rangle \) such that \( \Delta_2'' = \Delta_2' \). For this, let \( \Delta = \Delta_2'' \backslash f(\Delta_2'' \backslash \Delta_2') \) and select some domain element \( d_1 \in \Delta_1'' \). We define a mapping \( m_1 : \Delta_1'' \to 2^{\Delta_2'} \) by setting \( m_1(d) = (f(d)) \) for \( d \neq d_1 \) and \( m_1(d_1) = \{ f(d_1) \} \cup \Delta \). It remains to set \( a^{I_p''} = f(a^{I_p''}) \) for all \( a \in N_t \), \( A_1'' = \bigcup_{i \in \mathbb{N}} m_i(d) \), and \( R^{I_p''} = \{ (d_1, d_2) \in \Delta_1'' \times \Delta_2' \mid (d_1', d_2') \in R^{I_p''}, d_1 \in m_i(d_1') \} \). Since \( \Delta_2' = \Delta_2'' \) for every \( i \in [0, n] \), there exists a bijection \( b_i : \Delta_2' \to \Delta_2'' \) such that for all \( a \in N_t \), \( b_i(a^{I_p''}) = a^{I_p} \).

Using these bijections, we can construct a model \( \mathcal{J} = (I_i)_{i \geq 0} \) of \( K \) based on the interpretations \( I_0', \ldots, I_n' \) as follows. For every \( i > n \), \( I_i = I_i^{\mathcal{J}} \).
Proposition 3.6. If $\langle T, A, p \rangle$, then for every BQC $q = \exists \exists \exists \exists \exists y \psi(y)$ and $p \in [0, n]$, $K, p \models q$ iff $\langle T, A, p \rangle \models q$.

Proof. $K, p \models q$ iff for every model $\mathcal{F} = (I_i)_{i \geq 0}$ of $K$ that respects rigid predicates, $I_p \models q$. By Proposition 3.5, this is the case iff for every model $I_p$ of $\langle T, A, p \rangle$, $I_p \models q$, which is equivalent to $\langle T, A, p \rangle \models q$. $
$

4. Complexity analysis overview

In the next four sections, we investigate the complexity of inconsistency-tolerant BTCQ entailment in DL-Lite$_R$ and $\mathcal{EL}_\bot$. Apart from the different DLs, we also consider two settings of query languages: in the first setting, all TCQs as defined in Section 2 are considered, in the second setting, we analyze the complexity for TCQs that do not use the negation operator. Furthermore, we investigate complexity depending on whether rigid symbols are present or absent and which of the 4 semantics are used (classical, AR, IAR or brave). All these investigations are carried out for combined or data complexity.

For classical semantics, some complexities have been investigated earlier for the different settings we consider. For the case where negations are allowed in the queries, the complexity of BTCQ entailment under the classical semantics has been studied in [6] for DL-Lite$_R$ and in [23] for $\mathcal{EL}$ (cf. Section 2, Figure 2). Furthermore, it has also been shown in [18, 43] that in DL-Lite$_R$, TCQs without negation (and with a bounded future semantics) can be rewritten into FO-queries for temporal databases, but this was shown only in a restricted setting without rigid names and only for rooted TCQs. In this case, rootedness of a CQ means that it contains at least a constant or an answer variable and all its terms are connected via chains of role atoms. A TCQ is rooted if every CQ in it is rooted.

Most of our complexity upper bounds are based on a set of general algorithms for BTCQ entailment under the different inconsistency-tolerant semantics, which we present in Section 5. Those allow us to obtain complexity upper bounds for the different settings based on the complexity of BTCQ entailment under classical semantics, on the complexity of recognizing repairs, and on the complexity of consistency checking. In Section 6, we establish the complexity of these basic tasks, and give complexity results for our two DLs of interest, $\mathcal{EL}_\bot$ and DL-Lite$_R$, regarding both data and combined complexity. In the cases where the general algorithms are insufficient to give tight bounds, we try to provide specialized algorithms.

We then restrict the query language and study the complexity of the entailment of BTCQs without negation. In Section 7, we first investigate this case under classical semantics, and observe that in some cases, disallowing negation leads to lower worst case complexities, even if we alleviate the limitations imposed in [18, 43]. In Section 8 these lower complexities allow us to also improve the complexity bounds for inconsistency-tolerant reasoning when there is no negation in the TCQs. Furthermore, we take advantage of the absence of negation to tighten an upper bound for brave semantics without rigid predicates.

Synopsis of complexity classes. We recall the definitions of the complexity classes that appear in the following sections.

- $P$: problems solvable in polynomial time.
- $NP$: problems solvable in non-deterministic polynomial time.
- $coNP$: problems whose complement is in $NP$.
- $\Delta^p_2$: problems solvable in polynomial time with at most logarithmically many calls to an NP oracle.
- $\Sigma^p_2$: problems solvable in non-deterministic polynomial time with an NP oracle.
- $\Pi^p_2$: problems whose complement is in $\Sigma^p_2$.
- $\mathcal{AC}^0$: problems that can be solved by a uniform family of circuits of constant depth and polynomial size, with unbounded-fanin AND and OR gates. We have $\mathcal{AC}^0 \subseteq P$.
- $\mathcal{ALOGTIME}$: problems solvable in logarithmic time by a random access alternating Turing machine. We have $\mathcal{AC}^0 \subseteq \mathcal{ALOGTIME} \subseteq P$.
- $\mathcal{PSPACE}$: problems solvable in polynomial space.
- $\mathcal{EXP}$: problems solvable in exponential time.
2. Verify that

1. Guess a sequence \((p_0, p_1, \ldots, p_m)\) of time points decides whether \(\entails_{\text{IAR}}\) holds under brave semantics.

Furthermore, we assume \(K = \langle T, (A_i)_{0 \leq i \leq n} \rangle\) to be the TKB we evaluate our query against, and \(\phi\) to be the considered query.

5. General algorithms for inconsistency-tolerant BTCQ entailment

Our complexity bounds are based on a set of general algorithms for deciding BTCQ entailment under the different semantics, which are inspired from known algorithms for inconsistency-tolerant BCA entailment in the atemporal case (see e.g., [35]). Depending on the semantics under consideration, it might be more natural to describe either a procedure for entailment or one for non-entailment, which is why these algorithms focus on different directions of the problem.

Non-entailment under AR semantics. The procedure \(\text{ARNonEntailment}\) decides whether \(\phi\) is not entailed by \(K\) at time point \(p\) under AR semantics, and is defined as follows.

1. Guess a sequence \((A_i')_{0 \leq i \leq n}\) of ABoxes.
2. Verify that \((A_i')_{0 \leq i \leq n}\) is a repair of \(K\) and that \(\langle T, (A_i')_{0 \leq i \leq n} \rangle, p \not\models \phi\).

It is common to focus on non-entailment when studying the complexity of AR semantics, since this procedure needs to consider only one repair instead of all of them.

Non-entailment under IAR semantics. The procedure \(\text{IARNonEntailment}\) decides whether \(\phi\) is not entailed by \(K\) at time point \(p\) under IAR semantics, and is defined as follows.

1. Guess a sequence \((A_i')_{0 \leq i \leq n}\) of ABoxes.
2. Verify that \((A_i')_{0 \leq i \leq n}\) is a repair of \(K\) and that \(\langle T, (A_i')_{0 \leq i \leq n} \rangle, p \not\models \phi\).

Similarly to the AR case, we use this procedure for non-entailment instead of a procedure for entailment. Deciding non-entailment allows us to consider a linear number of repairs instead of all of them, which can be exponentially many. However, for the IAR semantics, a direct procedure for IAR entailment is useful for showing combined complexity results for the DLs we consider. This procedure is based on the computation of the set of timed assertions that belong to some minimal inconsistent subset of the TKB.

Entailment under IAR semantics. We give an alternative procedure for IAR, \(\text{IAREntailment}\), which decides whether \(\phi\) is entailed under IAR semantics at time point \(p\).

1. For every \((\alpha, j) \in (A_i)_{0 \leq i \leq n}\), check whether \((\alpha, j)\) belongs to a minimal inconsistent subset of \(K\) by asking an oracle whether there exists a \(T\)-inconsistent set of timed assertions \(B \subseteq (A_i)_{0 \leq i \leq n}\) such that \(B \setminus \{(\alpha, j)\}\) is consistent.
2. Call the oracle to determine whether $\phi$ is entailed at time point $p$ by the TKB from which all timed assertions that belong to some minimal inconsistent subset have been removed.

We show that the intersection of the repairs of $K$ is obtained by removing the minimal inconsistent subsets of $K$. Let $B \subseteq (A_i)_{0 \leq i < n}$ be a minimal inconsistent subset of $K$ and $(\alpha, i) \in B$. Since $B \setminus \{(\alpha, i)\}$ is consistent, $(\alpha, i)$ is not in the repairs that contain $B \setminus \{(\alpha, i)\}$. In the other direction, if a timed assertion $(\alpha, i)$ does not appear in some repair $(A_i)_{0 \leq i < n}$ of $K$, since the repairs are maximal, $(A_i)_{0 \leq i < n} \cup \{(\alpha, i)\}$ is inconsistent, and $(\alpha, i)$ occurs in some minimal inconsistent subset of $K$.

6. Complexity of inconsistency-tolerant BTCQ entailment with negation in the query

In this section, we investigate the complexity of BTCQ entailment for general BTCQs, that is, BTCQs that may contain negation. For this, we first establish the complexities of consistency checking and repair recognition, i.e., the task of deciding whether a sequence of ABoxes is a repair of $K$. We then build on these results to prove the complexity of inconsistency-tolerant temporal query entailment using the general algorithms described in the last section, while showing matching lower bounds. We thus obtain the following theorem.

Theorem 6.1. The results in Figure 3 hold.

6.1. Consistency checking and repair recognition for TKBs

We reduce these tasks to the atemporal case by defining an atemporal KB $\tilde{K}$ based on $K$. For $K = \langle T, (A_i)_{0 \leq i < n} \rangle$, $\tilde{K} = \langle \tilde{T}, \tilde{A} \rangle$ is defined in Figure 4. We first show a correspondence between the models of $K$ that respect rigid predicates and the models of $\tilde{K}$.

Lemma 6.2. $K$ is consistent iff $\tilde{K}$ is consistent.

Proof. ($\Leftarrow$) Assume $\tilde{K}$ is consistent, and let $\tilde{I}$ be a model of $\tilde{K}$. We define $\text{temp}(\tilde{I}) = J = (I_i)_{i \geq 0}$ as follows. For every $i \in [0, n]$, we set

$- A^I_i = A^\tilde{I}_i$ for every $A \in \text{NR}$,

and for every $i > n$, we set

$- a^I_i = a^\tilde{I}_i$ for every $a \in N_i$,

$- A^I_i = A^\tilde{I}_i$ for every $A \in \text{NR}$,

$- R^I_i = R^\tilde{I}_i$ for every $R \in \text{NR}$.

We show that the sequence $\text{temp}(\tilde{I})$ of interpretations is a model of $K$ that respects rigid predicates.

1. For every time-point $i \in [0, n]$, $I_i$ is a model of $A_i$.

2. For every $i \in [0, n + 1]$, $I_i$ is a model of $T$.

3. For every $i > 0$, for every $A \in \text{NR}$, $A^I_i = A^\tilde{I}_i$ and $R^I_i = R^\tilde{I}_i$. Therefore, $J$ respects rigid predicates.

We obtain that $\text{temp}(\tilde{I})$ is a model of $K$ that respects rigid predicates.

($\Rightarrow$) For the other direction, we construct a function $\text{atemp}$ from the models of $K$ that respect rigid predicates to those of $\tilde{K}$. Assume $K$ is consistent, and let $J = (I_i)_{i \geq 0}$ be a model of $K$ that respects rigid predicates. We define $\text{atemp}(J) = \tilde{I}$ as follows.

$- a^\tilde{I}_i = a^I_i$ for every $i \geq 0$ for every $a \in N_i$,

$- A^\tilde{I}_i = A^I_i$ for every $A \in \text{NR}$,

$- R^\tilde{I}_i = R^I_i$ for every $R \in \text{NR}$.

$- A^\tilde{I}_i = A^I_i$ for every $A \in \text{NR}$ and $i \in [0, n]$, and
We thus obtain a direct correspondence between the models of $\mathcal{K}$ and those of $\tilde{K}$, and that $\mathcal{K}$ is satisfiable iff $\tilde{K}$ is satisfiable.

It follows that consistency checking of TKBs can be polynomially reduced to consistency checking of KBs.

**Lemma 6.3.** If for a DL $\mathcal{L}$, consistency checking of $\mathcal{L}$-KBs is in $P$, then consistency checking of $\mathcal{L}$-TKBs is in $P$ as well.

**Proof.** By Lemma 6.2, the TKB $\mathcal{K}$ is consistent iff the KB $\tilde{K}$ is consistent. If consistency checking is in $P$ for $\mathcal{L}$-KBs, the consistency of $\mathcal{K}$ can then be checked in time polynomial in the size of $\tilde{T}$ and $\tilde{A}$. Since the size of $\tilde{T}$ is polynomial in $|\mathcal{T}|$ and $n$, and the size of $\tilde{A}$ is at most the size of $(\mathcal{A}_{i})_{0 \leq i \leq n}$, we obtain that TKB consistency checking is in $P$. □

We next show that repair recognition can be done with a polynomial number of consistency checks.

**Lemma 6.4.** If for a DL $\mathcal{L}$, consistency checking of $\mathcal{L}$-TKBs is in $P$, then repair recognition, i.e., deciding whether a sequence of ABoxes $(\mathcal{A}_{j})_{0 \leq j \leq n}$ is a repair of $\mathcal{K}$, is in $P$.

**Proof.** Assume consistency checking of $\mathcal{L}$-TKBs is in $P$. Then, we can verify in $P$ whether a sequence of ABoxes $(\mathcal{A}_{j})_{0 \leq j \leq n}$ is a repair of $\mathcal{K}$ as follows.

1. For every $i$, check that $\mathcal{A}_{i} \subseteq \mathcal{A}_{j}$.
2. Check that $(\mathcal{A}_{j})_{0 \leq j \leq n}$ is $\mathcal{T}$-consistent.
3. For every $(\alpha, j) \in (\mathcal{A}_{j})_{0 \leq j \leq n} \setminus (\mathcal{A}_{j})_{0 \leq j \leq n}$, check that $(\mathcal{A}_{j})_{0 \leq j \leq n} \cup \{ (\alpha, j) \}$ is $\mathcal{T}$-inconsistent. □

Note that Lemmas 6.3 and 6.4 apply to DL-Lite$_R$ and $\mathcal{EL}_L$ alike.

<table>
<thead>
<tr>
<th>\mathcal{EL}_L</th>
<th>\mathcal{L} \setminus \mathcal{EL}_L</th>
</tr>
</thead>
<tbody>
<tr>
<td>classical affinity</td>
<td>AR affinity</td>
</tr>
<tr>
<td>classical affinity</td>
<td>AR affinity</td>
</tr>
</tbody>
</table>

### 3. Data Complexity of BTCQ Entailment for BTCQs with Negations

![Figure 3](image-url) Data complexity and combined complexity of BTCQ entailment for BTCQs with negations. All complexities are tight, except those preceded by “in”, which are upper bounds. The results for the classical semantics are shown in [23] for $\mathcal{EL}_L$ and in [6] for DL-Lite$_R$.

$\mathcal{EL}_L$

| $|p|_{\text{SB}} = |p|_{\text{NB}} = 0$ | $|p|_{\text{SB}} \neq 0, |p|_{\text{NB}} = 0$ | $|p|_{\text{SB}} \neq 0, |p|_{\text{NB}} \neq 0$ |
|-------------------------------|------------------------------|-------------------------------|
| $\Pi_{\text{NP}}$ | $\Pi_{\text{NP}}$ | $\Pi_{\text{NP}}$ |
| $\Pi_{\text{NP}}$ | $\Pi_{\text{NP}}$ | $\Pi_{\text{NP}}$ |

**Figure 4.** KB $\tilde{K} = \langle \mathcal{T}, \tilde{A} \rangle$ representing $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_{i})_{0 \leq i \leq n} \rangle$.

- $\mathcal{A}_{i} =_{\text{def}} \mathcal{A}_{i} \cap \mathcal{N}_{\text{RC}}$
- $\mathcal{A}_{i} =_{\text{def}} \mathcal{A}_{i} \cap \mathcal{N}_{\text{RC}}$
- $\mathcal{A}_{i} =_{\text{def}} \mathcal{A}_{i} \cap \mathcal{N}_{\text{RC}}$
- $\mathcal{A}_{i} =_{\text{def}} \mathcal{A}_{i} \cap \mathcal{N}_{\text{RC}}$

where for every set of axioms $\mathcal{O}$, the function $\text{RenameNotRig}(\mathcal{O}, i)$ substitutes every non-rigid predicate $X$ by $X_i$ in every axiom $\alpha \in \mathcal{O}$.
6.2. Combined complexity

We are now ready to establish the complexity of BTCQ entailment under inconsistency-tolerant semantics. We start with the combined complexity. The following upper bounds follow straightforwardly from the procedures described in Section 5.

**Proposition 6.5.** If repair recognition is in P and BTCQ entailment under the classical semantics is in \(\text{PSPACE}\) w.r.t. combined complexity, then BTCQ entailment under AR, IAR, and brave semantics is in \(\text{PSPACE}\) w.r.t. combined complexity. \(\square\)

**Proof.** If verifying that a sequence of ABoxes is a repair is in P and verifying the entailment, and thus also verifying the non-entailment, of a BTCQ is in \(\text{PSPACE}\), then the procedures ARNonEntailment, IARNonEntailment and braveEntailment all run in \(\text{NPSpace}\). \(\square\)

Proposition 6.5 applies to DL-Lite\(_R\) and \(\mathcal{EL}_\perp\) in all cases except for \(\mathcal{EL}_\perp\) with rigid roles, for which BTCQ entailment under classical semantics is \(\text{CO-NEXPTIME-hard}\) [23].

**Proposition 6.6.** BTCQ entailment from an \(\mathcal{EL}_\perp\)-TKB under AR, IAR, and brave semantics is in \(\text{CNEXPTIME}\) w.r.t. combined complexity, even if \(N_{\text{RR}} \neq \emptyset\). \(\square\)

**Proof.** For the AR and IAR semantics, we modify the procedures ARNonEntailment and IARNonEntailment described in Section 5 so that they also guess a certificate of the non-entailment of \(\phi\) in the first step. Then, in the second step, the non-entailment of \(\phi\) can be decided by simply verifying this certificate. The certificate can be checked in \(\text{EXPTIME}\), since the non-entailment of \(\phi\) can be decided in \(\text{NEXPTIME}\).

For the brave semantics’ upper bound, we give a \(\text{NEXPTIME}\) procedure to decide \(K, p \not\models_{\text{brave}} \phi\). For every subset \((A_i)_{0 \leq i \leq n}\) of \((A_i)_{0 \leq i \leq n}\), guess either “not a repair” or a certificate of the non-entailment of \(\phi\) from \((T, (A_i')_{0 \leq i \leq n})\) at time point \(p\). Note that there are \(2^{|(A_i)_{0 \leq i \leq n}|}\) such subsets. For every such subset, verify in \(\text{EXPTIME}\) whether it is indeed not a repair, or whether \((T, (A_i')_{0 \leq i \leq n}), p \not\models \phi\). \(\square\)

The matching \(\text{PSPACE}\) and \(\text{CNEXPTIME}\) combined complexity lower bounds for \(\mathcal{EL}_\perp\) and DL-Lite\(_R\) follow from the consistent case (cf. Section 2).

6.3. Data complexity for \(\mathcal{EL}_\perp\)-TKBs

We now prove the data complexity results, starting with \(\mathcal{EL}_\perp\). We first consider the case without rigid predicates.

**Proposition 6.7.** BTCQ entailment from an \(\mathcal{EL}_\perp\)-TKB with \(N_{\text{RC}} = N_{\text{RR}} = \emptyset\) is

\[
\begin{align*}
- &\text{coNP-complete w.r.t. data complexity under AR and IAR semantics, and} \\
- &\text{NP-complete w.r.t. data complexity under brave semantics.}
\end{align*}
\]

**Proof.** The upper bounds follow from the procedures described in Section 5: since verifying that a sequence of ABoxes is a repair as well as verifying the non-entailment and entailment of a BTCQ take polynomial time w.r.t. data complexity, the procedures ARNonEntailment, IARNonEntailment, and braveEntailment run in \(\text{NP}\) w.r.t. data complexity. The lower bounds follow from the atemporal case. \(\square\)

Next, we prove the complexity of BTCQ entailment with rigid predicates. The following proposition establishes the upper bounds for the case where both rigid concepts and rigid roles are allowed.

**Proposition 6.8.** BTCQ entailment from an \(\mathcal{EL}_\perp\)-TKB with \(N_{\text{RC}} \neq \emptyset\) and \(N_{\text{RR}} \neq \emptyset\) is

\[
\begin{align*}
- &\text{in coNP w.r.t. data complexity under AR and IAR semantics, and} \\
- &\text{in } \Sigma^p_2 \text{ w.r.t. data complexity under brave semantics.}
\end{align*}
\]

**Proof.** For AR and IAR semantics, we modify the procedures described in Section 5 to also guess a certificate for the non-entailment of \(\phi\). This certificate can be checked in \(\text{P}\), since the non-entailment of \(\phi\) can be decided in \(\text{NP}\). The upper bound for brave semantics is obtained using the procedure braveEntailment described in Section 5. \(\square\)

We show that these results are tight even if we only have rigid concepts.

**Proposition 6.9.** BTCQ entailment from an \(\mathcal{EL}_\perp\)-TKB with \(N_{\text{RC}} \neq \emptyset\) is

\[
\begin{align*}
- &\text{coNP-hard w.r.t. data complexity under AR and IAR semantics, and} \\
- &\text{\(\Sigma^p_2\)-hard w.r.t. data complexity under brave semantics.}
\end{align*}
\]
Proof (Sketch). The lower bounds for AR and IAR semantics follow from the atemporal case. For brave semantics, we show that the complement of brave TCQ entailment is ∃P-hard by reduction from QBF₂₁.

Let Φ = ∃∀x₁,...,xₙ∃y₁,...,yₙ Ψ be a QBF₂₁-formula, where Ψ = ⋃ₖ∈[0,h] ℓₖ where ℓₖ is a 3-CNf formula over the propositional variables \{x₁,...,xₙ,y₁,...,yₙ\}. Based on Φ, define the TKB K = \langle T, (Aᵢ)₀≤i≤3h+2 \rangle and the TCQ ϕ as follows, where N₉ᶜ = \{T\}.

T = \{∃Pos.T ⊑ Sat, ∃Neg.F ⊑ Sat,
   ∃FromPos.Sat ⊑ T, ∃FromNeg.Sat ⊑ F, ∃FromY.Sat ⊑ T \cap F \subseteq ⊥,
   T \cap ∃ValY.T \subseteq ⊥\}

ϕ = ¬□̂₁(NotFirst(c) ∨ Sat(c)) ∨
   ∃Sat(c) \lor ∃⊥Sat(c))

For each clause ℓₖ, ℓₖ + ℓₖ + ℓₖ, i.e., for every i ∈ [0, h], we define the following three ABoxes Aₖ₊ᵢ (0 ≤ k ≤ 2):

Aₖ₊₀ = B ∪ Bₙ₊₀
Aₖ₊₂ = B ∪ Bₙ₊₂ ∪ \{NotFirst(c)\}, 1 ≤ k ≤ 2,

where

B = \{T(xᵢ), F(xᵢ) \mid 1 ≤ j ≤ m\} ∪
   \{ValY(yᵢ, ¬yᵢ) \mid 1 ≤ j ≤ r\}

Bₙ₊₀ = \{Pos(c, xᵢ), FromPos(xᵢ, c) \mid ℓₖ = xᵢ\}

Bₙ₊₁ = \{Neg(c, xᵢ), FromNeg(xᵢ, c) \mid ℓₖ = ¬xᵢ\}

Bₙ₊₂ = \{FromY(yᵢ, c) \mid ℓₖ = yᵢ\}

Bₙ₊₂ = \{FromY(¬yᵢ, c) \mid ℓₖ = ¬yᵢ\}

We use the following claim, which we prove in detail in the appendix, and give the intuition behind the construction here.

Claim. ϕ is valid iff Kᵢ, 0 /∈ brave ϕ.

Recall that (i) ϕ is valid iff for every valuation of the xᵢ, there exists a valuation of the yᵢ that satisfies ϕ, and that (ii) Kᵢ, 0 /∈ brave ϕ iff every repair of K has a model in which ϕ does not hold. The correspondence between valuations of variables of ϕ and models of repairs of K is ensured by B. Intuitively, the rigid concept T is used to encode that a variable is true. The disjunct concept F encodes that a variable is false and is used to enforce that the repairs of K correspond to the evaluations of the xᵢ. The role ValY connects every pair of yᵢ, ¬yᵢ, which ensures together with the inclusion T ∩ ∃ValY.T ⊆ ⊥ that no model of a repair of K assigns both yᵢ and ¬yᵢ to the rigid concept T. As a consequence, each model of a repair of K corresponds to a valuation of the yᵢ.

Each clause ℓₖ + ℓₖ + ℓₖ is represented by the three consecutive ABoxes Aₙ₊₀, A₁₊₁ and A₂₊₂, which each encode a literal of the clause in the Bₙ₊₂ part.

The concept NotFirst is used to indicate that the literal represented is not the first of the clause. The concept Sat encodes the satisfaction of a clause, and the query expresses that it is not the case that every clause is satisfied (the disjunction is true iff the clause is satisfied by at least one of its three literals). Finally, the TBox expresses the conditions under which a clause is satisfied. In particular, the first two inclusions ensure that a clause is always satisfied if it contains a literal xᵢ (resp. ¬xᵢ) assigned to true (resp. false) in the repair, and the next three inclusions ensure that a clause is satisfied only if it contains a literal which is assigned to true (for ¬xᵢ, since the literals xᵢ and ¬xᵢ are represented with only one individual xᵢ, the condition is that xᵢ is assigned to false).

6.4. Data complexity for DL-Liteᵣ-TKBs

It remains to show the data complexity results for DL-Liteᵣ. For AR and brave semantics, the upper bounds follow from the guess and check procedures described in Section 5. (Recall that the complexity of BTCQ entailment under classical semantics is ALOGTIME-complete for this DL, and thus in P.)

The lower bound for AR follows from the atemporal case, which establishes a tight coNP-bound even if N₉ᶜ = N₀ᵣ = ∅.

In contrast, for brave semantics, BCQ entailment is tractable in the atemporal setting. However, we cannot directly extend this result to the temporal case. Indeed, the data complexity upper bound for brave CQ answering in DL-Liteᵣ relies on the fact that the size of the minimal sets of assertions that support the query is bounded by the query size. This is not true in the temporal setting, as can already be seen by the query ϕ = □¬A(a), whose entailment at time point p can depend on p assertions in the TKB. In fact, we show that in the presence of rigid concepts, brave BTCQ entailment becomes NP-hard.

Proposition 6.10. If N₉ᶜ ≠ ∅, then brave BTCQ entailment from DL-Liteᵣ-TKBs is NP-hard w.r.t. data.
complexity. This already holds for BTCQs that do not contain negation.

**Proof.** We show NP-hardness of brave BTCQ entailment from DL-Lite$_R$-TKBs by reduction from SAT. Let $c_0 \land \ldots \land c_n$ be a CNF formula over variables $x_1, \ldots, x_m$. We define the following problem of BTCQ entailment under brave semantics, with two rigid concepts $T$ and $F$. Let $K = \langle T, \langle A_i \rangle_{0 \leq i \leq n} \rangle$ be such that:

$$T = \{ \exists \text{Pos} \subseteq \text{Sat}, \exists \text{Neg} \subseteq \text{Sat}, \exists \text{Pos} \subseteq T, \exists \text{Neg} \subseteq F, T \subseteq \neg F \}$$

$$A_i = \{ \text{Pos}(a, x_j) \mid x_j \in c_i \} \cup \{ \text{Neg}(a, x_j) \mid \neg x_j \in c_i \} \text{ for } 0 \leq i \leq n$$

Let $\phi = \square \neg \text{Sat}(a)$. We show that $c_0 \land \ldots \land c_n$ is satisfiable iff $K, n \models_{\text{brave}} \phi$. Note that $\phi$ does not contain negation. Indeed, since $T$ and $F$ are rigid, a repair $(A_\prime_i)_{0 \leq i \leq n}$ of $K$ is such that each $x_j$ has either only Pos or only Neg incoming edges in $(A_\prime_i)_{0 \leq i \leq n}$, i.e., $x_j$ occurs either only in assertions of the form $\text{Pos}(a, x_j)$ or only in assertions of the form $\text{Neg}(a, x_j)$. We can thus define a valuation $v$ of the variables such that $v(x_j) = \text{true}$ if $(A_\prime_i)_{0 \leq i \leq n}$ does not contain a timed assertion of the form $\text{Pos}(a, x_j)$ or a repair $(A_\prime_i)_{0 \leq i \leq n}$ of $K$ that entails $\phi$ at time point $n$. \( \square \)

**Fact 6.11** ([29], Lemma 1). Due to the DL-Lite$_R$ syntax, the following holds: for every DL-Lite$_R$ TBox $T$, the size of a minimal $T$-inconsistent set of (timed) assertions is at most two.

Therefore, we can always compute the intersection of all repairs in polynomial time.

**Proposition 6.12.** BTCQ entailment from a DL-Lite$_R$-TKB under IAR semantics is in P w.r.t. data complexity, even if $N_{\text{RR}} \neq \emptyset$ and $N_{\text{RC}} \neq \emptyset$.

**Proof.** The size of the minimal $T$-inconsistent subsets of $(A_\prime_i)_{0 \leq i \leq n}$ is bounded by 2. We can thus skip the first step of the procedure IAREntailment described in Section 5 and compute the minimal inconsistent subsets in P by checking the consistency of every timed assertion and pair of timed assertions (with a quadratic number of consistency checks), and then verify the entailment of the query in P w.r.t. data complexity over the TKB from which they have been removed. \( \square \)

7. BTCQ entailment under classical semantics without negation in the query

This section completes the complexity picture for BTCQ entailment under the classical semantics by investigating the case where TCQs do not use negation. We show that the absence of negation in the query induces a complexity drop in several cases. These results are based on a more general property: we show that for any DL $\mathcal{L}$, if $\mathcal{L}$ has the canonical model property for CQ answering over KBs, then $\mathcal{L}$ also has the canonical model property for TCQ answering over TKBs for TCQs without negation. We use the canonical model to prove that for the classical semantics, the complexity upper bounds of the atemporal case transfer to the temporal case.

We first show that BCQ entailment from a TKB $K$ can be reduced to BCQ entailment from the KB $\tilde{K}$ defined Section 6, Figure 4. For this, we define a similar transformation for BCQs as we did for TKBs. Let
Consider
\[ \bar{q}_p = \text{RenameNotRig}(q, p) \]
where \( \text{RenameNotRig}(q, p) \) replaces every non-rigid predicate \( X \) in \( q \) by \( X_p \) if \( p \leq n \), and by \( X_{n+1} \) otherwise.

**Lemma 7.1.** \( K, p \models q \) iff \( \bar{K} \models \bar{q}_p \).

**Proof.** The following proof is written for the case where \( p \in [0, n + 1) \). For the case \( p = n + 1 \), \( p \) is replaced by \( n + 1 \) in the predicates names.

Assume that \( K, p \models q \), and let \( \bar{I} \) be a model of \( \bar{K} \).

Let \( J = (I_0)_{i \geq 0} = \text{temp}(\bar{I}) \) be the corresponding model of \( K \) that respects rigid predicates, as defined in the proof for Lemma 6.2. For any BCQ \( \psi \) without existential variables, we denote by \( \text{atoms}(\psi) \) the set of (ground) atoms of \( \psi \). Since \( I_p \models q \), there then exists a mapping \( \pi \) from the set of constants and variables that appear in \( \psi \) into \( \Delta \) such that for every \( X(d) \in \text{atoms}(\psi) \), if \( X \) is rigid then \( d \in X^\Delta \), and otherwise \( d \in X^\bar{I} \).

Thus, \( \bar{I} \models \text{RenameNotRig}(q, p) \), i.e., \( \bar{I} \models \bar{q}_p \). Hence \( \bar{K} \models \bar{q}_p \).

For the other direction, assume that \( \bar{K} \models \bar{q}_p \) and let \( J = (I_0)_{i \geq 0} \) be a model of \( K \) that respects rigid predicates. Let \( \bar{I} = \text{atemp}(J) \) be as defined in the proof for Lemma 6.2. Since \( \bar{I} \models \bar{q}_p \), there exists a mapping \( \pi \) from the set of constants and variables that appear in \( \psi \) into \( \Delta \) such that for every \( X(d) \in \text{atoms}(\text{RenameNotRig}(\psi, p)) \), we have \( d \in X^\bar{I} \). It follows that for every \( X(d) \in \text{atoms}(\text{renameNotRig}(\psi, p)) \) such that \( X \) is rigid, \( d \in X^\Delta \). Furthermore, we have \( d \in X^\Delta \) for every \( X_p(d) \in \text{atoms}(\text{renameNotRig}(\psi, p)) \) such that \( X \) is not rigid. Thus, \( I_p \models q \), and we obtain \( K, p \models q \).

Moreover, the size of \( \bar{q}_p \) is the same as \( q \). We thus obtain the following lemma.

**Lemma 7.2.** If BCQ entailment from an \( L \)-KB is in \( P \) w.r.t. KB complexity, respectively in \( \text{NP} \) w.r.t. combined complexity, then so is BCQ entailment from an \( L \)-TKB.

**Proof.** If deciding whether \( \bar{K} \models \bar{q}_p \) is polynomial both in \( |T| \) and in \( |A| \), then it is polynomial in \( |T| \) and \( |(A)_{0 \leq i \leq n}| \). It follows that deciding whether \( K, p \models q \) is in \( P \) w.r.t. KB complexity.

Moreover, if deciding whether \( \bar{K} \models \bar{q}_p \) is in \( \text{NP} \) w.r.t. \( |T| \), \( |A| \) and \( |q| \), then verifying a certificate that \( \bar{K} \models \bar{q}_p \) can be done in polynomial time w.r.t. \( |T| \), \( |A| \) and \( |\bar{q}_p| \), so in polynomial time w.r.t. \( |T|, n, |(A)_{0 \leq i \leq n}| \) and \( |q| \). It follows that deciding whether \( K, p \models q \) is in \( \text{NP} \) w.r.t. combined complexity.

Next, we define the notion of canonical model property for BCQ entailment and for entailment of BTCQs without negation.

**Definition 7.3.** (Canonical model property) A DL \( L \) has the canonical model property for BCQ entailment iff for every consistent \( L \)-KB \( \langle T, A \rangle \), there exists a model \( I(T, A) \) such that for every BCQ \( q \), \( \langle T, A \rangle \models q \) iff \( I(T, A) \models q \). We call \( I(T, A) \) the canonical model of KB \( \langle T, A \rangle \).

A DL \( L \) has the canonical model property for entailment of BTCQs without negation iff for any \( L \)-TKB \( \langle T, A \rangle_{0 \leq i \leq n} \), there exists a model \( I(T, A)_{0 \leq i \leq n} \) such that for every BTCQ without negation \( \phi \) and every time point \( p \), holds that

\[ \langle T, A \rangle_{0 \leq i \leq n}, p \models \phi \text{ iff } I(T, A)_{0 \leq i \leq n} \models \phi. \]

We call this model \( I(T, A)_{0 \leq i \leq n} \) the canonical model of TKB \( \langle T, A \rangle_{0 \leq i \leq n} \).

Note that it is justified to speak of the canonical model of a KB or TKB \( K \) because such a model can be homomorphically mapped into any other model of \( K \). Indeed, for every assertion \( \alpha \) built over \( N_0, N_2 \), and \( N_3 \), if \( \alpha \) holds in the canonical model of \( K \) then it also holds in every model of \( K \).

The following theorem gives the relation between the canonical model property for BCQ entailment and the one for BTCQ entailment, and shows why the presence or absence of negation in the query matters.

**Theorem 7.4.** If \( L \) has the canonical model property for BCQ entailment, then \( L \) also has the canonical model property for entailment of BTCQs without negation.

**Proof (Sketch).** Let \( \bar{I}_K \) be the canonical model of \( \bar{K} \) and \( J_K = (I_0)_{i \geq 0} = \text{temp}(\bar{I}_K) \). We show that \( J_K \) is the canonical model of \( K \) for BTCQs without negation, that is, for every BTCQ \( \phi \) that does not contain any negation, \( K, p \models \phi \) iff \( J_K, p \models \phi \).

Since \( J_K \) is a model of \( K \) that respects rigid predicates, if \( K, p \models \phi \) then \( J_K, p \models \phi \). For the other di-
rection, we show in the appendix by induction on the structure of $\phi$ that if $J_K, p \models \phi$, then $K, p \models \phi$. The interesting case is when $\phi$ is a BCQ. For a BCQ $q$, by Lemma 7.1, $K, p \models q$ iff $\bar{K} \models \bar{q}_p$, which is exactly the case iff $\bar{J}_K \models \bar{q}_p$. By construction of $J_K$, it follows that $K, p \models q$ iff $J_K, p \models q$.

**Remark 7.5.** If $\phi$ contains negation, the preceding argument fails and $J_K$ is not a canonical model for TCQ answering, i.e., $J_K, p \models \neg \phi$ does not guarantee that $J, p \models \phi$ for every model $J$ that respects rigid predicates. As an example, consider the empty TKB, i.e., $T = \emptyset$ and $A_i = \emptyset$ for every $i \in [0..n]$. We have $J_K, 0 \not\models \exists x.A(x)$, but we can easily construct a model $J$ for $\langle T, (A_i)_{0 \leq i \leq n}\rangle$ such that $J, 0 \models \exists x.A(x)$, so that $K, p \not\models \exists x.A(x)$.

The following proposition is a direct consequence of the existence of a canonical model for entailment of BTCQs without negation.

**Proposition 7.6.** Assume $\mathcal{L}$ has the canonical model property for BCQ entailment. Then, for every $\mathcal{L}$-TKBs $K$ and $K'$, if $K$ and $K'$ coincide for BCQ entailment, then $K$ and $K'$ coincide for entailment of BTCQs without negation, i.e., if for every time point $p$ and BCQ $q$, $K, p \models q$ iff $K', p \models q$, then for every time point $p$ and BTCQ $\phi$ without negation, $K, p \models \phi$ iff $K', p \models \phi$.

**Proof (Sketch).** We show this result in the appendix by induction on the structure of $\phi$. We use the fact that by Theorem 7.4, $\mathcal{L}$ has the canonical model property for entailment of BTCQs without negation, to apply the definitions of BTCQ satisfaction of Table 1 to the canonical models of $K$ and $K'$.

We now prove a central proposition for TCQ answering over TKBs in DLs that have the canonical model property for entailment of BTCQs without negation. It amounts to reducing the entailment of BTCQs with unbounded future operators to the entailment of BTCQs with only bounded future operators. This can then be answered by considering only a finite number of time points.

Let $K^*$ be the following TKB:

$$K^* = \langle T, (A_i)_{0 \leq i \leq n} \cup (A_{n+1}) \rangle \text{ where}$$

$$A_{n+1} = \{A(a) \mid A \in \text{NT}, A(a) \in \bar{A}\} \cup$$

$$\{R(a, b) \mid R \in \text{NT}, R(a, b) \in \bar{A}\}.$$

**Proposition 7.7.** If $\mathcal{L}$ has the canonical model property for BCQ entailment, the relations in Table 2 hold for every $\mathcal{L}$-TKB $K$.

**Proof.** It is easy to see that $J_K = J_K$ holds by construction. We obtain that $K, p \models \phi$ iff $K^*, p \models \phi$ for every BTCQ without negation $\phi$.

All relations in Table 2 but those for the operators $\Box$, $\Diamond$ and $U$ are straightforwardly obtained by applying the definitions of BTCQ satisfaction of Table 1 to this canonical model. To show the three remaining relations, we rely on the fact that $J_K$ is such that for every $i > n$, $J_i = J_{i+1}$, and there are no past operators nested under unbounded future operators by definition of TCQs. Indeed, if a BTCQ $\phi$ does not contain any past operators and $i > n$, $J_K, i \models \phi$ iff $J_{K^*}, n + 1 \models \phi$. Hence we can show the following, from which the relations from Table 2 then follow straightforwardly.

- $K^*, p \models \Box \phi_1$ iff for every $k \geq p$, $K^*, k \models \phi_1$. Hence $K^*, p \models \Box \phi_1$ iff $\forall k, p \leq k \leq n, K^*, k \models \phi_1$ and $K^*, n + 1 \models \phi_1$.
- $K^*, p \models \Diamond \phi_1$ iff there exists $k \geq p$, $K^*, k \models \phi_1$. Hence $K^*, p \models \Diamond \phi_1$ iff $\exists k, p \leq k \leq n, K^*, k \models \phi_1$ or $K^*, n + 1 \models \phi_1$.
- $K^*, p \models \phi_1 \cup \phi_2$ iff $\exists k \geq p, K^*, k \models \phi_2$ and $\forall j, p \leq j < k, K^*, j \models \phi_1$. Hence $K^*, p \models \phi_1 \cup \phi_2$ iff $\exists k, p \leq k \leq n, K^*, k \models \phi_2$ and $\forall j, p \leq j < k, K^*, j \models \phi_1$, or $K^*, n + 1 \models \phi_2$ and $\forall j, p \leq j < n + 1, K^*, j \models \phi_1$.

In the next theorem, we transfer complexity upper bounds from the atemporal case to the temporal case, even with rigid predicates, for queries without negation and DLs that have the canonical model property for BCQ entailment. We consider DLs for which BCQ entailment is in P w.r.t. KB complexity and in NP w.r.t. combined complexity, such as DL-Lite\(\mathbb{Q}\) and $\mathcal{L}_{\ominus\downarrow}$.

**Theorem 7.8.** If $\mathcal{L}$ has the canonical model property for BCQ entailment and is such that BCQ entailment from KBs is in P w.r.t. KB complexity, respectively in NP w.r.t. combined complexity, then the entailment of BTCQs without negation from $\mathcal{L}$-TKBs is in P w.r.t. KB complexity, respectively in NP w.r.t. combined complexity.

**Proof.** By Lemma 7.2, it is possible to decide whether $K^*, p \models q$ in P w.r.t. KB complexity for any BCQ $q$. Based on this, we can show by induction on the structure of $\phi$ that $K^*, p \models \phi$ can be decided in P w.r.t. KB complexity. Assume that for two BTCQs $\phi_1, \phi_2$
and any $p \geq 0$, it is possible to decide in P whether $K^*, p \models \phi$. Using the relations in Table 2, we can prove the following.

- $K^*, p \models \phi_1 \land \phi_2$ iff $K^*, p \models \phi_1$ and $K^*, p \models \phi_2$.
- Therefore, deciding whether $K^*, p \models \phi_1 \land \phi_2$ can be done in P by checking that $K^*, p \models \phi_1$ and $K^*, p \models \phi_2$.

- $K^*, p \models \phi_1 \lor \phi_2$ iff $K^*, p \models \phi_1$ or $K^*, p \models \phi_2$.
- Therefore, deciding whether $K^*, p \models \phi_1 \lor \phi_2$ can be done in P by deciding whether $K^*, p \models \phi_1$ and $K^*, p \models \phi_2$ and checking that at least one is true.

- $K^*, p \models \Box \phi_1$ iff $K^*, p + 1 \models \phi_1$.
- Therefore, deciding whether $K^*, p \models \Box \phi_1$ can be done in P by checking whether $K^*, p + 1 \models \phi_1$.

- $K^*, p \models \Diamond \phi_1$ iff $p < n$ implies $K^*, p + 1 \models \phi_1$.
- Therefore, deciding whether $K^*, p \models \Diamond \phi_1$ can be done in P by deciding whether $p \geq n$ or $K^*, p + 1 \models \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Box \phi_1$ and whether $K^*, p \models \Box \phi_1$.

- $K^*, p \models \Diamond \phi_1$ iff for every $k$, $p \leq k \leq n + 1$, $K^*, k \models \phi_1$.
- Therefore, deciding whether $K^*, p \models \Diamond \phi_1$ can be done in P by checking for each $p \leq k \leq n + 1$ that $K^*, k \models \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Box \phi_1$ and whether $K^*, p \models \Box \phi_1$.

- $K^*, p \models \Diamond \phi_1$ iff there exists $k$, $p \leq k \leq n + 1$, $K^*, k \models \phi_1$.
- Therefore, deciding whether $K^*, p \models \Diamond \phi_1$ can be done in P by deciding for each $p \leq k \leq n + 1$ whether $K^*, k \models \phi_1$, and checking that this is true for at least one $k$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.

- We show in the same way that we can decide in P whether $K^*, p \models \Diamond \phi_1$ and whether $K^*, p \models \Diamond \phi_1$.
i > n + 1, we can show similar relations as those in Table 2 for the entailment of LTL formulas without past operators nested under unbounded future operators or negations from w. We can then use a similar induction as we used in the proof for the data complexity to show that w, p |= φ can be decided by checking which queries are in w'. For this, the number of queries to be tested is polynomial in n and the size of φ.

As a consequence of Theorems 7.4 and 7.8, and since \(\mathcal{EL}_1\) and DL-Lite\(\mathcal{R}\) have the canonical model property for BCQ entailment (cf. [11] for \(\mathcal{EL}_1\), and [44] for DL-Lite\(\mathcal{R}\)), we obtain the following theorem.

**Theorem 7.9.** For DL-Lite\(\mathcal{R}\) and \(\mathcal{EL}_1\), entailment of BTCQs without negation is in P w.r.t. KB complexity and in NP w.r.t. combined complexity, even if \(N_{\text{RR}} \neq \emptyset\).

Besides these results for DL-Lite\(\mathcal{R}\) and \(\mathcal{EL}_1\), Theorems 7.4 and 7.8 hold for all Horn-DLs satisfying the complexity constraints in the precondition of Theorem 7.8. For instance, they also hold for DL-Lite\(\mathcal{N}_{\text{horn}}\) [45].

### 8. Complexity of inconsistency-tolerant BTCQ entailment without negation in the query

The following proposition gives general complexity upper bounds for BTCQ entailment under the AR, IAR, and brave semantics. By Theorem 7.9, they hold in particular for \(\mathcal{L} = \mathcal{EL}_1\) and \(\mathcal{L} = \text{DL-Lite}\(\mathcal{R}\)\), provided no negation is used in the TCQs.

**Proposition 8.1.** If \(\mathcal{L}\) is such that consistency checking of a \(\mathcal{L}\)-TKB is in P and BTCQ entailment from a \(\mathcal{L}\)-TKB is in P w.r.t. data complexity and in NP w.r.t. combined complexity, then BTCQ entailment from a \(\mathcal{L}\)-TKB

\[\begin{align*}
&\text{under AR semantics is in } \text{coNP w.r.t. data complexity and in } \Pi^p_2 \text{ w.r.t. combined complexity;} \\
&\text{under IAR semantics is in } \text{coNP w.r.t. data complexity and in } \Delta^p_2[O(\log n)] \text{ w.r.t. combined complexity;} \\
&\text{under brave semantics is in } \text{NP w.r.t. data complexity and in } \Pi^p_2 \text{ w.r.t. combined complexity.}
\end{align*}\]

**Proof.** The data complexities follow from the procedures described in Section 5: since verifying that a sequence of ABoxes is a repair as well as non-entailment and entailment of a BTCQ can be decided in polynomial time, ARNonEntailment, IARNonEntailment and braveEntailment take non-deterministic polynomial time.

For the combined complexity of brave BTCQ entailment, a certificate that \((T, (A_i^j)_{0 \leq j \leq n}), p \models \phi\) can be guessed together with \((A_i^j)_{0 \leq j \leq n}\) and verified in P.

For the combined complexity of IAR, we use the procedure IARENtallment from Section 5 with an NP oracle to decide whether a timed assertion belongs to some minimal inconsistent subset in the first step, and to decide the entailment of \(\phi\) in the last step. Since the oracle calls can be structured as a tree, this procedure gives membership in \(\Delta^p_2[O(\log n)]\) [46].

For \(\mathcal{EL}_1\), matching lower bounds for all semantics follow from the atemporal case [28, 35].

For DL-Lite\(\mathcal{R}\), we can obtain matching lower bounds from the atemporal case for the combined complexity of all semantics as well as for the data complexity of the AR semantics [29, 33]. Regarding IAR semantics, entailment of BTCQs with negations under IAR semantics is already in P (see Figure 3), so this better upper bound applies. Finally, we show that for brave semantics and DL-Lite\(\mathcal{R}\), in the case where there are no rigid predicates, we can improve the NP upper bound of Figure 3 to a P bound. Recall that the lower bound established by Proposition 6.10 for brave semantics with rigid predicates already applies to BTCQs without negations, which means that no improvement of our bounds is possible in the presence of rigid predicates. The only remaining case is entailment under brave semantics where \(N_{\text{RC}} = N_{\text{RR}} = \emptyset\).

We describe a method for brave entailment of BTCQ without negation when \(N_{\text{RC}} = N_{\text{RR}} = \emptyset\). Our procedure proceeds by type elimination over a set of tuples built from the query and that represent the TCQs that are entailed at each time point.

First, we define the structure on which the method operates. We consider the set \(BCQ(\phi)\) of leaves of \(\phi\), that is, the set of all BCQs in \(\phi\), and the set \(F(\phi)\) of subformulas of \(\phi\). In what follows, we identify the BCQs of \(BCQ(\phi)\) and the BTCQs of \(F(\phi)\) with their propositional abstractions as defined in Section 2: if we write that a KB or a TKB entails some elements of \(BCQ(\phi)\) or \(F(\phi)\), we consider them as BCQs or BTCQs, and if we write that some elements of \(BCQ(\phi)\) or \(F(\phi)\) entail others, we consider the elements of \(BCQ(\phi)\) as propositional variables and those of \(F(\phi)\) as propositional LTL formulas built over these variables.

**Definition 8.2.** A brave-justification structure \(J\) for the BTCQ without negation \(\phi\) in the TKB \(K\) is a set of tuples of the form \((i, L_{\text{now}}, F_{\text{now}}, F_{\text{prev}}, F_{\text{next}})\), where
0 ≤ i ≤ n, \( L_{\text{now}} \subseteq BCQ(\phi) \), \( F_{\text{now}} \subseteq F(\phi) \), \( F_{\text{prev}} \subseteq F(\phi) \), and \( F_{\text{next}} \subseteq F(\phi) \).

Note that the size of a brave-justification structure for \( \phi \) in \( K = \langle T, (A_i)_{0 ≤ i ≤ n} \rangle \) is linearly bounded in \( n \) and independent of the size of the ABoxes. A tuple \( (i, L_{\text{now}}, F_{\text{now}}, F_{\text{prev}}, F_{\text{next}}) \) is justified in \( J \) iff it fulfills all of the following conditions.

1. \( \langle T, A_i \rangle \models \text{brave} \bigwedge q \in L_{\text{now}} q \).
2. If \( i > 0 \), there exists \( (i-1, L_{\text{now}}', F_{\text{now}}', F_{\text{prev}}', F_{\text{next}}') \in J \) such that \( F_{\text{prev}} = F_{\text{now}}' \) and \( F_{\text{now}} = F_{\text{next}}' \).
3. If \( i < n \), there exists \( (i+1, L_{\text{now}}', F_{\text{now}}', F_{\text{prev}}', F_{\text{next}}') \in J \) such that \( F_{\text{next}} = F_{\text{now}}' \) and \( F_{\text{now}} = F_{\text{prev}}' \).
4. For every \( \psi \in BCQ(\phi) \), if \( F_{\text{now}} \models \psi \), then \( \psi \in L_{\text{now}} \).
5. For every \( \psi \in F(\phi) \), if \( F_{\text{now}} \models \psi \), then \( \psi \in F_{\text{now}} \).
6. For every \( \psi \in F(\phi) \), if \( \bigwedge q \in L_{\text{now}} q \wedge \bigwedge q \in F_{\text{prev}} q \wedge \bigwedge q \in F_{\text{next}} q \) then \( \psi \in F_{\text{now}} \).
7. For every \( \psi, \psi' \in F(\phi) \):
   - if \( \psi \vee \psi' \in F_{\text{now}} \), then either \( \psi \in F_{\text{now}} \) or \( \psi' \in F_{\text{now}} \).
   - if \( \psi \psi' \in F_{\text{now}} \), then either \( \psi \in F_{\text{now}} \) or \( \psi' \in F_{\text{next}} \).
   - if \( \psi \psi' \in F_{\text{now}} \), then either \( \psi \in F_{\text{now}} \) or \( \psi' \in F_{\text{next}} \).
   - if \( \psi' \psi' \in F_{\text{now}} \), then either \( \psi \in F_{\text{next}} \) or \( \psi' \in F_{\text{next}} \).
   - if \( \psi' \psi' \in F_{\text{now}} \), then either \( \psi \in F_{\text{next}} \) or \( \psi' \in F_{\text{next}} \).
8. If \( i = n \), then for all \( \psi \in F(\phi) \) that are of the form \( \Box \varphi \), \( \psi \in F_{\text{now}} \).
9. If \( i = 0 \), then for all \( \psi \in F(\phi) \) that are of the form \( \Box \varphi \), \( \psi \in F_{\text{now}} \).

We give the intuition behind the elements of the tuples that fulfill these conditions. The first element \( i \) is the time point we are considering, \( L_{\text{now}} \) is a set of BCQs whose conjunction is entailed under brave semantics by \( \langle T, A_i \rangle \) (Condition 1), and \( F_{\text{now}} \) is the set of formulas that can be entailed together with \( L_{\text{now}} \), depending on what is entailed in the previous and next time points, this information being stored in \( F_{\text{prev}} \) and \( F_{\text{next}} \) respectively (Condition 6). Conditions 2 and 3 ensure that there is a sequence of tuples representing every time point from 0 to \( n \) such that this information is coherent between consecutive tuples. Condition 4 expresses that \( L_{\text{now}} \) is exactly the set of BCQs contained in \( F_{\text{now}} \) and Condition 5 that \( F_{\text{now}} \) is maximal in the sense that it contains its consequences. Condition 7 enforces that \( F_{\text{now}}, F_{\text{prev}} \) and \( F_{\text{next}} \) respect the semantics of LTL operators and Conditions 8 and 9 enforce this semantics at both ends of the finite sequence. (Note that we use here the fact that past operators cannot be nested under unbounded future operators, and that no BCQ can be entailed under brave semantics after time point \( n \) in the absence of rigid predicates.)

A brave-justification structure \( J \) is correct if every tuple is justified, and \( \phi \) is justified at time point \( p \) by \( J \) if there is \( \langle p, L_{\text{now}}, F_{\text{now}}, F_{\text{prev}}, F_{\text{next}} \rangle \in J \) such that \( \phi \in F_{\text{now}} \). We show that \( \phi \) is entailed from \( K \) at time point \( p \) under brave semantics iff there is a correct brave-justification structure for \( \phi \) in \( K \) that justifies \( \phi \) at time point \( p \). The main idea of the proof is to link the tuples of a sequence \( \langle (i, L_{\text{now}}, F_{\text{now}}, F_{\text{prev}}, F_{\text{next}}) \rangle_{0 ≤ i ≤ n} \) to a consistent TKB \( K' = \langle T, (C_i)_{0 ≤ i ≤ n} \rangle \) such that for every \( i \), we have \( C_i \subseteq A_i \) and \( \langle T, C_i \rangle \models \bigwedge q \in L_{\text{now}} q \). We show in the appendix that there is such a \( K' \) such that \( K', p \models \phi \) iff there is such a sequence of tuples that is a correct brave-justification structure for \( \phi \) in \( K \) and justifies \( \phi \) at time point \( p \).

The data complexity of brave entailment of BTCQ without negation when there are no rigid predicates follows from the characterization of brave BTCQ entailment with brave-justification structures.

**Proposition 8.3.** For DL-Lite\( _R \), if \( N_{\text{RC}} = N_{\text{RR}} = \emptyset \), then entailment of BTCQs without negation under brave semantics is in \( \mathbb{P} \) w.r.t. data complexity.

**Proof.** We describe a polynomial procedure that decides the existence of a brave-justification structure for \( \phi \) in \( K \) that justifies \( \phi \) at time point \( p \). We start with a brave-justification structure \( J \) for \( \phi \) in \( K \) that contains all possible tuples. Note that the size of this initial structure \( J \) is linearly bounded in the number \( n \) of
ABoxes, and otherwise independent of the data size. The reason is that, except for the time point, the elements of each justification tuple, namely the sets $L_{\text{now}}$, $F_{\text{now}}$, $F_{\text{prev}}$ and $F_{\text{next}}$, only depend on the query.

From the initial, maximal, brave-justification structure, we remove the unjustified tuples one after the other as follows. (i) Remove every tuple that does not satisfy Conditions 1, 4, 5, 6, 7, 8 or 9, and (ii) repeat the following steps until a fix-point has been reached: iterate over the tuples from time point 0 to $n$, eliminating those which do not satisfy Condition 3, and then iterate from $n$ to 0 eliminating those which do not satisfy Condition 2. For the resulting brave-justification structure, we check whether it contains a tuple $(p, L_{\text{now}}, F_{\text{now}}, F_{\text{prev}}, F_{\text{next}})$ such that $\phi \in F_{\text{now}}$. If yes, we return “entailed at time point $p$”, otherwise, we return “not entailed at time point $p$”. Since the size of $J$ is linear in $n$, this process requires at most quadratically many steps. The verification that a given tuple is justified requires polynomial time w.r.t. data complexity (the verification of Condition 3 and Condition 2 is linear in $n$, and only the brave entailment of a BCQ from a DL-Lite$_R$-KB for Condition 1 depends on the size of the ABox, which can be performed in $\mathsf{AC}^0$ w.r.t. data complexity). Therefore, the complete procedure runs in polynomial time w.r.t. data complexity.

9. Toward practical algorithms

Until now, work on TCQ answering has primarily focused on complexity analysis for different DL languages [5, 6, 22]. Attempts towards practical algorithms or implementations are as of now scarce [43, 47]. The only attempt toward more practical algorithms close to our scenario that we are aware of has been made for DL-Lite$_R$ and TCQs without negation in [43], and partially implemented [48]. Some of the results have then been generalized in [18] to query languages that are re writable in the atemporal case.

In this section, we mainly focus on DL-Lite$_R$-TKBs and TCQs without negation, building on this previous work. However, some of our results also apply to other DLs and we discuss the case of $\mathcal{EL}_{\bot}$.

Three different algorithms for answering TCQs without negation over DL-Lite$_R$-TKBs without rigid predicates are provided in [18, 43]. The first algorithm rewrites the TCQ into a query in ATSQL [49], an SQL variant for temporal databases. The second algorithm first rewrites the TCQ into an equivalent TCQ that does not contain future operators, and then iteratively computes the answers for each time point. The third algorithm computes the answers of the TCQ iteratively as well, but does not eliminate the future operators beforehand. For this, it uses a data structure called answer formulas, which represents the TCQs in which some parts have already been evaluated. This structure contains sets of already computed answers to subqueries, as well as variables that serve as place-holders for subqueries that have to be evaluated at the next time point.

Our first contribution is a method for handling rigid predicates (both concepts and roles) in polynomial time for TCQ answering over DL-Lite$_R$-TKBs under the classical semantics. Indeed, [18, 43] consider only rigid concepts (but not rigid roles) for which they provide a method that is restricted to TCQs that are rooted, i.e., in which each CQ contains an individual or an answer variable that is connected to all the other terms through roles. As a second contribution, we show that in the absence of rigid predicates, it is sometimes possible to combine the algorithms for inconsistency-tolerant query answering in the atemporal case with algorithms for temporal query answering in the consistent case in order to perform inconsistency-tolerant temporal query answering.  

9.1. TCQ answering under classical semantics with rigid predicates for DL-Lite$_R$ and TCQs without negation or unbounded future operators

In this section, we show how TCQ answering with rigid predicates can be reduced to TCQ answering without rigid predicates, enabling us to use the algorithms that have been proposed for this latter case. Throughout the section, $\mathcal{K}$ is a DL-Lite$_R$-TKB and $\phi$ a TCQ without negation and unbounded future operators $(\land, \lor, \neg, \exists)$. This restriction amounts to using the setting of [18, 43] in which the semantics is defined w.r.t. finite sequences of interpretations, and is necessary to reduce TCQ answering with rigid predicates to
TCQ answering without rigid predicates. Indeed, consider for instance the query $\square A(a)$. Such a query can be entailed with rigid predicates, e.g., if $A$ is rigid, but not without rigid predicates since for $p > n$, the interpretation of every predicate is empty in the $p^{th}$ component of the canonical model of a TKB without rigid predicates.

To the best of our knowledge, the only algorithm that has been proposed for TCQ answering with rigid predicates and aims at practicality is described as well in [18, 43], and deals only with rigid concepts and rooted TCQs. We briefly describe this algorithm, which aims at handling streaming data by computing the answers to the query at the last available time point. The key idea is to check all sets of potentially rigid concept assertions and test their compatibility with each of the ABoxes from the sequence together with the TBox. Unfortunately, the original algorithm omits the test whether the checked set of rigid concept assertions covers also the rigid information from the tested ABox together with the TBox. As we found this small flaw in the original algorithm, we present here a mended variant.

In the following, let $N^K_C$, $N^K_R$, $N^K_RC$, $N^K_RB$, and $N^K_I$ denote respectively the sets of concepts, roles, rigid concepts, rigid roles, and individuals that occur in the TKB $K$. The algorithm first constructs every possible set $R$ of assertions built from the rigid concepts and individuals in the TKB $K$. Note that there are $2^{N^K_C+|N^K_R|}$ such sets. It then runs, in parallel, for each such set $R$ an instance of the algorithm for TCQ answering without rigid predicates on the TKB that is obtained by adding the assertions in $R$ to every ABox of the TKB. For each time point $i$, it takes into account the new dataset available by eliminating the incompatible instances, i.e., those for which

1. $\langle T, A_i \cup R \rangle$ is inconsistent, or
2. a rigid concept assertion entailed by $\langle T, A_i \rangle$ does not belong to $R$.\footnote{This condition is new and added after consultation with the authors of [18, 43].}

The answers at time point $i$ are then obtained by taking the intersection of the answers returned by all active instances.

We follow a similar idea in the sense that we also add assertions to the TKB that propagate the effects of the rigid predicates. We show that this way, for DL-Lite$_R$, TCQ answering with rigid predicates can be reduced to TCQ answering without rigid predicates in polynomial time.

In order to show that TCQ answering with rigid predicates can be reduced to TCQ answering without, we construct in polynomial time a set $R$ of assertions that captures all information about rigid concepts and roles that is relevant for consistency checking and TCQ answering. Then, TCQ answering over $K$ with $N^K_RC \neq \emptyset$, $N^K_RB \neq \emptyset$ can be performed by TCQ answering over $\langle T, (A_i \cup R)_{0 \leq i \leq n} \rangle$ with $N^K_RC = N^K_RB = \emptyset$. Without any restriction on the TBox, $R$ may be infinite, as illustrated in the following example.

**Example 9.1.** Consider the TKB $K$ with the TBox $T = \{ \exists R^- \sqsubseteq \exists R, R \sqsubseteq S \}$, where $S$ is rigid, and the ABox sequence consisting of $A_0 = \{ R(a, b) \}$, and $A_i = \emptyset$ for $i \in [1, n]$. 
Every model of $\mathcal{K}$ that respects rigid predicates satisfies $\phi = \exists x_1 \ldots x_{k+1}. S(x_1, x_2) \land \cdots \land S(x_k, x_{k+1})$ for every $k > 0$ and at every time point. Since with $N_{\text{RC}} = N_{\text{RR}} = \emptyset$, $\mathcal{K}$ entails such a query only at time point 0, $\mathcal{R}$ should be such that $\langle \mathcal{T}, \mathcal{R} \rangle$ entails such a query, so that $\langle \mathcal{T}, (\mathcal{A} \cup \mathcal{R}) \rangle_{0 \leq i \leq n}$ entails it at every time point. Moreover, there exist models of $\mathcal{K}$ that respect rigid predicates and for which neither $\exists x_1 \ldots x_k.S(x_1, x_2) \land \cdots \land S(x_k, x_1)$ nor $\exists x.R(x, y)$ hold at any time point $i > 0$. Therefore, $\mathcal{R}$ cannot contain cycles of $S$, nor $R$-assertions. Consequently, $\mathcal{R}$ has to contain an infinite chain of $S$-assertions.

This problem motivates us to disallow rigid roles that have non-rigid sub-roles. In other words, we restrict ourselves in the following to TBoxes $\mathcal{T}$ that entail no role inclusions of the form $P_1 \sqsubseteq P_2$ with $P_1 := R_1^\prime \setminus R_1 \cap N_{\text{RR}}$ and $P_2 := R_2 \setminus R_2 \cap N_{\text{RR}}$. This condition avoids chains of rigid roles in the anonymous part of the canonical model $\mathcal{J}_\mathcal{K}$ that cannot be entailed by a single rigid assertion. In the example above, if rigid roles are only allowed to have rigid sub-roles, then $\mathcal{R}$ has to be rigid. In this case, adding the single assertion $R(x, y)$ to every $\mathcal{A}_i$ is sufficient for $\exists x_1 \ldots x_k.R(x_1, x_2) \land \cdots \land R(x_k, x_{k+1})$ to be entailed at every time point and for every $k > 0$.

As a first step, we explicitly construct the canonical model $\mathcal{J}_\mathcal{K}$ of the DL-Lite$^R$-TKB $\mathcal{K}$. This model will be used to prove that $\mathcal{K}$ with $N_{\text{RC}} \neq \emptyset$, $N_{\text{RR}} \neq \emptyset$ and $\langle \mathcal{T}, (\mathcal{A} \cup \mathcal{R}) \rangle_{0 \leq i \leq n}$ with $N_{\text{RC}} = N_{\text{RR}} = \emptyset$ entail the same BTOCs that do not have negation or unbounded future operators.

We build a sequence of (possibly infinite) ABoxes $\langle \text{chase}^\mathcal{K}_{\text{rig}}(\mathcal{A}_i) \rangle_{0 \leq i \leq n+1}$ similar to the chase presented in [50] for KBs. Let $\mathcal{S}$ be a set of DL-Lite$^R$ assertions. We say a positive inclusion (PI) $\alpha$ is applicable in $\mathcal{S}$ to an assertion $\beta \in \mathcal{S}$ if one of the following conditions is satisfied:

- $\alpha = A_1 \sqsubseteq A_2, \beta = A_1(a) \land A_2(a) \notin \mathcal{S}$,
- $\alpha = A \sqsubseteq \exists P, \beta = A(a)$ and there exists no $b$ such that $P(a, b) \in \mathcal{S}$,
- $\alpha = \exists P \sqsubseteq A, \beta = P(a, b) \land A(a) \notin \mathcal{S}$,
- $\alpha = \exists P_1 \sqsubseteq \exists P_2, \beta = P_1(a_1, a_2)$ and there exists no $b$ such that $P_2(a_1, b) \in \mathcal{S}$, or
- $\alpha = P_1 \sqsubseteq P_2, \beta = P_1(a_1, a_2)$, and $P_2(a_1, a_2) \notin \mathcal{S}$.

A PI $\alpha$ is applied to an assertion $\beta$ by adding a new assertion $\beta_{\text{new}}$ to $\mathcal{S}$ such that $\alpha$ is not applicable to $\beta$ in $\mathcal{S} \cup \{ \beta_{\text{new}} \}$ anymore.

**Definition 9.2** (Rigid chase of a TKB). Let $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{0 \leq i \leq n} \rangle$ be a DL-Lite$^R$-TKB. Let $\langle \mathcal{A}_i \rangle_{0 \leq i \leq n+1}$ be such that $\mathcal{A}_i = \mathcal{A}_0 \cup \{ \beta \mid \exists k, \beta \in \mathcal{A}_k \text{ and } \beta \text{ is rigid} \}$ for $i \in [0, n]$ and $\mathcal{A}_{n+1} = \emptyset$. Finally, let $\mathcal{T}_n$ be the set of positive inclusions in $\mathcal{T}$, and $N_i$ be the number of assertions in $\mathcal{A}_i$. Assume that the assertions in each $\mathcal{A}_i$ are enumerated from $N_i + 1 \ldots N_i + N$ to $N_i + 1 \ldots N_i + N$, following their lexicographic order. Consider the sequences of sets $\mathcal{S}^i = (\mathcal{S}_i)_{0 \leq i \leq n+1}$ of assertions defined by

$$
\mathcal{S}^0 = (\mathcal{A}_i)_{0 \leq i \leq n+1},
\mathcal{S}^{i+1} = \mathcal{S}^i \cup \mathcal{S}_{\text{new}} = (\mathcal{S}_i \cup \mathcal{S}_{\text{new}})_{0 \leq i \leq n+1},
$$

where $\mathcal{S}_{\text{new}}$ is defined in terms of the assertion $\beta_{\text{new}}$ obtained as follows: let $\beta \in \mathcal{S}^i$ be the first assertion in $\mathcal{S}^i$ such that there exists a PI in $\mathcal{T}_n$ applicable in $\mathcal{S}_i^i$ to $\beta$ and let $\alpha$ be the lexicographically first PI applicable in $\mathcal{S}_i^i$ to $\beta$. $\beta_{\text{new}}$ is then defined based on the syntactical form of $\alpha$ and $\beta$. Specifically,

- if $\alpha = A_1 \sqsubseteq A_2$ and $\beta = A_1(a)$, then $\beta_{\text{new}} = A_2(a)$,
- if $\alpha = A \sqsubseteq \exists P \land \beta = A(a)$, then $\beta_{\text{new}} = P(a, \beta_{\text{new}})$,
- if $\alpha = \exists P \sqsubseteq A \land \beta = P(a, b)$, then $\beta_{\text{new}} = A(a)$,
- if $\alpha = \exists P_1 \sqsubseteq \exists P_2 \land \beta = P_1(a_1, a_2)$, then $\beta_{\text{new}} = P_2(a_1, a_2)$,
- if $\alpha = P_2 \sqsubseteq P_1$ and $\beta = P_1(a_1, a_2)$, then $\beta_{\text{new}} = P_2(a_1, a_2)$,
- if $\alpha = P_1 \sqsubseteq A$, then $\beta_{\text{new}} = A(a)$,
- if $\alpha = A \sqsubseteq \exists P$, then $\beta_{\text{new}} = A(a)$.

where $\beta_{\text{new}}$ is constructed from $\alpha$ and $\beta$ as follows:

- if $\alpha \in N_i^\mathcal{K}$, then $\beta_{\text{new}} = \beta_{\alpha} \beta_{\alpha}$, and
- otherwise $\alpha \in N_i^\mathcal{K}$, then $\beta_{\text{new}} = \beta_{\alpha} \beta_{\alpha}$, and
- we define $\beta_{\text{new}} = \beta_{\alpha} \beta_{\alpha}$.

If $\beta_{\text{new}}$ is rigid, then $\mathcal{S}_{\text{new}} = (\{ \beta_{\text{new}} \})_{0 \leq i \leq n+1}$, otherwise, $\mathcal{S}_{\text{new}} = (\mathcal{S}_{\text{new}})_{0 \leq i \leq n+1}$ with $\mathcal{S}_{\text{new}} = \{ \beta_{\text{new}} \}$ and $\mathcal{S}_{\text{new}} = \emptyset$ for $i \neq i_j$.

Let $N$ be the total number of assertions in $\mathcal{S}^i$. The assertion(s) added are numbered as follows: if $\beta_{\text{new}}$ is not rigid, $\beta_{\text{new}}$ is numbered by $N + 1$, otherwise for every $i \in [0, n + 1]$, the assertion $\beta_{\text{new}} \in \mathcal{S}_{\text{new}}$ added to $\mathcal{S}^i$ is numbered by $N + 1 + i$.

We call the rigid chase of the TKB $\mathcal{K}$, denoted by $\text{chase}^\mathcal{K}_{\text{rig}}(\mathcal{K}) = (\text{chase}^\mathcal{K}_{\text{rig}}(\mathcal{A}_i))_{0 \leq i \leq n+1}$, the sequence of sets of assertions obtained as the infinite union of all $\mathcal{S}^i$, i.e.,

$$(\text{chase}^\mathcal{K}_{\text{rig}}(\mathcal{A}_i))_{0 \leq i \leq n+1} = (\bigcup_{j \in \mathbb{N}} \mathcal{S}^j)_{0 \leq i \leq n+1}.$$
Based on the rigid chase of $K$, we construct the sequence of interpretations $J_K = \langle I_i \rangle_{i \geq 0}$, where $I_i = \langle \Delta_i, T_i \rangle$ is defined as follows:

- $\Delta = N^K_0 \cup \Gamma_N$, where $\Gamma_N$ is the set of individuals that appear in $\text{chase}_{rg}(K)$ and not in $K$.
- For every $a \in \Delta$, $a^{T_i} = a$.
- For every $A \in N_G$, $A^{T_i} = \{a \mid A(a) \in \text{chase}_{rg}^K(A_i)\}$ if $i \leq n$, $A^{T_i} = \{a \mid A(a) \in \text{chase}_{rg}^K(A_{n+1})\}$ if $i > n$.
- For every $R \in N_R$, $R^{T_i} = \{(a, b) \mid R(a, b) \in \text{chase}_{rg}^K(A_i)\}$ if $i \leq n$, $R^{T_i} = \{(a, b) \mid R(a, b) \in \text{chase}_{rg}^K(A_{n+1})\}$ if $i > n$.

We show that $J_K$ is a model of $K$ that respects rigid predicates, and that for any BTCQ without negation $\phi$ such that $N^K_0 \subseteq N^K_1$, $K, p \models \phi$ if $J_K, p \models \phi$.

Lemma 9.3. If $K$ is consistent, then $J_K$ is a model of $K$ that respects rigid predicates.

Proof (Sketch). Since for every $i \in [0, n]$, $A_i \subseteq \text{chase}_{rg}^K(A_i)$, we directly obtain $I_i \models A_i$. We can show that for every $i$, $I_i$ further satisfies every PI in $T$ with similar arguments as those used in [50]. Indeed, every PI applicable to an assertion $\beta$ in $S^j_i$ at step $j$ of the construction of the rigid chase becomes not applicable to $\beta$ in $S^k_i$ for some $k \geq j$, because there are neither infinitely many assertions before $\beta$, nor infinitely many PIs applied to some assertion that precedes $\beta$. Finally, we show that $I_i$ satisfies every negative inclusion of $T$ because otherwise $K$ would be inconsistent. Moreover, the model $J_K$ respects rigid predicates because, if an assertion $\beta$ of $\text{chase}_{rg}^K(A_i)$ is rigid, either $\beta \in A_i$ and by construction $\beta \in S^0_i = A'_i$ for every $k$, or $\beta$ has been derived at some step $j$ by applying some PI to an assertion of $S^j_i$ and $\beta \in S^k_{i+1}$ for every $k$. We obtain that in both cases, $\beta \in \text{chase}_{rg}^K(A_k)$ for every $k$.

Next, we show that $J_K$ is the canonical model of $K$ for entailment of BTCQ without negation.

Lemma 9.4. If $K$ is consistent, then for every BTCQ without negation $\phi$ such that $N^K_0 \subseteq N^K_1$, $K, p \models \phi$ iff $J_K, p \models \phi$.

Proof (Sketch). Since $J_K = \langle I_i \rangle_{i \geq 0}$ with $I_i = \langle \Delta_i, T_i \rangle$ is a model of $K$ that respects rigid predicates, the first direction is straightforward, and we only need to show that $J_K, p \models \phi$ implies $K, p \models \phi$. Let $J = \langle I_i \rangle_{i \geq 0}$ with $I'_i = \langle \Delta'_i, T'_i \rangle$ be a model of $K$ that respects rigid predicates. We show by structural induction on $\phi$ that if $J_K, p \models \phi$, then $J', p \models \phi$. For the case where $\phi$ is a CQ $\exists y. \psi(\vec{y})$, we show that if there exists a homomorphism $h$ of $\exists y. \psi(\vec{y})$ into $I_p$, then $I'_p \models \exists y. \psi(\vec{y})$, by defining a homomorphism $h$ from $\Delta$ into $\Delta'$.

We are now ready to introduce the set $\mathcal{R}$ that, if added to every ABox of the TKB, allows us to reduce TCQ answering with rigid predicates to TCQ answering without.

Proposition 9.5. Let $\mathcal{R}$ be as defined in Figure 6. Then, $\mathcal{R}$ is computable in polynomial time and such that

1. $K$ is consistent iff $K_{\mathcal{R}} = \langle T, (A_i \cup \mathcal{R}) \rangle_{0 \leq i \leq n}$ is consistent with $N_{RC} = N_{RR} = \emptyset$, and
2. for any BTCQ $\phi$ without negation and unbounded future operators and such that $N^K_0 \subseteq N^K_1$, $K, p \models \phi$ iff $K_{\mathcal{R}}, p \models \phi$ with $N_{RC} = N_{RR} = \emptyset$.

The size of $\mathcal{R}$ is polynomial in the size of $N^K_0$, $N^R_0$, and $N^K_1$, and since query answering as well as subsumption checking are in P, $\mathcal{R}$ can be computed in polynomial time. The first three parts of $\mathcal{R}$ contain information about the participation of individuals of $N^K_1$ in rigid predicates. The last two capture the participation in rigid predicates of the role-successors w.r.t. non-rigid roles, and thus take into account also anonymous individuals that are created in $\text{chase}_{rg}(K)$ when applying PIs whose right-hand side is an existential restriction of a non-rigid role. The individuals created in $\text{chase}_{rg}(K)$ when applying such a PI with a rigid role are witnessed by the individuals $x_{p\alpha}$ or $x_{p\partial}$ if they do not follow from a rigid role assertion, and do not need to be witnessed otherwise, since the assertion $P_2(x_{p1\alpha}, x_{p1\partial})$ is sufficient to trigger the generation of the whole anonymous part implied by the fact that $x_{p1\alpha}$ is in the range of $P_2$.

We split the proof of Proposition 9.5 into several lemmas.

Lemma 9.6. $K$ is consistent iff $K_{\mathcal{R}}$ is consistent with $N_{RC} = N_{RR} = \emptyset$.

Proof. By Proposition 3.5, $K_{\mathcal{R}}$ is consistent with $N_{RC} = N_{RR} = \emptyset$ iff each $\langle T, A_i \cup \mathcal{R} \rangle$ is consistent. We show that also $K$ is consistent iff each $\langle T, A_i \cup \mathcal{R} \rangle$ is consistent.

For the case where $K$ is not consistent, let $B$ be a minimal inconsistent subset of $\mathcal{R}$. Recall that by Fact 6.11, $B$ contains at most 2 assertions. Then $B$ is either internal to some $A_i$, or is of the form $B = \{(\alpha, i), (\beta, j)\}$ with $i \neq j$. In the former case, $\langle T, A_i \cup \mathcal{R} \rangle$ is clearly inconsistent. In the latter case,
\[ \mathcal{R} = \{ A(a) \mid A \in \mathbb{N}^\text{RC} \cup \mathbb{N}^\text{RR}, a \in \mathbb{N}^\text{K}, \text{ and } \exists i, \langle T, A_i \rangle \models A(a) \} \cup \\
\{ R(a, b) \mid R \in \mathbb{N}^\text{RC} \cup \mathbb{N}^\text{RR}, a, b \in \mathbb{N}^\text{K}, \text{ and } \exists i, \langle T, A_i \rangle \models R(a, b) \} \cup \\
\{ P(a, x, p) \mid P := R, R \in \mathbb{N}^\text{RC} \cup \mathbb{N}^\text{RR}, a \in \mathbb{N}^\text{K}, \text{ and } \exists i, \langle T, A_i \rangle \models \exists x.P(a, x) \} \cup \\
\{ A(x, p_1) \mid A \in \mathbb{N}^\text{RC} \cup \mathbb{N}^\text{RR}, p_1 := S \mid S^-, S \in \mathbb{N}^\text{K} \setminus \mathbb{N}^\text{RR}, T \models \exists P_1^- \subseteq A, \text{ and } \exists i, \langle T, A_i \rangle \models \exists x.P_1(x, y) \} \cup \\
\{ P_2(x, p_1, p_2) \mid P_2 := R, R \in \mathbb{N}^\text{RC} \cup \mathbb{N}^\text{RR}, p_1 := S \mid S^-, S \in \mathbb{N}^\text{K} \setminus \mathbb{N}^\text{RR}, T \models \exists P_1^- \subseteq P_2 \text{ and } \\
\exists i, \langle T, A_i \rangle \models \exists x.P_1(x, y) \} \]

Figure 6. Set of rigid assertions added to every ABox of \( \mathcal{K} \)

\( \{ \alpha, \beta \} \) violates some negative inclusion in the closure of the TBox that involves at least a rigid concept \( A \) or a rigid role \( R \) by assigning an individual \( a \) (or two individuals \( a, b \)) to two disjoint concepts (or roles). We can then assume w.l.o.g. that we are in one of the following cases: (i) \( \langle T, A \rangle \models A(a) \), (ii) \( \langle T, A \rangle \models \exists x.R(a, x) \), (iii) \( \langle T, A \rangle \models \exists x.R(x, a) \), or (iv) \( \langle T, A \rangle \models R(a, b) \). It follows that respectively (i) \( \langle T, A_i \rangle \models A(a) \), (ii) \( \langle T, A_i \rangle \models \exists x.R(a, x) \), (iii) \( \langle T, A_i \rangle \models \exists x.R(x, a) \), or (iv) \( \langle T, A_i \rangle \models R(a, b) \). By construction of \( \mathcal{R} \), we then conclude that (i) \( A(a) \in \mathcal{R} \), (ii) \( R(a, x, p) \in \mathcal{R} \), (iii) \( R(x, a) \in \mathcal{R} \), or (iv) \( R(a, b) \in \mathcal{R} \) respectively, and therefore that \( \langle T, A_i \cup \mathcal{R} \rangle \) is inconsistent.

For the other direction, assume there exists \( i \in [0, n] \), such that \( \langle T, A_i \cup \mathcal{R} \rangle \) is inconsistent, and let \( \mathcal{B} \) be a minimal inconsistent subset of \( \langle T, A_i \cup \mathcal{R} \rangle \). If \( \mathcal{B} \) is internal to \( A_i \), \( \mathcal{K} \) is clearly inconsistent. Otherwise, \( \mathcal{B} \) is of the form \( \{ \alpha, \beta \} \) and involves at least one assertion from \( \mathcal{R} \). The assertions \( \alpha \) and \( \beta \) assign an individual \( x \) to two disjoint concepts \( C_1 \) or \( C_2 \), or a pair of individuals \( x, y \) to two disjoint roles \( R_1, R_2 \). We distinguish three cases.

1. In the case where \( x = x_{ap} \) (resp. \( x = x_{p1p2} \)), since \( P(a, x, p) \) (resp. \( P_2(x, p_1, p_2) \)) is the only assertion of \( \mathcal{R} \) that contains \( x \), we obtain that \( \langle T, \{ P(a, x, p) \} \rangle \) (resp. \( \langle T, \{ P_2(x, p_1, p_2) \} \rangle \)) is inconsistent, which implies that \( P \) (resp. \( P_2 \)) is unsatisfiable, i.e., has an empty interpretation in every model of \( T \). Since there exists \( j \) such that \( \langle T, A_j \rangle \models \exists x.P(a, x) \) (resp. \( \langle T, A_j \rangle \models \exists x.y.P_1(x, y) \) and \( T \models \exists P_1^- \subseteq P_2 \)), it follows that \( A_j \) is inconsistent.

2. In the case where \( x = x_{p1} \), since \( x_{p1} \) appears only in concepts that subssume \( \exists P_1^- \), the fact that \( x \) is assigned to two disjoint concepts implies that \( \exists P_1^- \) is unsatisfiable. Thus, and since there exists \( j \) such that \( \langle T, A_j \rangle \models \exists y.P_1(x, y) \), \( A_j \) is inconsistent.

3. Finally, in the case where \( x \in \mathbb{N}^\text{K} \), since \( \alpha \) or \( \beta \) is in \( \mathcal{R} \), at least one of \( C_1, C_2 \) (or \( R_1, R_2 \)) is rigid. As the case where some ABox \( A_i \) is inconsistent is trivial, we assume every \( A_i \) to be consistent. If \( \alpha \notin \mathcal{A} \), let \( e_\alpha \in A_i \) be an assertion responsible for the entailment that triggered the addition of \( \alpha \) to \( \mathcal{R} \), and otherwise let \( (e_\alpha, j_\alpha) = (a, i) \). If \( \beta \notin \mathcal{A} \), let \( e_\beta \in A_i \) be an assertion responsible for the entailment that triggered the addition of \( \beta \) to \( \mathcal{R} \), and otherwise \( (e_\beta, j_\beta) = (b, i) \). Then \( \{ (e_\alpha, j_\alpha), (e_\beta, j_\beta) \} \) is inconsistent because \( e_\alpha \) and \( e_\beta \) lead to \( (a, b) \) being assigned to two disjoint concepts (or disjoint roles) such that at least one of them is rigid. As a result, \( \mathcal{K} \) is inconsistent as well.

We obtain that in every case, if \( \mathcal{K}_{\mathcal{R}} \) is inconsistent with \( \mathbb{N}^\text{RC} = \mathbb{N}^\text{RR} = \emptyset \), then \( \mathcal{K} \) is inconsistent with rigid predications.

We now assume that \( \mathcal{K} \) and \( \mathcal{K}_{\mathcal{R}} \) are consistent. Note that if this is not the case, they both trivially entail any BTCQ. The following two lemmas show that if a

Lemma 9.8. Let \( q = \exists y.\psi(\mathcal{I}) \) be such that \( N_R \subseteq N^K \). For every \( p \in [0, n] \), if \( \mathcal{K}, p \models q \) then \( \mathcal{K}_R, p \models \psi \).

Proof (Sketch). Again, we make use of Lemma 9.4 and use the canonical model \( J_\mathcal{K} = (\mathcal{I}_x)_{x \geq 0} \). The lemma can then be shown by considering some model \( I_p \) of \( (\mathcal{T}, A_p \cup R) \), and defining a homomorphism of \( J_p \) into \( I_p \).

Since by Lemmas 9.7 and 9.8, \( \mathcal{K} \) and \( \mathcal{K}_R \) with \( N_{R_C} = N_{R_R} = \emptyset \) coincide on the entailment of BCQs for every time point \( p \in [0, n] \), we can show as in Proposition 7.6 that they coincide on entailment of BTCQs without negation nor unbounded future operators.

Lemma 9.9. Let \( \phi \) be a BTCQ with no negation or unbounded future operators such that \( N_R \subseteq N^K \). Then, \( \mathcal{K}, p \models \phi \) if \& only if \( \mathcal{K}_R, p \models \phi \) with \( N_{R_C} = N_{R_R} = \emptyset \).

It follows that TCQs can be answered in \( \mathcal{K} \) with rigid predicates by answering TCQs in \( \mathcal{K}_R \) without rigid predicates and pruning answers that contain individual names not in \( N_R \). Note that every model of \( \mathcal{K}_R \) is a model of \( \mathcal{K} \), but does not respect rigid predicates in general. We can reduce BTCQ entailment over \( \mathcal{K} \) with rigid predicates to BTCQ entailment over \( \mathcal{K}_R \) without rigid predicates only because our TCQs do not allow LTL operators to be nested in existential quantifications. This prevents existentially quantified variables to link different time points. To see this, consider the query \( \exists x. \Box \phi(R(a, x) \land R(x, y)) \) and the TKB \( \mathcal{K} = (\mathcal{T}, (A_i)_{i \geq i_R}) \) with \( \mathcal{T} = \{ B \subseteq \exists R, \exists R^- \subseteq \exists R \}, \) and \( R \in N_{R_C} \) and \( A_i = \{ B(a) \} \). For this TKB, we would have \( R = \{ R(a, x_R) \} \), and therefore \( x_R \) could have a different \( R \)-successors in each interpretation of a model of \( \mathcal{K}_R \), thus \( y \) cannot be mapped to the same object at every time point.

Remark 9.10. In the case of streaming data, if we want to take into account a newly available dataset, we do not need to fully recompute \( \mathcal{R} \): we only need to add the new rigid assertions that can be derived from the new dataset. Moreover, if we only reason over a window of \( n \) time points from our stream, we can annotate the assertions in \( \mathcal{R} \) with a counter that is initialised with \( n \) and decremented with each new time point. If an existing assertion is derived again, it is reset to \( n \). Assertions are then removed from \( \mathcal{R} \) if their counter reaches \( 0 \).

Remark 9.11. The main goal of the approaches presented in [18, 43] for TCQ answering in DL-Lite\(_R\) is to obtain the query answers at the last time point without storing all the data for all previous time points. Their algorithm uses a bounded history encoding, which means that the space required by the algorithm is constant w.r.t. the number \( n \) of previous time points: only the current dataset and some auxiliary relations required for computing the query answers are stored and updated at each time point.

Unfortunately, with rigid predicates present, our approach does not achieve bounded history encoding, since the answers of the subqueries of \( \phi \) at previous time points may change when new rigid assertions are derived from the last dataset. However, if the algorithm of [18, 43] has this property, it requires exponential space w.r.t. \( N_{R_C} \) and \( N_{R_R} \) which can also be problematic, while our algorithm requires only polynomial space and time. To achieve bounded history encoding (but in exponential time w.r.t. \( N_{R_C}, N_{R_R} \) and \( N^K \)), we could adapt the algorithm of [18, 43] to support rigid roles. We would consider all possible sets \( \mathcal{R} \) built from \( N_{R_C}, N_{R_R} \) and \( N^K \) following the form of Figure 6, then verify at each time point whether \( \mathcal{R} \) is consistent with \( A_i \) and \( T \) and contains all rigid assertions that can be derived from \( A_i \) as described in Figure 6.

Our procedure only applies to TBoxes in which rigid roles do not have non-rigid subroles. A possible direction to alleviate these restrictions would be to use ideas similar to those developed in [44] for CQ answering over DL-Lite\(_R\)-KB using the combined approach. This CQ answering approach saturates the data by adding to the ABox every assertion that can be derived, introducing individual names to witness existential role restrictions, and then uses a special rewriting to prune spurious answers. In our setting, we could model infinite chains of rigid roles by adding cycles of rigid roles to \( \mathcal{R} \), then prune the spurious answers resulting from these cycles.

Regarding \( \mathcal{E}L \), we conjecture that we could have a similar approach for rigid predicates. The main difference would be that since in \( \mathcal{E}L \) several assertions may be needed to derive one, \( \mathcal{R} \) would have to be computed iteratively, taking into account its own assertions to derive new ones only if a fix-point is reached. Moreover, the problem of infinite chains of rigid roles that cannot be entailed by a polynomial set of assertions would appear as soon as \( N_{R_R} \neq \emptyset \). The combined approach for \( \mathcal{E}L \) [11] could provide ideas to overcome this difficulty.
Table 3

Entailment under classical or IAR semantics from a $\mathcal{L}$-TKB without rigid predicates and such that $\mathcal{L}$ has the canonical model property for BCQ entailment

<table>
<thead>
<tr>
<th>$\phi$</th>
<th>$\mathcal{K}, p \models_{\phi}$ S iff</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\exists \vec{y} \psi(\vec{y})$</td>
<td>$p \leq n$ and $\langle T, \mathcal{A}<em>p \rangle \models</em>{\phi} \exists \vec{y} \psi(\vec{y})$</td>
</tr>
<tr>
<td>$\phi_1 \land \phi_2$</td>
<td>$\mathcal{K}, p \models_{\phi_1}$ and $\mathcal{K}, p \models_{\phi_2}$</td>
</tr>
<tr>
<td>$\phi_1 \lor \phi_2$</td>
<td>$\mathcal{K}, p \models_{\phi_1}$ or $\mathcal{K}, p \models_{\phi_2}$</td>
</tr>
<tr>
<td>$\Box \phi_1$</td>
<td>$\mathcal{K}, p+1 \models_{\phi_1}$</td>
</tr>
<tr>
<td>$\neg \phi_1$</td>
<td>$p &lt; n$ implies $\mathcal{K}, p+1 \models_{\phi_1}$</td>
</tr>
<tr>
<td>$\mathcal{K}, p$</td>
<td>$p &gt; 0$ and $\mathcal{K}, p-1 \models_{\phi_1}$</td>
</tr>
<tr>
<td>$\phi_1$</td>
<td>$p &gt; 0$ implies $\mathcal{K}, p-1 \models_{\phi_1}$</td>
</tr>
</tbody>
</table>

9.2. Inconsistency-tolerant TCQ answering without rigid predicates

In this section $\mathcal{K}$ is a $\mathcal{L}$-TKB and $\phi$ a TCQ without negation.

When $N_{\text{RC}} = N_{\text{IR}} = \emptyset$, an important consequence of Proposition 3.5 is that the repairs of $\mathcal{K}$ are all possible sequences $(\mathcal{A}_i^k)_{0 \leq i \leq n}$, where $\mathcal{A}_i$ is a repair of $\langle T, \mathcal{A}_i \rangle$, so the intersection of the repairs of $\mathcal{K}$ is $(\mathcal{A}_i^k)_{0 \leq i \leq n}$, where $\mathcal{A}_i^k$ is the intersection of the repairs of $\langle T, \mathcal{A}_i \rangle$. This allows us to show that the entailment (resp. IAR entailment) of a BTCQ without negation from a consistent (resp. possibly inconsistent) TKB in a DL $\mathcal{L}$ that has the canonical model property for BCQ entailment can be equivalently defined w.r.t. the entailment (resp. IAR entailment) of the BCQs it contains as follows.

Proposition 9.12. If $\mathcal{L}$ has the canonical model property for BCQ entailment and $N_{\text{RC}} = N_{\text{IR}} = \emptyset$, then the entailments shown in Table 3 hold for $S = \text{classical}$ when $\mathcal{K}$ is consistent, and for $S = \text{IAR}$.

Proof. For the consistent case, all relations in Table 3 but the first are straightforwardly obtained by applying the definitions of BTCQ satisfaction of Table 1 in Section 2 to the canonical model of $\mathcal{K}$. Moreover, by Proposition 3.6, if $p < n$, then $\mathcal{K}, p \models_{\exists \vec{y} \psi(\vec{y})}$ iff $\langle T, \mathcal{A}_p \rangle \models_{\exists \vec{y} \psi(\vec{y})}$. Finally, $\mathcal{K}, p \models_{\exists \vec{y} \psi(\vec{y})}$ if $p > n$, because there exists a model of $\mathcal{K}$ whose $p^{th}$ component interprets every predicate as the empty set.

For IAR semantics, let $(\mathcal{A}_i^k)_{0 \leq i \leq n}$ denote the intersection of the repairs of $\mathcal{K}$ and $\mathcal{A}_i$ denote the intersection of the repairs of $\langle T, \mathcal{A}_i \rangle$.

- $\mathcal{K}, p \models_{\text{IAR}} \exists \vec{y} \psi(\vec{y})$ iff $\langle T, (\mathcal{A}_i^k)_{0 \leq i \leq n} \rangle, p \models_{\exists \vec{y} \psi(\vec{y})}$, i.e., iff $p \leq n$ and $\langle T, \mathcal{A}_p \rangle \models_{\exists \vec{y} \psi(\vec{y})}$ because $(\mathcal{A}_i^k)_{0 \leq i \leq n}$ is consistent. Since the repairs of $\mathcal{K}$ are the sequences of the repairs of $\langle T, \mathcal{A}_i \rangle$, $\mathcal{A}_i^k = \mathcal{A}_i$; so $\mathcal{K}, p \models_{\text{IAR}} \exists \vec{y} \psi(\vec{y})$ iff $p \leq n$ and $\langle T, \mathcal{A}_p \rangle \models_{\text{IAR}} \exists \vec{y} \psi(\vec{y})$.

- $\mathcal{K}, p \models_{\text{IAR}} \phi_1 \land \phi_2$ iff $\langle T, (\mathcal{A}_i^k)_{0 \leq i \leq n} \rangle, p \models_{\phi_1}$ and $\langle T, (\mathcal{A}_i^k)_{0 \leq i \leq n} \rangle, p \models_{\phi_2}$, i.e., iff $\langle T, (\mathcal{A}_i^k)_{0 \leq i \leq n} \rangle, p \models_{\phi_1}$ and $\langle T, (\mathcal{A}_i^k)_{0 \leq i \leq n} \rangle, p \models_{\phi_2}$ because $(\mathcal{A}_i^k)_{0 \leq i \leq n}$ is consistent. It follows that $\mathcal{K}, p \models_{\text{IAR}} \phi_1 \land \phi_2$ if $\mathcal{K}, p \models_{\text{IAR}} \phi_1$ and $\mathcal{K}, p \models_{\text{IAR}} \phi_2$.

- We show all remaining relations in the same way, applying the definition of IAR semantics and using the fact that $(\mathcal{A}_i^k)_{0 \leq i \leq n}$ is consistent.

This is an important result, since it implies that answering temporal CQs under IAR semantics can be done with the algorithms developed for the consistent case (see [18, 43] for algorithms for DL-Lite_R without unbounded future operators) by replacing classical CQ answering by IAR CQ answering (see [30, 37, 38] for algorithms for DL-Lite_R). The following example shows that this is unfortunately not true for brave or AR semantics.

Example 9.13. Consider the TKB $\mathcal{K} = \langle T, (\mathcal{A}_i)_{0 \leq i \leq n} \rangle$ and the TCQ $\phi$ defined as follows.

$T = \{ T \subseteq \neg F \}$

$\mathcal{A}_i = \{ T(a), F(a) \}$ for $0 \leq i \leq n$

$\phi = \Box \neg (T(a) \land \bullet F(a))$

Now, $\mathcal{K}, k \models_{\text{brave}} T(a) \land \bullet F(a)$ for every $0 \leq k \leq n$, but $\mathcal{K}, n \not\models_{\text{brave}} \phi$. This is because the same repair cannot entail $T(a) \land \bullet F(a)$ both at time point $k$ and $k+1$, since it would contain both $(T(a), k)$ and $(F(a), k)$ which is not possible. For AR semantics, consider $\phi = T(a) \lor F(a)$ over the TKB $\mathcal{K}$; while $\phi$ holds under AR semantics at each time point, neither $T(a)$ nor $F(a)$ does.

However, if the operators allowed in the TCQ are restricted to $\land, \lor, \bullet, \circ, \neg, \Box, \Rightarrow, \Rightarrow^p$, and $\neg$, then AR TCQ answering can be done with the algorithms developed for the consistent case by simply replacing classical CQ answering by AR CQ answering (see [38] for algorithms for DL-Lite_R). Indeed, for these operators, the relations of Proposition 9.12 hold for $S = \text{AR}$.  

28  

C. Bourgaux et al. / Ontology-mediated query answering over temporal and inconsistent data
We can construct similar counter-examples for general TCQs the "if" direction of Proposition 9.12 is consistent. Since the repairs of K are the sequences of the repairs of \( \langle T, A_0 \rangle \), this is the case iff \( p \leq n \) and for every repair \( A'_p \) of \( \langle T, A_0 \rangle \), \( \langle T, A_0 \rangle \models \exists \vec{y} \psi(\vec{y}) \), i.e., iff \( p \leq n \) and \( \langle T, A_p \rangle \models \exists \vec{y} \psi(\vec{y}) \).

We show all remaining relations in the same way, applying the definition of AR semantics and using the fact that TKB repairs are consistent.

The following counter-examples show that this is not the case for the other operators: \( \lor, \land, \land^\ast, \land^\ast_k \) for every \( k \neq 0 \), and \( \land \).

\[ K, p \models \land^\ast_k \phi \land \phi_2 \] for every \( \phi \) of \( \text{TBox} \) such that the query can be entailed at the same time point appears in an NI entailed by the \( \text{TBox} \).

\[ K, 0 \models \land^\ast_k \phi \quad \text{or} \quad K, 0 \not\models \land^\ast_k \phi \] but for every \( k \) (resp. such that \( 0 \leq k \leq 2 \)), \( K, k \not\models \land^\ast_k \phi_1 \):

\[ \mathcal{T} = \{ A \sqsubseteq \neg B \} \quad A_0 = \{ A(a), B(a) \} \]
\[ \phi_1 = A(a) \quad \phi_2 = B(a) \]

\[ K, 0 \models \land^\ast_k \phi_1 \land \phi_2 \] but for every \( k \) (resp. such that \( 0 \leq k \leq 2 \)), either \( K, k \not\models \land^\ast_k \phi_2 \) or there exists \( j \), such that \( 0 \leq j < k \) and \( K, j \not\models \land^\ast_k \phi_1 \):

\[ \mathcal{T} = \{ A \sqsubseteq \neg B \} \quad A_0 = \{ A(a) \} \]
\[ A_1 = \{ A(a), B(a) \} \quad A_2 = \{ B(a) \} \]
\[ \phi_1 = A(a) \land \land^\ast_k B(a) \quad \phi_2 = B(a) \]

We can construct similar counter-examples for \( \land^\ast \) and \( \land \).

Interestingly, contrary to the brave semantics, even for general TCQs the "if" direction of Proposition 9.12 is true.

\[ - K, p \models \land^\ast_k \phi_1 \lor \land^\ast_k \phi_2 \] if \( K, p \models \land^\ast_k \phi_1 \) or \( K, p \models \land^\ast_k \phi_2 \), then \( K, p \models \land^\ast_k \phi_1 \lor \land^\ast_k \phi_2 \).

\[ - K, p \models \land^\ast_k \phi_1 \lor \land^\ast_k \phi_2 \] if \( K, p \models \land^\ast_k \phi_1 \) or \( K, p \models \land^\ast_k \phi_2 \), then \( K, p \models \land^\ast_k \phi_1 \lor \land^\ast_k \phi_2 \).

It follows that even for unrestricted TCQs, combining algorithms for TCQ answering with algorithms for AR query answering will provide a sound approximation of AR answers.

For brave semantics, it would be useful to characterize the queries for which this method would be correct. Indeed, for many pairs of a TBox and a query, the minimal subsets of the TKB such that the query can be mapped into them cannot be inconsistent. For instance, for DL-Lite\(_R\) TKBs, this is the case if no pair of predicates that may be involved at the same time point appears in an NI entailed by the TBox. Consider for instance \( \mathcal{T} = \{ A \sqsubseteq \neg C, B \sqsubseteq \neg C \} \) and \( \phi = \exists x A(x) \land (\exists x B(x) \land \land^\ast (\exists x C(x))) \). For \( \phi \) to be entailed at time point \( p \), \( \exists x A(x) \) should hold at \( p \), \( \exists x B(x) \) at time point \( i \geq p \) and \( \exists x C(x) \) at \( i + 1 > p \), so there cannot be a conflict between the \( C \) and the \( A \) or \( B \) timed assertions used to satisfy the different CQs.

10. Conclusions and future work

For stream reasoning, handling the temporal dimension of the collected data and being resilient against errors in the data are expedient requirements. In the presence of erroneous data, handling inconsistencies is indispensable for logic-based approaches to stream reasoning. In this paper, we have lifted the standard inconsistency-tolerant semantics AR, IAR and brave to a temporal query answering setting that has been widely studied in the literature—namely, where the data is associated with time points and only the query language admits the use of temporal operators.
from LTL. We have presented complexity results and techniques to combine temporal with inconsistency-tolerant query answering over lightweight DL temporal knowledge bases suited for ontology-mediated situation recognition.

Our main contribution is a complexity analysis of the three semantics, focusing on the DLs $\mathcal{E}L_\perp$ and DL-Lite$_R$, where we distinguished the cases based on whether rigid concept or role names occur in the TKB, and on whether the query contains negation. We provided general algorithms that allow us to derive the complexity of temporal inconsistency-tolerant query answering from the complexities of consistency checking and classical entailment of temporal conjunctive queries. We furthermore completed the complexity picture for the classical semantics for TCQs without negations. Indeed, for the case where the query language does not provide negation, we devised a general approach to assess the complexity by the use of the canonical model property for B(T)CQ answering and thus not only limited to a particular DL. This approach allows us to derive the complexity of temporal query answering from the complexity of conjunctive query entailment for DLs that have this canonical model property.

Encouragingly, our analysis shows that either with or without negation in the query, in most cases, inconsistency-tolerant reasoning and temporal query answering can be combined without increasing the computational complexity. Furthermore, our results show that disallowing negation in the query language results in a drop in the combined complexity of TCQ answering, and, in the case of $\mathcal{E}L_\perp$ with rigid predicates, even in the data complexity. This raises hope that ontology-based stream reasoning applications which are resilient against noise in the data can be feasibly implemented and used in practice.

As a second major contribution, we investigated two techniques useful for developing practical algorithms for inconsistency-tolerant temporal query answering. We first showed that in DL-Lite$_R$, under the classical semantics and for queries without negation and unbounded temporal operators, rigid predicates can be handled by adding a set of assertions of polynomial size to each ABox from the TKB. However, our approach, which reduces TCQ answering with rigid predicates to TCQ answering without rigid predicates, works only for BTCQ entailment under the classical semantics.

We then showed that in the case without rigid predicates and for queries without negation, TCQ answering under IAR semantics can be implemented by combining algorithms developed for TCQ answering under the classical semantics with algorithms for CQ answering under IAR semantics over atemporal KBs. Moreover, we showed that when disallowing some of the operators in the queries, the same method can be used for AR semantics, while for full TCQs without negation, it provides for a sound approximation of the AR answers. Unfortunately, this is not the case for brave semantics, which is relevant for practical applications such as recognizing highly critical situations. Thus it would be useful to characterize the queries and TBoxes for which this method is correct. Now, fully fledged practical algorithms still remain to be found for inconsistency-tolerant temporal query answering with rigid predicates.

Another interesting open question is how the complexity changes if we admit not only the qualitative temporal operators from LTL, but also metric temporal operators as found in MTL or LTL$^{\text{bin}}$ [20, 51, 52]. In LTL$^{\text{bin}}$, temporal operators can be annotated by time intervals over which they are evaluated. In a query language with these operators, it would for example be possible to directly express that a query was entailed at some point during the last 5–10 time units, using the expression $\bigodot_{[5,10]} \phi$. It is well-known that LTL$^{\text{bin}}$ formulas can be exponentially encoded in LTL (assuming a binary encoding of the intervals) [52], which would also apply to our setting. This immediately gives complexity upper bounds due to our results, tight in the case of data complexity, but with an exponential increase in the case of combined complexity. This blow-up might be unavoidable, since propositional LTL$^{\text{bin}}$ is EXPSPACE-complete. In fact, combinations of LTL$^{\text{bin}}$ operators with description logics often lead to an exponential increase in complexity [20, 53], and it would be interesting to investigate whether this also happens for the settings considered in this paper. Regarding MTL, an initial investigation of query answering with a Horn-fragment of MTL has been performed in [19].

Acknowledgements

This work has been supported by the DFG in CRC 912 (HAEC), the DAAD, and the grant ANR-16-CE23-0007-01 (“DICOS”). The authors would like to thank Stefan Borgwardt for helpful discussions and the anonymous reviewers for helpful comments.
References


We start by defining the notions of conflicts and causes that will be used in some proofs. A conflict for a KB $K = \langle T, A \rangle$ is a minimal $T$-inconsistent subset of $A$. A cause for a BCQ $q$ w.r.t. $K$ is a minimal $T$-consistent subset $C \subseteq A$ such that $(T, C) \models q$. The following definitions extend these notions to the temporal setting.

**Definition A.1** (Conflicts of a TKB). A conflict of a TKB $K = \langle T, (A_i)_{0 \leq i < n} \rangle$ is a sequence of ABoxes $(A_i)_{0 \leq i < n}$ such that $\{ (a, i) | \alpha \in A_i, 0 \leq i < n \}$ is a minimal $T$-inconsistent subset of $\{ (a, i) | \alpha \in A_i, 0 \leq i < n \}$.

The conflicts of a DL-Lite$_R$-TKB are at most binary, i.e., contain at most two timed assertions (Fact 6.11).

**Definition A.2** (Causes for a BTCQ in a TKB). A cause for a BTCQ $\phi$ at point $p$ in $K = \langle T, (A_i)_{0 \leq i < n} \rangle$ is a sequence of ABoxes $(C_i)_{0 \leq i < n}$ such that $\{ (a, i) | \alpha \in C_i, 0 \leq i \leq n \}$ is a minimal $T$-consistent subset of $\{ (a, i) | \alpha \in A_i, 0 \leq i \leq n \}$ such that $(T, (C_i)_{0 \leq i < n}, p) \models \phi$.

Note that a KB (resp. TKB) is consistent iff it has no conflict, and that a BCQ (resp. BTCQ) is entailed from a KB (resp. a TKB) $K$ under brave semantics iff it has some cause in $K$, since such a cause can be extended to a repair that entails the query.

### A.1. Proofs of complexity results

**Hardness of data complexity for brave semantics in $\mathcal{E}_{L\downarrow}$**

**Proposition 6.9.** BTCQ entailment from an $\mathcal{E}_{L\downarrow}$-TKB with $\text{Neg} \neq \emptyset$ is

- coNP-hard w.r.t. data complexity under AR and IAR semantics,
- $\Sigma_2^p$-hard w.r.t. data complexity under brave semantics.

**Proof.** The lower bounds for AR and IAR semantics follow from the atemporal case, so that we only have to provide a lower bound for brave semantics.

We show that the complement of brave TCQ entailment is $\Pi_2^p$-hard by reduction from $\text{QBF}_{2,y}$. Let $\phi = \forall x_1 \ldots x_m \exists y_1 \ldots y_r. \varphi$ be a $\text{QBF}_{2,y}$-formula, where $\varphi = \bigwedge_{i=0}^{m} \ell_i^0 \lor \ell_i^1 \lor \ell_i^2$: a 3-CNF formula over the propositional variables $\{ x_1, \ldots, x_m, y_1, \ldots, y_r \}$. Based on $\Phi$, we define the TKB $K = \langle T, (A_i)_{0 \leq i < 3k+2} \rangle$ and the TCQ $\phi$ as follows, where $N_{\text{TCQ}} = \{ T \}$.

$$T = \{ \exists \text{Pos}. T \sqsubseteq \text{Sat}, \exists \text{Neg}. \text{F} \sqsubseteq \text{Sat}, \exists \text{FromPos}. \text{Sat} \sqsubseteq T, \exists \text{FromNeg}. \text{Sat} \sqsubseteq \text{F}, \exists \text{FromY}. \text{Sat} \sqsubseteq T, T \sqcap \text{F} \sqsubseteq \bot, T \sqcap \exists \text{ValY}. T \sqsubseteq \bot \}$$

$$\phi = \neg \Box^0 \text{NotFirst}(c) \lor \text{Sat}(c) \lor \circ \text{Sat}(c) \lor \bigcirc \text{Sat}(c)$$

For each clause $\ell_i^0 \lor \ell_i^1 \lor \ell_i^2$, $i \in [0, m]$, we define the following three ABoxes $A_{3i+k}$ ($0 \leq k \leq 2$):

$$A_{3i+0} = \mathcal{B} \cup B_{3i+0}$$
$$A_{3i+k} = \mathcal{B} \cup B_{3i+k} \cup \{ \text{NotFirst}(c) \}, 1 \leq k \leq 2,$$

where

$$\mathcal{B} = \{ T(x_j), F(x_j) | 1 \leq j \leq m \} \cup \{ \text{ValY}(y_j, \neg y_j) | 1 \leq j \leq r \}$$
$$B_{3i+k} = \{ \text{Pos}(c, x_j), \text{FromPos}(x_j, c) \}$$ if $\ell_i^k = x_j$.
Claim. \( \Phi \) is valid iff \( \mathcal{K} \vdash_{\text{brave}} \phi \).

Proof of claim. The repairs of \( \mathcal{K} \) correspond to the valuations of the \( x_j \). Indeed, since \( \mathcal{T} \) is rigid and disjoint from \( F \), each pair of timed assertions \( \{(T(x_j), i), (N(x_j), i')\} \) is inconsistent, so every \( x_j \) is such that a repair of \( \mathcal{K} \) contains either \( (T(x_j), i) \) for every \( i \), or \( (F(x_j), i') \) for every \( i \). For each repair \( \mathcal{A}' = (\mathcal{A}'_i)_{0 \leq i \leq 3l^2 + 2} \) of \( \mathcal{K} \), we denote by \( v_{\mathcal{A}'}^j \) the valuation of the \( x_j \) defined by \( v_{\mathcal{A}'}^j(x_j) = \text{true} \) if \( (T(x_j), i) \in \mathcal{A}'_i \). Correspondingly, for every valuation \( v_T \) of \( x_j \), we denote by \( \mathcal{A}^v = (\mathcal{A}^v_i)_{0 \leq i \leq 3l^2 + 2} \) the repair of \( \mathcal{K} \) defined by \( (T(x_j), i) \in \mathcal{A}^v_i \) for every \( i \) if \( v_T(x_j) = \text{true} \).

Assume that \( \Phi \) is valid and let \( \mathcal{A}' = (\mathcal{A}'_i)_{0 \leq i \leq 3l^2 + 2} \) be a repair of \( \mathcal{K} \). Since \( \Phi \) is valid, there then exists a valuation \( v_T \) of the \( x_j \) such that \( v_T(x_j) = \text{true} \) is satisfied by \( v_T \). Let \( \mathcal{J} = (\mathcal{L}_i)_{i \geq 0} \) be a model of \( \langle \mathcal{T}, (\mathcal{A}'_i)_{0 \leq i \leq 3l^2 + 2} \rangle \) that respects rigid predicates and such that for every \( i \):

- \( y_{iJ}^T \in \mathcal{T}_i \) iff \( v_T(y_{ij}) = \text{true} \), and
- \( -y_{iJ}^T \in \mathcal{T}_i \) iff \( v_T(y_{ij}) = \text{false} \), and
- if there exists some \( d \) such that \( (d^T, c^T) \in \text{FromY}^T \) and \( d^T \in \mathcal{T}_i \), then \( c^T \in \text{SatY}^T \).

One can verify that such a model always exists. First, because the role \( \text{ValY} \) connects only individuals of the type \( y_j \) and \( y_{ij} \), and we only assign \( T \) to one of them, these additional constraints respect the TBox axioms \( T \sqcap \text{ValY} \sqsubseteq T \sqcup \text{ValY} \). Second, the assignment of \( c \) to \( \mathcal{K} \) respects \( \exists \text{FromY} \sqsubseteq \text{Sat} \sqsubseteq \mathcal{T} \) by construction.

It is easy to see that if \( \mathcal{J} = (\mathcal{L}_i)_{i \geq 0} \sqsubseteq \boxplus (\text{NotFirst}(c) \lor \text{Sat}(c) \lor \Box \text{Sat}(c) \lor \Box \text{Sat}(c)) \), indeed, at each time point \( p \in [0, 3l^2 + 2] \), either \( \text{NotFirst}(c) \) is true, or \( p = 3i \), in which case we show that \( \mathcal{J} \) is true at time point \( 3i \) \( k \), where \( l^2 \) is the first literal of the clause \( \ell^2_i \lor \ell^1_i \lor \ell^3_i \) satisfied by \( v_T^i \lor v_T \).

- If \( \ell^1_i = x_j \), then \( v_T^i(x_j) = \text{true} \). Thus, by construction, \( T(x_j) \in \mathcal{A}'_{l^2 + 1} \), and therefore \( \mathcal{J}, 3i + k \models T(x_j) \). Moreover, since \( \mathcal{J}, 3i + k \models \text{Pos}(c, x_j) \), also \( \mathcal{J}, 3i + k \models \text{Sat}(c) \), because \( \mathcal{J} \) is a model of \( \mathcal{T} \).

- If \( \ell^2_i = -x_j \), then \( v_T^i(x_j) = \text{false} \). Thus, by construction, \( F(x_j) \in \mathcal{A}'_{l^2 + 1} \), and therefore \( \mathcal{J}, 3i + k \models F(x_j) \). Moreover, since \( \mathcal{J}, 3i + k \models \neg \text{Neg}(c, x_j) \), we obtain \( \mathcal{J}, 3i + k \models \text{Sat}(c) \) because \( \mathcal{J} \) is a model of \( \mathcal{T} \).

- If \( \ell^3_i = y_j \), then \( v_T^i(y_j) = \text{true} \). Thus, \( \mathcal{J}, 3i + k \vdash T(y_j) \), and since \( \mathcal{J}, 3i + k \models \text{FromY}(y_j, c) \) by construction of \( \mathcal{J} \), it follows that \( \mathcal{J}, 3i + k \models \text{Sat}(c) \).

- If \( \ell^4_i = -y_j \), then \( v_T^i(y_j) = \text{false} \). Thus, \( \mathcal{J}, 3i + k \vdash T(-y_j) \), and since \( \mathcal{J}, 3i + k \models \text{FromY}(-y_j, c) \) by construction of \( \mathcal{J} \), it follows that \( \mathcal{J}, 3i + k \models \text{Sat}(c) \).

It follows that \( \mathcal{J}, 0 \not\models \phi \), so \( \langle \mathcal{T}, (\mathcal{A}'_i)_{0 \leq i \leq 3l^2 + 2} \rangle, 0 \not\models \phi \). Hence, \( \mathcal{K}, 0 \not\models \phi \).
It follows that \( v_X \cup v_Y \) satisfies every clause \( \ell_1 \vee \ell_2 \). We have thus shown that \( \varphi[x_i \leftarrow v_X(x_i), y_j \leftarrow v_Y(y_j)] \) evaluates to true, and that \( \Phi \) is valid.

Canonical model property.

**Theorem 7.4.** If \( L \) has the canonical model property for BCQ entailment, then \( L \) also has the canonical model property for entailment of BTCQs without negation.

**Proof.** Let \( \overline{I}_K \) be the canonical model of \( K \) and \( \overline{I}_K = (\overline{I}_i)_{i \geq 0} = \text{temp}(\overline{I}_K) \). We show that \( J_K \) is a canonical model of \( K \) for BTCQs without negation, that is, for every BTCQ \( \phi \) that does not contain any negation, \( K, p \models \phi \iff J_K, p \models \phi \).

Since by Lemma 6.2, \( J_K \) is a model of \( K \) that respects rigid predicates, if \( K, p \models \phi \), then also \( J_K, p \models \phi \). For the other direction, we show by induction on the structure of \( \phi \) that \( J_K, p \models \phi \) implies \( K, p \models \phi \).

If \( \phi = q \) is a BCQ, by Lemma 7.1, \( K, p \models q \iff K \models \tilde{q}_p \), which is exactly the case iff \( \overline{I}_K \models \tilde{q}_p \).

By construction of \( J_K \), it follows that \( K, p \models q \iff J_K, p \models q \).

Assume that for two BTCQs \( \phi_1, \phi_2 \) and any \( p \geq 0 \), \( J_K, p \models \phi_1 \) implies \( K, p \models \phi_1 \).

- If \( J_K, p \models \phi_1 \land \phi_2 \), then \( J_K, p \models \phi_1 \) and \( J_K, p \models \phi_2 \). Hence by assumption, \( K, p \models \phi_1 \) and \( K, p \models \phi_2 \), and thus \( K, p \models \phi_1 \land \phi_2 \).
- If \( J_K, p \models \phi_1 \lor \phi_2 \), then \( J_K, p \models \phi_1 \) or \( J_K, p \models \phi_2 \). Hence, by assumption, \( K, p \models \phi_1 \) or \( K, p \models \phi_2 \), and thus \( K, p \models \phi_1 \lor \phi_2 \).
- If \( J_K, p \models \Box \phi_1 \), then \( J_K, p + 1 \models \phi_1 \). Hence, by assumption, \( K, p + 1 \models \phi_1 \), and thus \( K, p \models \Box \phi_1 \).
- We can show similarly that \( J_K, p \models \Box \phi_1 \) implies \( K, p \models \Box \phi_1 \), that \( J_K, p \models \Box \phi_1 \) implies \( K, p \models \Box \phi_1 \), and that \( K, p \models \Box \phi_1 \).
- If \( J_K, p \models \Box \phi_1 \), then for every \( k \geq p \), \( J_K, k \models \phi_1 \). Hence, by assumption, for every \( k \geq p \), \( K, k \models \phi_1 \), and thus \( K, p \models \Box \phi_1 \).

We conclude that for every BTCQ \( \phi \) without negation, \( J_K, p \models \phi \) implies \( K, p \models \phi \), and that \( J_K \) is a canonical model of \( K \) for BTCQs without negation. It follows that if a DL \( L \) has the canonical model property for BCQ entailment, then \( L \) also has the canonical model property for entailment of BTCQs without negation.

**Proposition 7.6.** Assume \( L \) has the canonical model property for BCQ entailment. Then, for every \( L \)-TKBs \( K, K' \), if \( K, K' \) coincide for BCQ entailment, then \( K \) and \( K' \) coincide for entailment of BTCQs without negation. I.e., if for every time point \( p \) and BCQ \( q \), \( K, p \models q \iff K', p \models q \) then for every time point \( p \) and BCQ \( \phi \) without negation, \( K, p \models \phi \iff K', p \models \phi \).

**Proof.** Assume that for every time point \( p \) and BCQ \( q \), \( K, p \models q \iff K', p \models q \). We can then show by induction on the structure of \( \phi \) that \( K, p \models \phi \iff K', p \models \phi \). Assume that for two BTCQs \( \phi_1, \phi_2, K, p \models \phi_1 \iff K', p \models \phi_1 \), which by assumption is the case iff \( K, p \models \phi_1 \) and \( K', p \models \phi_1 \). Then, since by Theorem 7.4, \( L \) has the canonical model property for entailment of BCQ without negation, by applying the definitions of BTCQ satisfaction in Table 1 to the canonical models of \( K \) and \( K' \), we obtain the following about formulas composed of \( \phi_1 \) and \( \phi_2 \).

- \( K, p \models \phi_1 \land \phi_2 \iff K, p \models \phi_1 \) and \( K, p \models \phi_2 \), which by assumption is the case iff \( K', p \models \phi_1 \) and \( K', p \models \phi_2 \), i.e., iff \( K, p \models \phi_1 \land \phi_2 \).
- \( K, p \models \phi_1 \lor \phi_2 \iff K, p \models \phi_1 \) or \( K, p \models \phi_2 \), which by assumption is the case iff \( K', p \models \phi_1 \) or \( K', p \models \phi_2 \), i.e., iff \( K', p \models \phi_1 \lor \phi_2 \).
- \( K, p \models \Box \phi_1 \iff K, p + 1 \models \phi_1 \), which by assumption is the case iff \( K', p + 1 \models \phi_1 \), i.e., iff \( K', p \models \Box \phi_1 \).
- We show in the same way that \( K, p \models \Box \phi_1 \iff K', p \models \Box \phi_1 \), that \( K, p \models \Box \phi_1 \iff K', p \models \Box \phi_1 \), that \( K, p \models \Box \phi_1 \iff K', p \models \Box \phi_1 \), and thus \( K, p \models \Box \phi_1 \).
- We show in the same way that \( K, p \models \Box \phi_1 \iff K', p \models \Box \phi_1 \), that \( K, p \models \Box \phi_1 \), and thus \( K, p \models \Box \phi_1 \).
We show in the same way that $K, p \models \square \phi_1$ iff $K', p \models \square \phi_1$, and that $K, p \models \square \neg \phi_1$ iff $K', p \models \square \neg \phi_1$.

We show in the same way that $K, p \models \neg \phi_1$ iff there exists $k, k \geq p, K, k \models \phi_1$, which by assumption is the case iff there exists $k, k \geq p, K, k \models \phi_1$, i.e., iff $K', p \models \phi_1$.

We show in the same way that $K, p \models \exists \phi_1$ iff $K', p \models \exists \phi_1$, and that $K, p \models \exists \neg \phi_1$ iff $K', p \models \exists \neg \phi_1$.

We show in the same way that $K, p \models \phi_1 U \phi_2$ iff $K', p \models \phi_1 U \phi_2$, and that $K, p \models \phi_1 S \phi_2$ iff $K', p \models \phi_1 S \phi_2$.

We conclude that for every BTCQ without negation $\phi$ and time point $p$, we have $K, p \models \phi$ iff $K', p \models \phi$. $\square$

Justification structures for brave entailment of BTCQs without negation in the case $N_{RC} = N_{RR} = 0$. We show that if $N_{RC} = N_{RR} = 0$ and $\phi$ is a BTCQ without negation, then $K, p \models brave \phi$ if there exists a correct brave-justification structure $J$ for $\phi$ in $K$ that justifies $\phi$ at time point $p$. We prove both directions in separate lemmas.

**Lemma A.3.** If $N_{RC} = N_{RR} = 0$ and there is a correct brave-justification structure $J$ for $\phi$ in $K$ that justifies $\phi$ at time point $p$, then $K, p \models brave \phi$.

**Proof.** In order to show $K, p \models brave \phi$, we determine a cause $(C_i)_{0 \leq i \leq n}$ for $\phi$. To do this, we first select a sequence of tuples from $J$ as follows, where we make sure that we have exactly one tuple for every $i$, $0 \leq i \leq n$.

1. The tuple $(p, L_{\text{now}}^p, F_{\text{now}}^p, F_{\text{prev}}^p, F_{\text{next}}^p)$ is such that $\phi \in F_{\text{now}}^p$.

2. If the tuple $(i, L_{\text{now}}^i, F_{\text{now}}^i, F_{\text{prev}}^i, F_{\text{next}}^i)$ was selected and $0 < i < p$, then select a tuple of the form $(i - 1, L_{\text{now}}^{i - 1}, F_{\text{now}}^{i - 1}, F_{\text{prev}}^{i - 1}, F_{\text{next}}^{i - 1})$, where $F_{\text{next}}^{i - 1} = F_{\text{prev}}^p$ and $F_{\text{prev}}^{i - 1} = F_{\text{now}}^p$.

3. If the tuple $(i, L_{\text{now}}^i, F_{\text{now}}^i, F_{\text{prev}}^i, F_{\text{next}}^i)$ was selected and $p \leq i \leq n$, then select a tuple of the form $(i + 1, L_{\text{now}}^{i + 1}, F_{\text{now}}^{i + 1}, F_{\text{prev}}^{i + 1}, F_{\text{next}}^{i + 1})$, where $F_{\text{next}}^i = F_{\text{next}}^n$ and $F_{\text{prev}}^i = F_{\text{now}}^p$.

Because $J$ is correct and justifies $\phi$ at time point $p$, such a sequence can always be selected.

Based on this sequence, we construct a sequence of ABoxes $(C_i)_{0 \leq i \leq n}$. For this, we take for each of the tuples $(i, L_{\text{now}}^i, F_{\text{now}}^i, F_{\text{prev}}^i, F_{\text{next}}^i)$ a cause $C_i \subseteq A_i$ that entails $\bigwedge_{q \in L_{\text{now}}^i} q$. Such a cause exists because $(T, A_i) \models brave \bigwedge_{q \in L_{\text{now}}^i} q$ by Condition 1. Since each $C_i$ is consistent and we do not have rigid predicates, the TKB $(T, (C_i)_{0 \leq i \leq n})$ is consistent.

We prove $(T, (C_i)_{0 \leq i \leq n}), p \models \phi$ by showing that $(T, (C_i)_{0 \leq i \leq n}), p \models F_{\text{now}}^p$. To do this, we consider the sets $F_{\text{now}}^d = \{ \psi \mid \psi \in F_{\text{now}}, \text{degree}(\psi) \leq d \}$ of LTL formulas, where $\text{degree}(\psi)$ is the maximal number of nested LTL operators in $\psi$, and prove by induction on $d$ that for all $0 \leq i \leq n$ and for all $\psi \in F_{\text{now}}^d$, we have $(T, (C_i)_{0 \leq i \leq n}), i \models \psi$, i.e., $(T, (C_i)_{0 \leq i \leq n}), i \models F_{\text{now}}^d$.

For $d = 0$, $F_{\text{now}}^0$ contains only conjunctive queries of the form $\exists \psi \exists \psi(\phi)$. Since for every $\psi \in BCQ(\phi)$, $F_{\text{now}}^0 \models \psi$ implies $\psi \in L_{\text{now}}$ (Condition 4), we have that $F_{\text{now}}^0 \subseteq L_{\text{now}}$. From this, since $(T, C_i) \models \bigwedge_{q \in L_{\text{now}}} q$, it follows that $(T, (C_i)_{0 \leq i \leq n}), i \models F_{\text{now}}^0$.

Assume that for all $0 \leq i \leq n, (T, (C_i)_{0 \leq i \leq n}), i \models F_{\text{now}}^d$. Let $\psi \in F_{\text{now}}^{d + 1}$ for some $0 \leq i \leq n$. If $\psi \in F_{\text{now}}^{d + 1}$ then $(T, (C_i)_{0 \leq i \leq n}), i \models \psi$. Otherwise, $\text{degree}(\psi) = d + 1$ and we distinguish the cases based on the syntactical form of $\psi$.

- $\psi = \psi_1 \land \psi_2$, where $\text{degree}(\psi_1) \leq d, \text{degree}(\psi_2) \leq d$. Since $\psi \in F_{\text{now}}$, then $F_{\text{now}} \models \psi_1$ and $F_{\text{now}} \models \psi_2$, and by Condition 5, $\psi_1 \in F_{\text{now}}$ and $\psi_2 \in F_{\text{now}}$. It follows that $\psi_1 \in F_{\text{now}}$ and $\psi_2 \in F_{\text{now}}$, so $(T, (C_i)_{0 \leq i \leq n}), i \models \psi_1$ and $(T, (C_i)_{0 \leq i \leq n}), i \models \psi_2$. Hence, $(T, (C_i)_{0 \leq i \leq n}), i \models \psi_1 \land \psi_2$.

- $\psi = \psi_1 \lor \psi_2$, where $\text{degree}(\psi_1) \leq d, \text{degree}(\psi_2) \leq d$. Since $\psi \in F_{\text{now}}$, then by Condition 7 either $\psi_1 \in F_{\text{now}}$ or $\psi_2 \in F_{\text{now}}$. It follows that $\psi_1 \in F_{\text{now}}$ or $\psi_2 \in F_{\text{now}}$, so $(T, (C_i)_{0 \leq i \leq n}), i \models \psi_1$ or $(T, (C_i)_{0 \leq i \leq n}), i \models \psi_2$. Hence, $(T, (C_i)_{0 \leq i \leq n}), i \models \psi_1 \lor \psi_2$.

- $\psi = \psi_1 \land \neg \psi_2$, where $\text{degree}(\psi_1) \leq d$. By Condition 8, either $i < n$, or $i = n$ and $(\emptyset, (\emptyset)_{0 \leq i \leq n}), n + 1 \models \square \psi_1$. In the latter case, note that the canonical models of $(T, (C_i)_{0 \leq i \leq n})$ coincide after time point $n$ (atomic concepts and roles have empty interpretations), and since $\psi_1$ does not contain any past operators, $(T, (C_i)_{0 \leq i \leq n}), n + 1 \models \square \psi_1$ is a direct consequence of $(\emptyset, (\emptyset)_{0 \leq i \leq n}), n + 1 \models \square \psi_1$. Then $\psi_1$ is true at any time point $j > n$ and in particular, $(T, (C_i)_{0 \leq i \leq n}), n \models \psi_1$. In the former case, since there $\psi_1 \in F_{\text{now}}^d \models F_{\text{now}}^{d + 1}$, we have $\bigwedge_{q \in L_{\text{now}}} q \land \neg \exists \psi_1 \models \bigwedge_{q \in L_{\text{now}}} q \land \neg \psi_1 \models \bigwedge_{q \in L_{\text{now}}} q \land \neg \psi_1 \models \psi_1$, and by Condition 6, $\psi_1 \in F_{\text{now}}^{d + 1}$. 

Hence, $\psi_1 \in F_{i+1,d}^i$, $\langle T, (C_i)_{0 \leq i \leq n} \rangle, i + 1 \models \psi_1$, and $\langle T, (C_i)_{0 \leq i \leq n} \rangle, i \models \Box \psi_1$.

- $\psi = \Box^i \psi_1$, where degree$(\psi_1) \leq d$. This case is similar to $\Box \psi_1$.

- $\psi = \psi_i$, where degree$(\psi_i) \leq d$. If $i < n$, since $\Box^i \psi_i \in F_{i+1}^{i-1}$, we have that $\Lambda_{\psi_i \in F_{i+1}} \varphi \wedge \neg (\bigvee_{\chi \in F_{i+1}^{i-1}} \chi) <- (\bigvee_{\chi \in F_{i+1}^{i-1}} \chi) = \neg \Box^i \psi_i \models \psi_1$, so that by Condition 6, $\psi_i \in F_{i+1}^{i-1}$. Hence $\psi_1 \in F_{i+1}^{i-1}$ and $\langle T, (C_i)_{0 \leq i \leq n} \rangle, i + 1 \models \psi_1$, which implies $\langle T, (C_i)_{0 \leq i \leq n} \rangle, i \models \psi_i$. Otherwise, $i = n$, and $\langle T, (C_i)_{0 \leq i \leq n} \rangle, n \models \Box \psi_i$ by definition of $\psi_i$.

- $\psi = \Box \psi_i$, where degree$(\psi_i) \leq d$. This case is similar to $\Box \psi_i$.

- $\psi = \Box^i \psi_i$, where degree$(\psi_i) \leq d$. By Condition 7, $\langle \varnothing, (\varnothing)_{0 \leq i \leq n} \rangle, n + 1 \models \Box \psi_i$. We show that $\langle T, (C_i)_{0 \leq i \leq n} \rangle, i \models \Box \psi_i$ by descending induction on $i$.

For $i = n$, note that the canonical models of $\langle T, (C_i)_{0 \leq i \leq n} \rangle$ and $\langle \varnothing, (\varnothing)_{0 \leq i \leq n} \rangle$ coincide after time point $n$ (atomic concepts and roles have empty interpretations), and since $\psi_i$ does not contain any past interpretations, $\langle T, (C_i)_{0 \leq i \leq n} \rangle, n + 1 \models \Box \psi_i$ is a direct consequence of $\langle \varnothing, (\varnothing)_{0 \leq i \leq n} \rangle, n + 1 \models \Box \psi_i$. Then, $\langle T, (C_i)_{0 \leq i \leq n} \rangle, n \models \Box \psi_i$ iff $\langle T, (C_i)_{0 \leq i \leq n} \rangle, n \models \psi_i$, which in turn implies $\psi_i \in F_{i+1}$ by induction. This is the case by Condition 5.

For $i < n$, we assume by inductive hypothesis that $\Box \psi_i \in F_{i+1}$ implies $\langle T, (C_i)_{0 \leq i \leq n} \rangle, i + 1 \models \Box \psi_i$. Since $\Box \psi_i \in F_{i+1}$, we have that $\Lambda_{\psi_i \in F_{i+1}} \varphi \wedge \neg (\bigvee_{\chi \in F_{i+1}} \chi) <- (\bigvee_{\chi \in F_{i+1}} \chi) = \neg \Box \psi_i \models \psi_i$, so that by Condition 6, $\Box \psi_i \in F_{i+1}$, and by assumption $\langle T, (C_i)_{0 \leq i \leq n} \rangle, i + 1 \models \Box \psi_i$. Moreover, since $\Box \psi_i \in F_{i+1}$, then $F_{i+1} \models \psi_i$, and $\psi_i \in F_{i+1}$ by Condition 5. Hence, $\psi_i \in F_{i+1}$ and $\langle T, (C_i)_{0 \leq i \leq n} \rangle, i \models \psi_i$.

- $\psi = \Box^i \psi_i$ where degree$(\psi_i) \leq d$. We show $\langle T, (C_i)_{0 \leq i \leq n} \rangle, i \models \Box \psi_i$ by descending induction on $i$.

For $i = n$, if $\Box^i \psi_i \in F_{i+1}$, then $\psi_i \in F_{i+1}$ by Condition 8, and therefore $\psi_i \in F_{i+1}$ and $\langle T, (C_i)_{0 \leq i \leq n} \rangle, n \models \psi_i$. As a consequence, we obtain that $\langle T, (C_i)_{0 \leq i \leq n} \rangle, n \models \Box \psi_i$.

For $i < n$, we assume by inductive hypothesis that $\Box^i \psi_i \in F_{i+1}$ implies $\langle T, (C_i)_{0 \leq i \leq n} \rangle, i + 1 \models \Box^i \psi_i$. Then, since $\Box^i \psi_i \in F_{i+1}$, we have that $\Lambda_{\psi_i \in F_{i+1}} \varphi \wedge \neg (\bigvee_{\chi \in F_{i+1}} \chi) <- (\bigvee_{\chi \in F_{i+1}} \chi) = \neg \Box^i \psi_i \models \psi_i$. Therefore, by Condition 6, $\Box^i \psi_i \in F_{i+1}$, and by assumption $\langle T, (C_i)_{0 \leq i \leq n} \rangle, i + 1 \models \Box^i \psi_i$. Moreover, since $\Box^i \psi_i \in F_{i+1}$, then $F_{i+1} \models \psi_i$, and $\psi_i \in F_{i+1}$ by Condition 8, and therefore $\psi_i \in F_{i+1}$ and $\langle T, (C_i)_{0 \leq i \leq n} \rangle, i \models \psi_i$. It follows that $\langle T, (C_i)_{0 \leq i \leq n} \rangle, i \models \Box \psi_i$.
Lemma A.4. If $N_{\text{BG}} = N_{\text{BR}} = \emptyset$ and $K, p \models \text{brave } \phi$, then there exists a brave-justification structure for $\phi$ in $K$ that is correct and justifies $\phi$ at time point $p$.

Proof. Assume $K, p \models \text{brave } \phi$. Then there exists a TKB $K' = (T, (C_i)_{0 \leq i < n})$, such that $C_i \subseteq A$, and $K'$ is consistent and $K', p \models \phi$. Based on $K'$, we construct a brave-justification structure $J$ for $\phi$ in $K$ that justifies $\phi$ at time point $p$. The elements of the tuples $(i, L'_{\text{now}}, F'_{\text{now}}, F'_{\text{prev}}, F'_{\text{next}})$ are selected as follows.

1. $L'_{\text{now}}$ is the largest subset of $BCQ(\phi)$ such that $K', i \models \bigwedge_{q \in L'_{\text{now}}} q$.
2. $F'_{\text{now}}$ is the largest subset of $F(\phi)$ such that $K', i \models F'_{\text{now}}$.
3. $F'_{\text{prev}} = F'_{\text{now}} - 1$ for $i > 0$.
4. $F'_{\text{next}} = F'_{\text{now}} + 1$ for $i < n$, and $F'_{\text{next}} = \emptyset$.

We show that $J$ is correct and justifies $\phi$ at time point $p$. The latter case is easy: since $K', p \models \phi$, we have $\phi \in F'_{\text{now}}$, and therefore $\phi$ is justified by $J$ at time point $p$. It remains to show that $J$ is correct, i.e., that every tuple of $J$ satisfies all conditions in the definition of justified tuples.

Conditions 1, 2, 3 and 4 follow straightforwardly from the construction. Condition 5 is satisfied because if $\psi \in F(\phi)$ is such that $\psi \not\in F'_{\text{now}}$, then $K', i \not\models \psi$ and $F'_{\text{now}} \not\models \psi$.

For Condition 6, we show that for every $\psi \in F(\phi)$ and for every $0 \leq i \leq n$, if $\bigwedge_{q \in L'_{\text{now}}} q \land \bigwedge_{q \in F'_{\text{next}}} q \models \psi$, then $K', i \not\models \psi$, which in turn implies $\psi \not\in F'_{\text{now}}$. Since $K'$ entails every CQ in $F'_{\text{now}}$ at time point $i$, every TCQ in $F'_{\text{prev}}$ at time point $i - 1$, and every TCQ in $F'_{\text{next}}$ at time point $i + 1$, every TCQ that corresponds to a formula entailed by $L'_{\text{now}} \land \bigwedge_{q \in F'_{\text{next}}} q$ or $\bigwedge_{q \in F'_{\text{next}}} q$ is entailed from $K'$ at time point $i$. Hence, if $\bigwedge_{q \in L'_{\text{now}}} q \land \bigwedge_{q \in F'_{\text{next}}} q \models \psi$, then $K', i \not\models \psi$.

For Condition 7, we do a case analysis based on the structure of the elements in $F_{\text{now}}$, using the Proposition 9.12 and the fact that $N_{\text{BG}} = N_{\text{BR}} = \emptyset$.

– If $K', i \models \psi \land \psi'$, then $K', i \models \psi$ or $K', i \models \psi'$. Therefore, if $\psi \land \psi' \in F'_{\text{now}}$, either $\psi \in F'_{\text{now}}$, or $\psi' \in F'_{\text{now}}$.

– If $K', i \not\models \psi \land \psi'$, then $K', i \models \psi$ or $K', i \models \psi'$. Therefore, if $\psi \lor \psi' \in F'_{\text{now}}$, either $\psi \in F'_{\text{now}}$ or $\psi' \in F'_{\text{now}}$. ─
Proposition A.5. chase properties.

Condition 9 can be shown similarly as Condition 8.

We have shown that every tuple in \( J \) is justified, and consequently that \( J \) is correct and justifies \( \varphi \) at \( p \).

A.2. Proofs for Section 9

The following properties of \( chase_{rig}(K) \) will be useful for the proofs of Subsection 9.1.

Proposition A.5. \( chase_{rig}(K) \) satisfies the following properties.

(P1) \( x_{i+1}^j \in \Gamma_N \) implies \( P_1(a, x_{i+1}^j) \in chase_{rig}^K(A_{i+1}) \).

(P2) \( x_{i+1}^j \in \Gamma_N, l > 1 \) implies that \( chase_{rig}^K(A_i) \) contains \( P_l(x_{i+1}^{j-1}, x_{i+1}^{j-l}) \).

(P3) \( \chi^j_{i+1} \in \Gamma_N \) implies \( \exists \implies P_{i+1} \subseteq B \).

(P4) \( x_{i+1}^{j-1} \in \Gamma_N, l > 1 \) implies \( \exists \implies P_{i+1} \subseteq B \).

(P5) \( chase_{rig}^K(A_i) \models B(a), a \in N_{par}^K \) implies \( T \models B(a) \) or there exists \( B' := A \exists R \exists R' \) with \( A \in N_{par}, R \in N_{par} \) such that \( T \models B' \subseteq B \) and there exists \( j \) such that \( (T, A_i) \models B'(a) \).

(P6) \( chase_{rig}^K(A_i) \models B(x_{i+1}^{j-1}), i = i \) or there exists \( B' := A \exists R \exists R' \) with \( A \in N_{par}, R \in N_{par} \) such that \( T \models B' \subseteq B \) and \( chase_{rig}^K(A_i) \models B'(x_{i+1}^{j-1}) \).

(P7) \( P(a, b) \in chase_{rig}^K(A_i), a, b \in N_{par}^K \) implies \( (T, A_i) \models P(a, b) \) or there exists \( P' := R \exists R' \) with \( R \in N_{par} \) such that \( T \models P' \subseteq P \) and there exists \( j \) such that \( (T, A_i) \models P'(a, b) \).

(P8) \( P(a, x_{i+1}^j) \in chase_{rig}^K(A_i), a \in N_{par}^K, i = i \) implies \( T \models P_1 \subseteq P \) and \( (T, A_i) \models \exists x P_1(a, x) \) or there exists \( B := A \exists R \exists R' \) with \( A \in N_{par}, R \in N_{par} \) such that \( T \models B \subseteq P_1 \) and there exists \( j \) such that \( (T, A_i) \models B(a) \).

(P9) \( P(a, x_{i+1}^j) \in chase_{rig}^K(A_i), a \in N_{par}^K, i = i \) implies \( T \models P_1 \subseteq P \).

(P10) \( P(x, y) \in chase_{rig}^K(A_i), x, y \in \Gamma_N \) implies \( x = x_{i+1}^j \) or there exists \( y = y_{i+1}^j \) and \( T \models P_1 \subseteq P \) or \( x = x_{i+1}^j \) or there exists \( y = y_{i+1}^j \) and \( T \models P_1 \subseteq P \).

(P11) \( P(x_{i+1}^j, y_{i+1}^j) \in chase_{rig}^K(A_i), i = i \) implies \( T \models P_1 \subseteq P \) and \( P'(x_{i+1}^j, y_{i+1}^j) \in chase_{rig}^K(A_i+1) \).

(P12) \( P(x_{i+1}^j, y_{i+1}^j) \in chase_{rig}^K(A_i) \) implies \( \exists y', (T, A_i) \models \exists y' P_{i+1} \).

Proof. We refer to [54] for a detailed proof of these properties.

Lemma 9.3. If \( K \) is consistent, then \( J_K \) is a model of \( K \) that respects rigid predicates.

Proof. We first show that \( J_K \) is a model of \( K \), i.e., that for every \( i \in [0, n], I_i \models A_i \) and for every \( i > 0, I_i \models T_i \). It is easy to see that for every \( i \in [0, n], I_i \models A_i \), because \( A_i \subseteq chase_{rig}^K(A_i) \). We can show that \( I_i \) satisfies every PI in \( T_i \) with similar arguments as those used in [50]. We only consider the case where \( i \leq n + 1 \) explicitly. For the case where \( i > n + 1 \), we assume \( A_i \) to be replaced by \( A_{i+1} \) in what follows. If a PI \( \alpha \in \Pi_T \) is not satisfied, there is an assertion \( \beta \in chase_{rig}^K(A_i) \) such that \( \alpha \) is applicable to \( \beta \) in \( chase_{rig}^K(A_i) \). This is impossible given that every PI applicable to \( \beta \) in \( S_T \) at step \( j \) of the construction of the rigid chase becomes not applicable to \( \beta \) in \( S_T \) for some \( k \geq i \). Indeed, because each PI can only be applied once to a given assertion, there are only finitely many assertions before \( \beta \), and only finitely many PIs are applied to the assertions that precede \( \beta \). Finally, we show that because \( K \) is consistent, \( I_i \) satisfies every negative inclusion of \( T_i \). Indeed, if a negative inclusion would not be satisfied, this would imply the existence of a conflict \( B \) in \( chase_{rig}^K(A_i) \). If \( B = \{ a \} \), the timed assertion \( (a, j) \in (A_i)_{i \leq i \leq n} \) from which \( \alpha \) has been
derived by applying PIs from \( T_p \) is clearly inconsistent. Otherwise \( B = \{ \{ \alpha, \beta \} \} \) with \( \alpha \) derived from \( \langle \alpha', j \rangle \) and \( \beta \) derived from \( \langle \beta', k \rangle \). If \( j = k \), \( \{ \{ \alpha', j \}, \{ \beta', k \} \} \) is clearly inconsistent. If \( j \neq k \), since \( \alpha \) and \( \beta \) belong to \( \text{chase}^K_{\text{rig}}(A_i) \), if \( j \neq i \) (resp. \( k \neq i \)), there exists \( \alpha'' \in \text{chase}^K_{\text{rig}}(A_i) \) rigid such that \( \alpha \) derives from \( \alpha'' \), which derives from \( \alpha' \) (resp. \( \beta'' \in \text{chase}^K_{\text{rig}}(A_i) \) rigid such that \( \beta \) derives from \( \beta'' \), which derives from \( \beta' \)). Therefore, and because no sequence of interpretations that respects rigid predicates can be a model of \( \{ \{ \alpha', j \}, \{ \beta', k \} \} \) and \( T \), \( \{ \{ \alpha', j \}, \{ \beta', k \} \} \) is inconsistent.

Moreover, the model \( J_K \) respects rigid predicates, because if an assertion \( \beta \) of \( \text{chase}^K_{\text{rig}}(A_i) \) is rigid, either \( \beta \in A_i \) and by construction \( \beta \in S_i^0 = A_i \) for every \( k \), or \( \beta \) has been derived at some step \( j \) by applying some PI to an assertion of \( S_i^j \) and \( \beta \in S_i^{j+1} \) for every \( k \), so that in both cases \( \beta \in \text{chase}^K_{\text{rig}}(A_i) \) for every \( k \).

**Lemma 9.4.** If \( K \) is consistent, then for every \( BTOC \) without negation \( \phi \) such that \( N^K \subseteq N^K, K, p \models \phi \) iff \( J_K, p \models \phi \).

**Proof.** Since \( J_K \) respects rigid predicates, because if an assertion \( \beta \) of \( \text{chase}^K_{\text{rig}}(A_i) \) is rigid, either \( \beta \in A_i \) and by construction \( \beta \in S_i^0 = A_i \) for every \( k \), or \( \beta \) has been derived at some step \( j \) by applying some PI to an assertion of \( S_i^j \) and \( \beta \in S_i^{j+1} \) for every \( k \), so that in both cases \( \beta \in \text{chase}^K_{\text{rig}}(A_i) \) for every \( k \).

For every \( a \in N^K \) and concept \( A \), if \( a^{T_i} \in A^{T_i} \), i.e., \( A(a) \in \text{chase}^K_{\text{rig}}(A_p) \), then by (P5), either (i) \( (T, A_p) \models A(a) \), and since \( T_p \) is a model of \( (T, A_p) \), also \( h(a^{T_i}) = a^{T_{i-1}} \in A^{T_{i-1}} \), or (ii) there exists a concept \( B = C \upharpoonright R \subseteq C \subseteq N_{RB}, R \subseteq N_{RB}, R \subseteq N_{RB}, \text{ such that } T \models B \subseteq A, R \subseteq N_{RB}, R \subseteq N_{RB}, \text{ such that } T \models B \subseteq A \), and since \( T_p \) is a model of \( T \), it follows that \( h(a^{T_i}) \in A^{T_i} \).

For every \( x^{i-1}_{\alpha_1, \cdots, \alpha_l} \in N^K \), such that \( x^{i-1}_{\alpha_1, \cdots, \alpha_l} \in A^{T_i} \), i.e., \( A(a^{i-1}_{\alpha_1, \cdots, \alpha_l}) \in \text{chase}^K_{\text{rig}}(A_p) \), by (P6) we are in one of the following cases.

1. \( i = p \). By (P3), \( T_p \models A(a) \) and by construction of \( h \), \( h(x^{i-1}_{\alpha_1, \cdots, \alpha_l}) = y \) with \( h(x^{i-1}_{\alpha_1, \cdots, \alpha_l}) = y \) with \( h(x^{i-1}_{\alpha_1, \cdots, \alpha_l}) = y \) with \( h(x^{i-1}_{\alpha_1, \cdots, \alpha_l}) = y \) with \( h(x^{i-1}_{\alpha_1, \cdots, \alpha_l}) = y \). Hence if \( l = 1 \), then \( x^{i-1}_{\alpha_1, \cdots, \alpha_l} = a \). Since \( T_p \) is a model of \( T \), it follows that \( y \in A^{T_i} \).

2. There exists \( B := C \upharpoonright \exists R \subseteq C \subseteq N_{RB}, R \subseteq N_{RB}, \text{ such that } T \models B \subseteq A \), and since \( A^{T_i} \subseteq B \subseteq A \), it follows that \( y \in A^{T_i} \).

For every \( P \) such that \( (x^{T_i}, y^{T_i}) \in B^{T_i} \), by (P10) \( x = x^{i-1}_{\alpha_1, \cdots, \alpha_l} \), \( y = y^{i-1}_{\alpha_1, \cdots, \alpha_l} \), and \( T \models \exists P \subseteq A \), or \( x = x^{i-1}_{\alpha_1, \cdots, \alpha_l} \), \( y = y^{i-1}_{\alpha_1, \cdots, \alpha_l} \), and \( T \models \exists P \subseteq A \). We can assume w.l.o.g. that we are in the first case. Otherwise we
consider $(y^{x'}, x^{x'}) \in P'^{x'}$. If $i_{t+1} = p$, by definition of $h$, $(h(x^{x'}), h(y^{x'})) \in P'^{x'}_{t+1}$, and since $P'^{x'}$ is a model of $T$, $(h(x^{x'}), h(y^{x'})) \in P'^{x'}$. Otherwise, by (P11), there exists $P' = R|R^R$ with $R \in N_R$ such that $T \models P_{t+1} \subseteq P' \subseteq P$ and $P'(x, y) \in chase^K_{\mathcal{R}}(A_{t+1})$. With the same arguments as in the first case, we show that $(h(x^{x'}), h(y^{x'})) \in P'^{x'}_{t+1}$, and since $P'$ is rigid $(h(x^{x'}), h(y^{x'})) \in P'^{x'}$. Since $P'^{x'}$ is a model of $T$, it follows that $(h(x^{x'}), h(y^{x'})) \in P'^{x'}$. Finally, if $a \in N^K$ and $x \in \Gamma_N$, then $(a^{x'}, x^{x'}) \in P'^{x'}$ only if $x = x_{a^{x'}}$. If $i_t = p$, by definition of $h$, $(h(a^{x'}), h(x^{x'})) \in P'^{x'}_{t+1}$. Since by (P8), $T \models P_t \subseteq P' \subseteq P$ and $P'$ is a model of $T$, it follows that $(h(a^{x'}), h(x^{x'})) \in P'^{x'}_{t+1}$. If $i_t \neq p$, by (P9), there exists a rigid role $P'$ such that $T \models P_t \subseteq P' \subseteq P$, and since by definition of $h$, $(h(a^{x'}), h(x^{x'})) \in P'^{x'}_{t+1}$, then $(h(a^{x'}), h(x^{x'})) \in P'^{x'}$. Since $J$ respects rigid predicates, it follows that $(h(a^{x'}), h(x^{x'})) \in P'^{x'}$ and $(h(a^{x'}), h(x^{x'})) \in P'^{x'}$. We thus have shown that $J_{t+1}, p \models \exists y\psi(y)$ implies $J, p \models \exists y\psi(y)$.

Now for the inductive step, assume that for two BTCQs $\phi_1, \phi_2$ such that $N^\phi_1 \subseteq N^K$ and $N^\phi_2 \subseteq N^K$, we have that $J, p \models \phi_1$ and $J_{t+1}, p \models \phi_1$ and $J_{t+1}, p \models \phi_2$, and therefore by assumption, $J, p \models \phi_1$ and $J, p \models \phi_2$. Hence, $J, p \models \phi_1 \land \phi_2$.

- If $J_{t+1}, p \models \phi_1 \land \phi_2$, then $J_{t+1}, p \models \phi_1$ and $J_{t+1}, p \models \phi_2$, and therefore by assumption, $J, p \models \phi_1 \land \phi_2$.

- If $J_{t+1}, p \models \phi_1 \lor \phi_2$, then $J_{t+1}, p \models \phi_1$ or $J_{t+1}, p \models \phi_2$, and therefore by assumption, $J, p \models \phi_1$ or $J, p \models \phi_2$. Hence, $J, p \models \phi_1 \lor \phi_2$.

- If $J_{t+1}, p \models \phi \land \phi_1$, then $J_{t+1}, p \models \phi_1$, and therefore by assumption, $J, p \models \phi_1$. Hence, $J, p \models \phi$. In the same way as in the last case, we can show that $J, p \models \phi$ implies $J, p \models \phi$.

- If $J_{t+1}, p \models \phi_1 \land \phi$, then every $k \geq p$, $J_{t+1}, k \models \phi_1$, and therefore, for every $k \geq p$, $J_{t+1}, k \models \phi_1$. Hence, $J, p \models \phi$.

- If $J_{t+1}, p \models \phi_1 \lor \phi$, then there exists $k \geq p$, $J_{t+1}, k \models \phi_1$, and therefore by assumption, $J, k \models \phi_1$. Hence, $J, p \models \phi_1$.

- In the same way as in the last case, we can show that $J, p \models \phi_1 \lor \phi_1$ implies $J, p \models \phi_1 \lor \phi_1$.

- If $J_{t+1}, p \models \phi_1 \land \phi$, then there exists $k \geq p$, $J_{t+1}, k \models \phi_1$, and therefore by assumption, $J, k \models \phi_1$. Hence, $J, p \models \phi_1$.

- In the same way as in the last case, we can show that $J, p \models \phi_1 \lor \phi_1$ implies $J, p \models \phi_1 \lor \phi_1$.

- If $J_{t+1}, p \models \phi_1 \land \phi$, then there exists $k \geq p$, $J_{t+1}, k \models \phi_1$, and therefore by assumption, $J, k \models \phi_1$. Hence, $J, p \models \phi_1$. In the same way as in the last case, we can show that $J, p \models \phi_1 \lor \phi_1$ implies $J, p \models \phi_1 \lor \phi_1$.

We conclude by induction that for every BTCQ $\phi$ without negation such that $N^\phi \subseteq N^K$, $J_{t+1}, p \models \phi$ implies $J, p \models \phi$. It follows that $J_{t+1}, p \models \phi$ implies $K, p \models \phi$.

We have thus shown that for every BTCQ $\phi$ without negation such that $N^\phi \subseteq N^K$, we have $K, p \models \phi$ if $J_{t+1}, p \models \phi$.

Lemma 9.7. Let $q = \exists y\psi(y)$ be such that $N^\phi \subseteq N^K$. Then, for every $p \in [0, n]$, if $K_{R, p} \models q$ then $K, p \models q$.

Proof. Assume that $K_{R, p} \models \exists y\psi(y)$. By Proposition 3.6, since $N_{RC} = N_{RR} = \emptyset$, $(T, (A_p \cup R)) \models \exists y\psi(y)$. Let $P^R_{\Phi} = (\Delta^R, \tau^R_{\Phi})$ be the canonical model of $(T, (A_p \cup R))$, and let $J = \{\hat{p}_{i}\}_{i \geq 0}$ be the canonical model of $K$, where $\hat{p}_i = (\Delta, \Delta^R)$. There exists a homomorphism $\pi$ of $\exists y\psi(y)$ into $P^R_{\Phi}$. We first define a mapping $\sigma$ from $\{x^{R_i} \mid x \in N^K \text{ or occurs in } R\}$ into $\{x^{R_i} \mid x \in N^K \cup \Gamma_N, x \text{ occurs in } chase^{K}_{\mathcal{R}}(A_p)\}$, where we assume $\Delta$ and $\Delta^R$ be disjoint, by

- $\sigma(x^{R_i}) = x^{R_i}$ for $a \in N^K$.
- $\sigma(x^{R_i}_{p}) = x^{R_i}$ such that $P(a, x) \in chase^{K}_{\mathcal{R}}(A_p)$.
- $\sigma(x^{R_i}) = x^{R_i}$ such that there exists $P(a, x) \in \bigcup_{i=0}^{n} chase^{K}_{\mathcal{R}}(A_i)$.
- $\sigma(x^{R_i}_{p}) = x^{R_i}$ such that $P(a, x) \in chase^{K}_{\mathcal{R}}(A_p)$.

Claim 1. $\sigma$ is well-defined.

Proof of claim. If $x_{a^p}$ occurs in $R$, there exists $i$ such that $(T, A_i) \models \exists x_P(a, x)$. Since $I_i$ is a model of $(T, A_i)$, it follows that there is some $P(a, x) \in chase^{K}_{\mathcal{R}}(A_i)$. Moreover, since $P$ is rigid, $P(a, x) \in chase^{K}_{\mathcal{R}}(A_p)$.

If $x_{a^p}$ occurs in $R$, there exists $i$ such that $(T, A_i) \models \exists x_P(a, x)$. Since $I_i$ is a model of $(T, A_i)$, it follows that there exist $x, y \in N^K \cup \Gamma_N$ such that $P(a, x) \in chase^{K}_{\mathcal{R}}(A_i)$. Moreover, $x$ occurs in $chase^{K}_{\mathcal{R}}(A_p)$ because there exists $B := A[\exists R] \exists R$ with $A \in N_{RC}$. 


and \( R \in \mathbb{N}_R \) such that \( T \models \exists P^x \subseteq B \), and therefore there is a rigid assertion \( \beta \models B(x) \) such that \( \beta \in \text{chase}_{k}(A_{p}). \)

If \( x_{P^1} \) occurs in \( R \), then \( x_{p} \) also occurs in \( R \). It follows that there exist \( i \) and \( y \in \mathbb{N}_C \cup \Gamma_{N} \) such that \( P(y, \sigma(x_{P^1}^{y_{P^1}})) \in \text{chase}_{k}(A_{i}) \). Moreover, by construction of \( R \), \( P^{x} \) is rigid and such that \( T \models \exists P^{x} \subseteq \exists P \).

Since \( I_{p} \) is a model of \( T \), there then exists \( x \in \mathbb{N}_C \cup \Gamma_{N} \) such that \( P'((x_{P^1}^x), (y_{P^1}^x)) \in \text{chase}_{k}(A_{p}) \).

Hence, \( P'((x_{P^1}^x), (y_{P^1}^x)) \in \text{chase}_{k}(A_{p}). \)  

\[ \text{Claim 2.} \sigma \text{ is a partial homomorphism of } I_{p}^R \text{ into } I_{p}. \]

\text{Proof of claim.} For every \( a \in \mathbb{N}_C \) and concept \( A \), if \((a_{P^1}^x) \in \text{chase}_{k}(A_{p}) \), then \( I_{p}^R \) is the canonical model of \( T \), \( A_{p} \subseteq \exists A \). Therefore, \( I_{p}^R \) is a model of \( T \), \( A_{p} \subseteq \exists A \).

For every \( i \) \( x_{P^1} \) \( y_{P^1} \) \( A \), we obtain \( \sigma((a_{P^1}^x), (b_{P^1}^x)) \in \text{chase}_{k}(A_{p}). \)

For every pair \( a, b \in \mathbb{N}_C \) and role \( P \), if \((a_{P^1}^x), (b_{P^1}^x)) \in \text{chase}_{k}(A_{p}) \), then \( \sigma((a_{P^1}^x), (b_{P^1}^x)) \in \text{chase}_{k}(A_{p}). \)

For every \( x_{P^1} \) that occurs in \( R \) and \( A \in \mathbb{N}_C \), if \((a_{P^1}^x) \in \text{chase}_{k}(A_{p}) \), then \( I_{p}^R \) is the canonical model of \( T \), \( A_{p} \subseteq \exists A \). Therefore, \( I_{p}^R \) is a model of \( T \), \( A_{p} \subseteq \exists A \).

For every \( i \) \( x_{P^1} \) \( y_{P^1} \) \( A \), we obtain \( \sigma((a_{P^1}^x), (b_{P^1}^x)) \in \text{chase}_{k}(A_{p}). \)

Finally, for every \( x, y \notin \mathbb{N}_C \) that occur in \( R \) and for every role \( P \), if \((x_{P^1}^x), (y_{P^1}^x)) \in \text{chase}_{k}(A_{p}) \), then \( I_{p}^R \) is the canonical model of \( T \), \( A_{p} \subseteq \exists A \). Therefore, \( I_{p}^R \) is a model of \( T \), \( A_{p} \subseteq \exists A \).

From Claim 3, \( \sigma' \) is a homomorphism of \( I_{p}^R \) into \( I_{p} \).
For every \( x \sim y \) such that \( y \) occurs in \( \text{chase}_{K}(A) \), we set \( x_{aP_{1}} \mathrel{\triangleq} y \), where \( (h_{p}^{\mathcal{R}}(x_{aP_{1}}), y) \in P_{L}^{\mathcal{R}} \).

We show by induction on the length \( l = r \) of the sequence of rigid roles \( P_{r} \cdots P_{1} \) that there is always such a \( (h_{p}^{\mathcal{R}}(x_{aP_{1}}), y) \in P_{L}^{\mathcal{R}} \).

If \( l = 1 \), we are in one of the following cases.

(i) \( r > 1 \) and \( h_{p}^{\mathcal{R}}(x_{aP_{1}}) = x_{aP_{1}} \).

Then \( (x_{aP_{1}}), y) \in P_{L}^{\mathcal{R}} \) because \( P_{L}^{\mathcal{R}} \) is a model of \( \mathcal{R} \). Since \( x_{aP_{1}} \in \Gamma_{X} \), by (P4), \( T = \exists P_{r} \cdots P_{1} \). Therefore, since \( T_{L}^{\mathcal{R}} \) is a model of \( T \), there exists \( (x_{aP_{1}}, y) \in P_{L}^{\mathcal{R}} \).

For length > 1, \( T = \exists P_{r} \cdots P_{1} \). It follows that, since by inductive hypothesis there exists \( (x_{aP_{1}}), y) \in P_{L}^{\mathcal{R}} \), then there exists \( (h_{p}^{\mathcal{R}}(x_{aP_{1}}), y) \in P_{L}^{\mathcal{R}} \).

- Case \( x_{aP_{1}} \), such that for every \( j \in [1, l], i_{j} \neq p \), and role \( P_{j} \) that is not rigid, \( h_{p}^{\mathcal{R}}(x_{aP_{1}}) = x_{aP_{1}} \).

Since \( T_{L}^{\mathcal{R}} \) does not contain any role inclusion of the form \( P' \subseteq P \) with \( P' \cdots P_{1} = R_{1} \) then \( \exists R_{1} \exists R_{2} \), \( R_{1} = N_{a} \) and \( R_{2} = N_{a} \) and \( P_{1} \) is not rigid, there is no \( P_{1} \) such that \( P_{1} \subseteq P \) and \( P_{1} \) is rigid. Therefore, since \( i_{1} \neq p \), there is no \( P_{1} \) such that \( P(x_{aP_{1}}, x_{aP_{1}}) \in \text{chase}_{K}(A) \). We obtain that \( x_{aP_{1}} \) occurs in \( \text{chase}_{K}(A) \) only if there is a \( B_{1} := A \exists R \exists R' \) with \( A \in \mathcal{N}_{a} \), \( R \in \mathcal{R} \) such that \( \text{chase}_{K}(A) = B_{1}(x_{aP_{1}}) \). By (P3), \( T = \exists P_{r} \cdots P_{1} \). Then, (P4), \( x_{aP_{1}} \) appears in \( \mathcal{R} \).

- Case \( x_{aP_{1}} \), such that for every \( j \in [1, l], i_{j} \neq p \), and role \( P_{j} \) that is not rigid, \( h_{p}^{\mathcal{R}}(x_{aP_{1}}) = x_{aP_{1}} \).

We show by induction on the length \( l = r \) of the chain of roles that links \( x_{aP_{1}} \) to the first individual \( x_{aP_{1}} \) such that \( i_{1} \neq p \) and there is always
such \( h^R_p(x_{i_1, \ldots, i_p}^1, y) \in P^R_{i_1, \ldots, i_p} \).

- If \( l = 0 \), then \( i_p = p \) and there is no \( j < l \) such that \( i_j = p \). We are thus in one of the following cases. Either (i) \( h^R_p(x_{i_1, \ldots, i_p}^1, p) = a^p \), (ii) \( h^R_p(x_{i_1, \ldots, i_p}^1, x_{i_2, \ldots, i_p}^1) = x_{i_2, \ldots, i_p}^1 \), (iii) \( h^R_p(x_{i_1, \ldots, i_p}^1, x_{i_2, \ldots, i_p}^1) = \exists^R P_{i_1, \ldots, i_p} \), ((iv) \( h^R_p(x_{i_1, \ldots, i_p}^1, y) \in P^R_{i_1, \ldots, i_p}, \) y) \in P^R_{i_1, \ldots, i_p} \).

- If \( l > 0 \), then \( i_p = p \), and there exists \( j \) such that \( (h^R_p(x_{i_1, \ldots, i_p}^1, y) \in P^R_{i_1, \ldots, i_p} \).

\( T \mid (P8) \), either (i) there exists \( \exists a \), and \( A \in \mathbb{N} \). In the latter case, \( T \mid (P8) \) is a model of \( \mathcal{T} \), then \( \exists a \) is such that \( (T, \mathcal{A}) \models (a, \mathcal{A}) \). In the latter case, \( \mathcal{T} \) is a model of \( \mathcal{R} \), we obtain \( \exists \mathcal{R} \mid (P8) \).

\( \exists a \) is a homomorphism of \( \mathcal{T} \) into \( \exists \mathcal{R} \).

**Proof of claim** For every \( a \in \mathbb{N}^\mathcal{R} \) and concept \( A \), if \( a^p \in A^p \), i.e., \( A(a) \in \text{chase}^\mathcal{R}((a, \mathcal{A})) \), then by \( (P5) \), either (i) \( (\mathcal{T}, \mathcal{A}) \models (a, \mathcal{A}) \), and since \( T \mid (P8) \) is a model of \( (\mathcal{T}, \mathcal{A}) \), then \( h^R_p(a^p) = a^p \in A^p \), or (ii) there exists \( B = C \mid \exists a \), and if \( T \mid (P8) \) is a model of \( \mathcal{R} \), we obtain \( \exists \mathcal{R} \mid (P8) \) because \( \exists \mathcal{R} \mid (P8) \) is a model of \( T \). It follows that \( h^R_p((a^p, b^p)) = (a^p, b^p) \) in \( A^p \). For every pair \( a, b \in \mathbb{N}^\mathcal{R} \) and role \( R \), if \( (a^p, b^p) \in R^p \), by \( (P7) \), similar arguments can be used to prove that \( h^R_p((a^p, b^p)) = (a^p, b^p) \) in \( R^p \).

For every \( a \in \mathbb{N}^\mathcal{R} \), such that \( a^p \in A^p \), by \( (P3) \), \( T \mid (P8) \subseteq A \), and by construction of \( h^R \), \( h^R_p(x_{i_1, \ldots, i_p}^1, y) = y \) is such that either (i) there exists \( (x, y) \in P^R_{i_1, \ldots, i_p} \), and since \( T \mid (P8) \) is a model of \( T \), we have \( y \in A^p \), or (ii) \( y = x^p \). Since \( \mathcal{T} \) is not rigid, and for every \( j \in [1, l], i_j \neq p \). In the latter case, \( (P6) \), there exists \( \exists \mathcal{R} \mid \exists a \), and if \( (x, y) \in P^R_{i_1, \ldots, i_p} \), by \( (P3) \), \( T \mid (P8) \subseteq B \). Therefore, by construction of \( \exists \mathcal{R} \), we obtain \( (x, y) \in A^p \). It follows that \( y \in A^p \).

For every pair \( x, y \in \mathbb{N}^\mathcal{R} \) and role \( P \) such that \( (x^p, y^p) \in \mathcal{R} \), by \( (P10) \), either (i) \( x = x^p \), and if \( \mathcal{T} \mid (P10) \subseteq P \), or (ii) \( x = y^p \) and \( \mathcal{T} \mid (P10) \subseteq P \). We
can assume w.l.o.g. that we are in the first case. Otherwise, we consider \((y^T, x^T)\) \(\in P^T \cap R^T\). If \(i_{i+1} = p\), by definition of \(h^R_p\), we have \((h^R_p(x^T), h^R_p(y^T)) \in P^T_{i+1}\) otherwise, by (P11), there exists \(P' := R|R^−\) with \(R \in \mathbb{N}_{RR}\) such that \(T \models P_{i+1} \subseteq P' \subseteq P\) and \(P'(x, y) \in chase^N_k(A_{i+1})\). In this case, there are two possibilities.

(i) If \(P_i\) is not rigid, given that \(T \models P_{i+1} \subseteq P'\) and \(P'\) is rigid, \(P_{i+1}\) is rigid by our hypothesis on the TBox.

It follows that \(h^R_p(y^T) = x^T_{P_{i+1}}\). If there is no \(i_j = p\), then \(h^R_p(x^T) = x^T_{P_{i+1}}\). Therefore, since \(P_{i+1}(x_{P_i}, x_{P_{i+1}}) \in R\), then \((h^R_p(x^T), h^R_p(y^T)) \in P^T_{i+1}\). Otherwise, there exists \(i_j = p\), and we obtain \((h^R_p(x^T), h^R_p(y^T)) \in P^T_{i+1}\) by definition of \(h^R_p\).

(ii) If \(P_i\) is rigid, then \(h^R_p(y^T)\) is such that \((h^R_p(x^T), h^R_p(y^T)) \in P^T_{i+1}\).

Since in all cases, \((h^R_p(x^T), h^R_p(y^T)) \in P^T_{i+1}\) and \(P^T_R\) is a model of \(T\), we obtain that \(P^T_R\) contains \((h^R_p(x^T), h^R_p(y^T))\).

Finally, if \(a \in \mathbb{N}^R_k\) and \(x \in \Gamma_N\), we have that \((a^T, x^T) \in P^T_T\) only if \(x = x^T_{aP_i}\). If \(i_1 = p\), by definition of \(h^R_p\), \((h^R_p(a^T), h^R_p(x^T)) \in P^T_{i+1}\), and since by (P8), \(T \models P_1 \subseteq P\), we obtain that \((h^R_p(a^T), h^R_p(x^T)) \in P^T_{i+1}\). If \(i_1 \neq p\), by (P9), there exists a rigid role \(P'\) such that \(T \models P_1 \subseteq P' \subseteq P\), and by our restriction on the TBox, \(P_1\) must be rigid. By (P1) and (P8), there is some \(j\) such that \(\langle T, A_j \rangle \models \exists x.P_1(a, x)\). Therefore, \(P_1(a, x_{aP_i}) \in R\) and \((h^R_p(a^T), h^R_p(x^T)) = (a^T, x_{aP_i}) \in P^T_{i+1}\). We obtain that \((h^R_p(a^T), h^R_p(x^T)) \in P^T_R\).

It follows from Claim 2 that \(h^R_p \circ \pi\) is a homomorphism of \(\exists \check{y}.\psi(\check{y})\) into \(T^R_p\). Therefore, we have shown that \(T_p \models \exists \check{y}.\psi(\check{y})\) implies \(K_R, p \models \exists \check{y}.\psi(\check{y})\).

This means that \(K, p \models \exists \check{y}.\psi(\check{y})\) implies \(K_R, p \models \exists \check{y}.\psi(\check{y})\).