



HAL
open science

Temporal Query Answering in DL-Lite over Inconsistent Data

Camille Bourgaux, Anni-Yasmin Turhan

► **To cite this version:**

Camille Bourgaux, Anni-Yasmin Turhan. Temporal Query Answering in DL-Lite over Inconsistent Data. The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Oct 2017, Vienna, Austria. hal-02053254

HAL Id: hal-02053254

<https://inria.hal.science/hal-02053254>

Submitted on 1 Mar 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.

Temporal Query Answering in DL-Lite over Inconsistent Data^{*}

Camille Bourgaux and Anni-Yasmin Turhan

Technische Universität Dresden, Dresden, Germany
`firstname.lastname@tu-dresden.de`

Abstract. In ontology-based systems that process data stemming from different sources and that is received over time, as in context-aware systems, 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-based query answering. We consider a recently proposed temporal query language that combines conjunctive queries with operators of propositional linear temporal logic and extend to this setting three inconsistency-tolerant semantics that have been introduced for querying inconsistent description logic knowledge bases. We investigate their complexity for DL-Lite _{\mathcal{R}} temporal knowledge bases, and furthermore complete the picture for the consistent case.

1 Introduction

Context-aware systems [18, 3] observe their environment over time and are able to detect situations while running in order to adapt their behaviour. They rely upon heterogeneous sources such as sensors (in a broad sense) or other applications that provide them with data. A context-aware system needs to integrate this data and should behave resilient towards erroneous or contradictory data. 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. Moreover, it is convenient to use some knowledge about the system to reason with the data and get more complete answers to the queries that capture the situations to be recognized than from the data alone. To address these requirements and facilitate data integration, *ontologies* have been used to implement situation recognition [18, 3, 14, 25].

Ontology-mediated query answering [15] performs database-style query answering over description logic (DL) knowledge bases that consist of an ontology (called a TBox) expressing conceptual knowledge about a domain and a dataset (or ABox) containing facts about particular individuals [5]. An important issue that may arise when querying data through ontology reasoning is the inconsistency of the data w.r.t. the ontology. This is especially true for context-aware

^{*} Supported by the DFG in CRC 912 (HAEC) and the DAAD.

systems, since in the applications that need to perform situation recognition, the ABox is usually populated by frequently changing data from sensors or other sources. The problem is that under the classical semantics, every query is entailed from an inconsistent knowledge base and thus classical reasoners are rendered useless. Several inconsistency-tolerant semantics have been introduced for DL knowledge bases (see [7] for a survey) to remedy this problem.

A situation is often defined not only w.r.t. the current state of the system but depends also on its history. For instance, a system that operates on a cluster of servers may need the list of servers which have been almost overloaded at least twice in the past ten time units. Likewise in the medical domain a critical situation for a patient can depend on the patient’s medical history. That is why research efforts have recently been devoted to temporalizing query answering [4, 11] by allowing to use operators of *linear temporal logic* (LTL) [26] in the queries. In this setting, the query is answered over a *temporal knowledge base* consisting of a global TBox and a sequence of ABoxes that represents the data at different time points. The situation previously described can then be recognized by answering the query “ $\diamond^- (\text{AlmostOverloaded}(x) \wedge \circ^- \diamond^- \text{AlmostOverloaded}(x))$ ”, where \diamond^- is the LTL operator “eventually in the past” and \circ^- the operator “previous”, over the sequence of datasets that correspond to the last ten observations of the system, an ontology defining the concept `AlmostOverloaded`. A lot of work has been dedicated to the temporalization of DL, combining different temporal logics and DL languages (see [23] for a survey). As efficiency is a primary concern, particular attention has been paid to temporalized DLs of the DL-Lite family [16] (see [2] for different temporal extensions of DL-Lite). The DLs of this family cover an important fragment of the RDF query language SPARQL and underlie the OWL 2 QL profile of the Semantic Web standard [24]. They possess the notable property that query answering can be reduced to evaluation of standard database queries. The construction of temporal queries has also attracted a lot of interest recently [19, 20, 1], and querying temporal databases has been studied as well (see e.g., [17]). Here, we consider the setting proposed in [11] which does not allow for temporalized concepts or axioms in the TBox but focuses on querying sequences of ABoxes with temporal queries.

This work presents results on lifting inconsistency-tolerant reasoning to temporal query answering. To the best of our knowledge, this is the first investigation of temporal query answering under inconsistency-tolerant semantics. We consider three semantics that have been defined for DL knowledge bases and that we find particularly relevant. They are all based upon the notion of a *repair*, which is a maximal consistent subset of the data. The *AR semantics* [21, 22], inspired by consistent query answering in the database setting [6], considers the queries that hold in *every repair*. This semantics is arguably the most natural and is widely accepted to query inconsistent knowledge bases. However, AR query answering is intractable even for DL-Lite, which leads [21, 22] to propose a tractable approximation of AR, 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, so it may be interesting in our setting when the system should change its behaviour only if some situation has been recognized with a very high confidence. Finally, the *brave semantics* [9] 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 be handled imperatively.

The contributions of this paper are as follows. In Section 3 we extend the AR, IAR and brave semantics to the setting of temporal query answering. We distinguish in our analysis three cases for *rigid predicates*, i.e., whose extensions do not change between time points: no rigid predicates, rigid concepts only, or rigid concepts and roles. We show that when there is no rigid predicate, existing algorithms for temporal query answering and for IAR query answering can be combined to perform IAR temporal query answering. We also show that this method can sometimes be used for AR and provides in any case an approximation of the AR answers. In Section 4 we investigate the computational properties of the three semantics, considering both *data complexity* (in the size of the data only), and *combined complexity* (in the size of the whole problem), and distinguishing three different cases regarding the rigid symbols that are allowed. We show that in all cases except for brave semantics with rigid predicates, the data complexity is not higher than in the atemporal setting. In all cases, adding the temporal dimension does not increase the combined complexity. Our complexity analysis also leads us to close some open questions about temporal query answering under the classical semantics in the presence of rigid predicates. In particular, we show that it can often be reduced to the case without rigid predicates.

Detailed proofs of all the results are provided in [13].

2 Preliminaries

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

Syntax. A DL *knowledge base (KB)* \mathcal{K} consists of an ABox \mathcal{A} and a TBox \mathcal{T} , constructed from three countably infinite sets: a set \mathbf{N}_C of *concept names* (unary predicates), a set \mathbf{N}_R of *role names* (binary predicates), and a set \mathbf{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 $A \in \mathbf{N}_C$, $R \in \mathbf{N}_R$, $a, b \in \mathbf{N}_I$. The *TBox* (ontology) is a finite set of axioms whose form depends on the particular DL. In DL-Lite \mathcal{R} , TBox axioms are either *concept inclusions* $B \sqsubseteq C$ or *role inclusions* $P \sqsubseteq S$ built according to the following syntax (where $A \in \mathbf{N}_C$ and $R \in \mathbf{N}_R$):

$$B := A \mid \exists P, \quad C := B \mid \neg B, \quad P := R \mid R^-, \quad S := P \mid \neg P$$

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

Semantics. An *interpretation* has the form $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a non-empty set and $\cdot^{\mathcal{I}}$ maps each $a \in \mathbf{N}_I$ to $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, each $A \in \mathbf{N}_C$ to $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$,

and each $R \in \mathbf{N}_R$ to $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. We adopt the unique name assumption (i.e., for all $a, b \in \mathbf{N}_I$, $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ if $a \neq b$). The function $\cdot^{\mathcal{I}}$ is straightforwardly extended to general concepts and roles, e.g., $(R^-)^{\mathcal{I}} = \{(d, e) \mid (e, d) \in R^{\mathcal{I}}\}$ and $(\exists P)^{\mathcal{I}} = \{d \mid \exists e : (d, e) \in P^{\mathcal{I}}\}$. An interpretation \mathcal{I} satisfies an inclusion $G \sqsubseteq H$ if $G^{\mathcal{I}} \subseteq H^{\mathcal{I}}$; it satisfies $A(a)$ (resp. $R(a, b)$) if $a^{\mathcal{I}} \in A^{\mathcal{I}}$ (resp. $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$). We call \mathcal{I} a *model* of $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ if \mathcal{I} satisfies all axioms in \mathcal{T} and all assertions in \mathcal{A} . A KB is *consistent* if it has a model, and we say that an ABox \mathcal{A} is \mathcal{T} -*consistent* (or simply *consistent* for short), if the KB $\langle \mathcal{T}, \mathcal{A} \rangle$ is consistent.

Queries. A *conjunctive query* (CQ) takes the form $q = \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y})$, where ψ is a conjunction of atoms of the forms $A(t)$ or $R(t, t')$, with t, t' individuals or variables from $\mathbf{x} \cup \mathbf{y}$. A CQ is called *Boolean* (BCQ) if it has no free variables (i.e. $\mathbf{x} = \emptyset$). A BCQ q is *entailed* from \mathcal{K} , written $\mathcal{K} \models q$, iff q holds in every model of \mathcal{K} . Given a CQ q with free variables $\mathbf{x} = (x_1, \dots, x_k)$ and a tuple of individuals $\mathbf{a} = (a_1, \dots, a_k)$, \mathbf{a} is a *certain answer* to q over \mathcal{K} just in the case that $\mathcal{K} \models q(\mathbf{a})$, where $q(\mathbf{a})$ is the BCQ resulting from replacing each x_j by a_j .

Inconsistency-tolerant semantics. A *repair* of $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is an inclusion-maximal subset of \mathcal{A} that is \mathcal{T} -consistent. We consider three semantics based on repairs. A tuple \mathbf{a} is an answer to q over \mathcal{K} under

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

In DL-Lite $_{\mathcal{R}}$, IAR or brave CQ answering is in P w.r.t. data complexity (in the size of the ABox) and NP-complete w.r.t. combined complexity (in the size of the whole KB and the query), and AR CQ answering is coNP-complete w.r.t. data complexity and Π_2^P -complete w.r.t. combined complexity [21, 9].

Temporal query answering. We consider the framework presented in [11].

Definition 1 (TKB). A temporal knowledge base (TKB) $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{0 \leq i \leq n} \rangle$ consists of a TBox \mathcal{T} and a finite sequence of ABoxes $(\mathcal{A}_i)_{0 \leq i \leq n}$. A sequence $\mathcal{J} = (\mathcal{I}_i)_{0 \leq i \leq n}$ of interpretations $\mathcal{I}_i = (\Delta, \cdot^{\mathcal{I}_i})$ over a fixed non-empty domain Δ is a model of \mathcal{K} iff for all $0 \leq i \leq n$, \mathcal{I}_i is a model of $\langle \mathcal{T}, \mathcal{A}_i \rangle$, and for every $a \in \mathbf{N}_I$ and all $1 \leq i \leq j \leq n$, $a^{\mathcal{I}_i} = a^{\mathcal{I}_j}$. Rigid predicates are elements from the set of rigid concepts $\mathbf{N}_{\text{RC}} \subseteq \mathbf{N}_{\text{C}}$ or of rigid roles $\mathbf{N}_{\text{RR}} \subseteq \mathbf{N}_{\text{R}}$. A sequence of interpretations $\mathcal{J} = (\mathcal{I}_i)_{0 \leq i \leq n}$ respects the rigid predicates iff for every $X \in \mathbf{N}_{\text{RC}} \cup \mathbf{N}_{\text{RR}}$ and all $1 \leq i \leq j \leq n$, $X^{\mathcal{I}_i} = X^{\mathcal{I}_j}$. A TKB is consistent if it has a model that respects the rigid predicates. A sequence of ABoxes $(\mathcal{A}_i)_{0 \leq i \leq n}$ is \mathcal{T} -consistent, or simply consistent, if the TKB $\langle \mathcal{T}, (\mathcal{A}_i)_{0 \leq i \leq n} \rangle$ is consistent.

It is sometimes convenient to represent a sequence of ABoxes as a set of assertions associated with timestamps, which we call *timed-assertions*: $(\mathcal{A}_i)_{0 \leq i \leq n}$ becomes $\{(\alpha, i) \mid \alpha \in \mathcal{A}_i, 0 \leq i \leq n\}$. A *rigid assertion* is of the form $A(a)$ with $A \in \mathbf{N}_{\text{RC}}$

Table 1: Entailment of BTCQs

ϕ	$\mathcal{I}, p \models \phi$ iff
$\exists \mathbf{y} \psi(\mathbf{y})$	$\mathcal{I}_p \models \exists \mathbf{y} \psi(\mathbf{y})$
$\phi_1 \wedge \phi_2$	$\mathcal{I}, p \models \phi_1$ and $\mathcal{I}, p \models \phi_2$
$\phi_1 \vee \phi_2$	$\mathcal{I}, p \models \phi_1$ or $\mathcal{I}, p \models \phi_2$
$\bigcirc \phi_1$	$p < n$ and $\mathcal{I}, p+1 \models \phi_1$
$\bullet \phi_1$	$p < n$ implies $\mathcal{I}, p+1 \models \phi_1$
$\bigcirc^- \phi_1$	$p > 0$ and $\mathcal{I}, p-1 \models \phi_1$
$\bullet^- \phi_1$	$p > 0$ implies $\mathcal{I}, p-1 \models \phi_1$
$\Box \phi_1$	$\forall k, p \leq k \leq n, \mathcal{I}, k \models \phi_1$
$\Box^- \phi_1$	$\forall k, 0 \leq k \leq p, \mathcal{I}, k \models \phi_1$
$\Diamond \phi_1$	$\exists k, p \leq k \leq n, \mathcal{I}, k \models \phi_1$
$\Diamond^- \phi_1$	$\exists k, 0 \leq k \leq p, \mathcal{I}, k \models \phi_1$
$\phi_1 \mathbf{U} \phi_2$	$\exists k, p \leq k \leq n, \mathcal{I}, k \models \phi_2$ and $\forall j, p \leq j < k, \mathcal{I}, j \models \phi_1$
$\phi_1 \mathbf{S} \phi_2$	$\exists k, 0 \leq k \leq p, \mathcal{I}, k \models \phi_2$ and $\forall j, k < j \leq p, \mathcal{I}, j \models \phi_1$

or $R(a, b)$ with $R \in \mathbf{N}_{\text{RR}}$. We distinguish three cases in our analysis : Case 1 with $\mathbf{N}_{\text{RC}} = \mathbf{N}_{\text{RR}} = \emptyset$, Case 2 with $\mathbf{N}_{\text{RC}} \neq \emptyset$ and $\mathbf{N}_{\text{RR}} = \emptyset$, and Case 3 with $\mathbf{N}_{\text{RC}} \neq \emptyset$ and $\mathbf{N}_{\text{RR}} \neq \emptyset$. Note that since rigid roles can simulate rigid concepts, these three cases cover all possibilities. We denote by $\mathbf{N}_X^{\mathcal{K}}$ the elements of \mathbf{N}_X occurring in \mathcal{K} .

Definition 2 (TCQ). Temporal conjunctive queries (TCQs) are built from CQs as follows: each CQ is a TCQ, and if ϕ_1 and ϕ_2 are TCQs, then so are $\phi_1 \wedge \phi_2$ (conjunction), $\phi_1 \vee \phi_2$ (disjunction), $\bigcirc \phi_1$ (strong next), $\bullet \phi_1$ (weak next), $\bigcirc^- \phi_1$ (strong previous), $\bullet^- \phi_1$ (weak previous), $\Box \phi_1$ (always), $\Box^- \phi_1$ (always in the past), $\Diamond \phi_1$ (eventually), $\Diamond^- \phi_1$ (some time in the past), $\phi_1 \mathbf{U} \phi_2$ (until), and $\phi_1 \mathbf{S} \phi_2$ (since). Given a TCQ ϕ with free variables $\mathbf{x} = (x_1, \dots, x_k)$ and a tuple of individuals $\mathbf{a} = (a_1, \dots, a_k)$, $\phi(\mathbf{a})$ denotes the Boolean TCQ (BTCQ) resulting from replacing each x_j by a_j . The tuple \mathbf{a} is an answer to ϕ in a sequence of interpretations $\mathcal{I} = (\mathcal{I}_i)_{0 \leq i \leq n}$ at time point p ($0 \leq p \leq n$) iff $\mathcal{I}, p \models \phi(\mathbf{a})$, where the entailment of a BTCQ ϕ is defined by induction on its structure as shown in Table 1. It is a certain answer to ϕ over \mathcal{K} at time point p , written $\mathcal{K}, p \models \phi(\mathbf{a})$, iff $\mathcal{I}, p \models \phi(\mathbf{a})$ for every model \mathcal{I} of \mathcal{K} that respects the rigid predicates.

Thus, TCQ answering is straightforwardly reduced to entailment of BTCQs and we can focus w.l.o.g. on the latter problem.

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 predicates, a TKB $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{0 \leq i \leq n} \rangle$ may be inconsistent even if each KB $\langle \mathcal{T}, \mathcal{A}_i \rangle$ is consistent. In this case there need not exist a sequence of interpretations $\mathcal{I} = (\mathcal{I}_i)_{0 \leq i \leq n}$ such that each \mathcal{I}_i is a model of $\langle \mathcal{T}, \mathcal{A}_i \rangle$ and which respects

rigid predicates. That is why we need to consider as repairs the \mathcal{T} -consistent sequences of subsets of the initial ABoxes that are component-wise maximal.

Definition 3 (Repair of a TKB). A repair of a TKB $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{0 \leq i \leq n} \rangle$ is a sequence of ABoxes $(\mathcal{A}'_i)_{0 \leq i \leq n}$ such that $\{(\alpha, i) \mid \alpha \in \mathcal{A}'_i, 0 \leq i \leq n\}$ is a maximal \mathcal{T} -consistent subset of $\{(\alpha, i) \mid \alpha \in \mathcal{A}_i, 0 \leq i \leq n\}$. We denote the set of repairs of \mathcal{K} by $\text{Rep}(\mathcal{K})$.

The next example shows the influence of rigid predicates on the repairs.

Example 1. Consider the following TKB $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{1 \leq i \leq 2} \rangle$. 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.

$$\begin{aligned} \mathcal{T} &= \{\text{WebServer} \sqsubseteq \text{Server}, \text{AppServer} \sqsubseteq \text{Server}, \text{WebServer} \sqsubseteq \neg \text{AppServer}\} \\ \mathcal{A}_1 &= \{\text{WebServer}(a), \text{execute}(a, b)\} \\ \mathcal{A}_2 &= \{\text{AppServer}(a), \text{WebServer}(a), \text{execute}(a, c)\} \end{aligned}$$

Assume that no predicate is rigid. The TKB \mathcal{K} is inconsistent because the timed-assertions $(\text{AppServer}(a), 2)$ and $(\text{WebServer}(a), 2)$ violate the negative inclusion of \mathcal{T} , since $\text{AppServer}(a)$ and $\text{WebServer}(a)$ cannot both be true at time point 2. It follows that \mathcal{K} has two repairs $(\mathcal{A}'_i)_{1 \leq i \leq 2}$ and $(\mathcal{A}''_i)_{1 \leq i \leq 2}$ with $\mathcal{A}'_1 = \mathcal{A}''_1 = \mathcal{A}_1$, and $\mathcal{A}'_2 = \{\text{AppServer}(a), \text{execute}(a, c)\}$ and $\mathcal{A}''_2 = \{\text{WebServer}(a), \text{execute}(a, c)\}$ which correspond to the two different ways of restoring consistency.

Assume now that **AppServer** is rigid. There is a new reason for \mathcal{K} being inconsistent: the timed-assertions $(\text{WebServer}(a), 1)$ and $(\text{AppServer}(a), 2)$ violate the negative inclusion of \mathcal{T} due to the rigidity of **AppServer** which implies that $\text{AppServer}(a)$ and $\text{WebServer}(a)$ should be both entailed at time point 1. Then \mathcal{K} has two repairs $(\mathcal{A}'_i)_{1 \leq i \leq 2}$ and $(\mathcal{A}''_i)_{1 \leq i \leq 2}$ with $\mathcal{A}'_1 = \{\text{execute}(a, b)\}$, $\mathcal{A}'_2 = \{\text{AppServer}(a), \text{execute}(a, c)\}$, and $\mathcal{A}''_1 = \mathcal{A}_1$, $\mathcal{A}''_2 = \{\text{WebServer}(a), \text{execute}(a, c)\}$. Note that even if $(\mathcal{A}'_i)_{1 \leq i \leq 2}$ is maximal (since adding $\text{WebServer}(a)$ to \mathcal{A}'_1 renders the TKB inconsistent), \mathcal{A}'_1 is not a repair of $\langle \mathcal{T}, \mathcal{A}_1 \rangle$ since it is not maximal.

Next we extend the semantics AR, IAR, and brave to the temporal case in the natural way by regarding sequences of ABoxes.

Definition 4 (AR, IAR, brave semantics for TCQs). A tuple \mathbf{a} is an answer to a TCQ ϕ over a TKB $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{0 \leq i \leq n} \rangle$ at time point p under

- AR semantics, written $\mathcal{K}, p \models_{\text{AR}} \phi(\mathbf{a})$,
iff $\langle \mathcal{T}, (\mathcal{A}'_i)_{0 \leq i \leq n} \rangle, p \models \phi(\mathbf{a})$ for every repair $(\mathcal{A}'_i)_{0 \leq i \leq n}$ of \mathcal{K} ;
- IAR semantics, written $\mathcal{K}, p \models_{\text{IAR}} \phi(\mathbf{a})$,
iff $\langle \mathcal{T}, (\mathcal{A}^{\text{IR}}_i)_{0 \leq i \leq n} \rangle, p \models \phi(\mathbf{a})$, with $\mathcal{A}^{\text{IR}}_i = \bigcap_{(\mathcal{A}'_j)_{0 \leq j \leq n} \in \text{Rep}(\mathcal{K})} \mathcal{A}'_i$, $0 \leq i \leq n$;
- brave semantics, written $\mathcal{K}, p \models_{\text{brave}} \phi(\mathbf{a})$,
iff $\langle \mathcal{T}, (\mathcal{A}'_i)_{0 \leq i \leq n} \rangle, p \models \phi(\mathbf{a})$ for some repair $(\mathcal{A}'_i)_{0 \leq i \leq n}$ of \mathcal{K} .

The following relationships between the semantics are implied by their definition:

$$\mathcal{K}, p \models_{\text{IAR}} \phi(\mathbf{a}) \quad \Rightarrow \quad \mathcal{K}, p \models_{\text{AR}} \phi(\mathbf{a}) \quad \Rightarrow \quad \mathcal{K}, p \models_{\text{brave}} \phi(\mathbf{a})$$

Next, we illustrate the effect of the different semantics in the temporal case.

Example 2 (Example 1 cont'd). Consider the three temporal conjunctive queries:

$$\begin{aligned}\phi_1 &= \Box(\exists y \text{ execute}(x, y)) & \phi_2 &= \Box(\exists y \text{ Server}(x) \wedge \text{execute}(x, y)) \\ \phi_3 &= \Box(\exists y \text{ AppServer}(x) \wedge \text{execute}(x, y))\end{aligned}$$

In **Case 1** with no rigid predicate, the intersection of the repairs is $(\mathcal{A}_i^{IR})_{1 \leq i \leq 2}$ with $\mathcal{A}_1^{IR} = \mathcal{A}_1$, $\mathcal{A}_2^{IR} = \{\text{execute}(a, c)\}$. Then $\mathcal{K}, 1 \models_{\text{IAR}} \phi_1(a)$, since in every model of the intersection of the repairs a executes b at time point 1 and c at time point 2. For ϕ_2 , $\mathcal{K}, 1 \models_{\text{AR}} \phi_2(a)$, since every model of every repair assigns a to **WebServer** at time point 1 and either to **AppServer** (in models of $(\mathcal{A}'_i)_{1 \leq i \leq 2}$) or to **WebServer** (in models of $(\mathcal{A}''_i)_{1 \leq i \leq 2}$) at time point 2, but $\mathcal{K}, 1 \not\models_{\text{IAR}} \phi_2(a)$. Finally, $\mathcal{K}, 1 \not\models_{\text{brave}} \phi_3(a)$ because no repair entails **AppServer**(a) at time point 1.

If **AppServer** is rigid, the intersection of the repairs is $(\mathcal{A}_i^{IR})_{1 \leq i \leq 2}$ with $\mathcal{A}_1^{IR} = \{\text{execute}(a, b)\}$, $\mathcal{A}_2^{IR} = \{\text{execute}(a, c)\}$. So still $\mathcal{K}, 1 \models_{\text{IAR}} \phi_1(a)$ holds. Since every model of every repair assigns a to **Server** at time points 1 and 2 (either because a is a web server or an application server), $\mathcal{K}, 1 \models_{\text{AR}} \phi_2(a)$, but $\mathcal{K}, 1 \not\models_{\text{IAR}} \phi_2(a)$. Finally, $\mathcal{K}, 1 \models_{\text{brave}} \phi_3(a)$ because every model of $\langle \mathcal{T}, (\mathcal{A}'_i)_{1 \leq i \leq 2} \rangle$ assigns a to **AppServer** at any time point by rigidity of **AppServer**, but $\mathcal{K}, 1 \not\models_{\text{AR}} \phi_3(a)$.

We point out some characteristics of **Case 1**. Since there is no rigid predicate, the interpretations \mathcal{I}_i of a model $\mathcal{J} = (\mathcal{I}_i)_{0 \leq i \leq n}$ of \mathcal{K} that respects the rigid predicates are independent, besides the interpretation of the constants.

Proposition 1. *If $\mathbf{N}_{\text{RC}} = \mathbf{N}_{\text{RR}} = \emptyset$, then $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{0 \leq i \leq n} \rangle$ is consistent iff every $\langle \mathcal{T}, \mathcal{A}_i \rangle$ is consistent. Moreover, if \mathcal{K} is consistent, for every $0 \leq p \leq n$, \mathcal{I}'_p is a model of $\langle \mathcal{T}, \mathcal{A}_p \rangle$ iff there exists a model $(\mathcal{I}_i)_{0 \leq i \leq n}$ of \mathcal{K} such that $\mathcal{I}_p = \mathcal{I}'_p$.*

Proposition 1 has important consequences. First, the repairs of \mathcal{K} are all possible sequences $(\mathcal{A}'_i)_{0 \leq i \leq n}$ where \mathcal{A}'_i is a repair of $\langle \mathcal{T}, \mathcal{A}_i \rangle$, so the intersection of the repairs of \mathcal{K} is $(\mathcal{A}_i^\cap)_{0 \leq i \leq n}$ where \mathcal{A}_i^\cap is the intersection of the repairs of $\langle \mathcal{T}, \mathcal{A}_i \rangle$. Second, we show that the entailment (resp. IAR entailment) of a BTCQ from a consistent (resp. inconsistent) DL-Lite $_{\mathcal{R}}$ TKB can be equivalently defined w.r.t. the entailment (resp. IAR entailment) of the BCQs it contains as follows:

Proposition 2. *If \mathcal{K} is a DL-Lite $_{\mathcal{R}}$ TKB and $\mathbf{N}_{\text{RC}} = \mathbf{N}_{\text{RR}} = \emptyset$, the entailments shown in Table 2 hold for $S = \text{classical}$ when \mathcal{K} is consistent, and for $S = \text{IAR}$.*

This is a remarkable result, since it implies that answering temporal CQs under IAR semantics can be done with the algorithms developed for the consistent case [10, 11] by replacing classical CQ answering by IAR CQ answering (see [22, 8, 27] for algorithms). The following example shows that this is unfortunately not true for brave or AR semantics.

Example 3. Consider the following TKB $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{1 \leq i \leq n} \rangle$ and TCQ ϕ .

$$\mathcal{T} = \{T \sqsubseteq \neg F\} \quad \mathcal{A}_i = \{T(a), F(a)\} \text{ for } 1 \leq i \leq n \quad \phi = \Box^-(T(a) \wedge \bullet^- F(a))$$

Now, $\mathcal{K}, k \models_{\text{brave}} T(a) \wedge \bullet^- F(a)$ for every $0 \leq k \leq n$, but $\mathcal{K}, n \not\models_{\text{brave}} \phi$. This is because the same repair cannot entail $T(a) \wedge \bullet^- F(a)$ both at time point k and

Table 2: Entailment under classical or IAR semantics without rigid predicates

ϕ	$\mathcal{K}, p \models_S \phi$ iff
$\exists \mathbf{y} \psi(\mathbf{y})$	$\langle \mathcal{T}, \mathcal{A}_p \rangle \models_S \exists \mathbf{y} \psi(\mathbf{y})$
$\phi_1 \wedge \phi_2$	$\mathcal{K}, p \models_S \phi_1$ and $\mathcal{K}, p \models_S \phi_2$
$\phi_1 \vee \phi_2$	$\mathcal{K}, p \models_S \phi_1$ or $\mathcal{K}, p \models_S \phi_2$
$\circ \phi_1$	$p < n$ and $\mathcal{K}, p + 1 \models_S \phi_1$
$\bullet \phi_1$	$p < n$ implies $\mathcal{K}, p + 1 \models_S \phi_1$
$\circ^- \phi_1$	$p > 0$ and $\mathcal{K}, p - 1 \models_S \phi_1$
$\bullet^- \phi_1$	$p > 0$ implies $\mathcal{K}, p - 1 \models_S \phi_1$
$\square \phi_1$	$\forall k, p \leq k \leq n, \mathcal{K}, k \models_S \phi_1$
$\square^- \phi_1$	$\forall k, 0 \leq k \leq p, \mathcal{K}, k \models_S \phi_1$
$\diamond \phi_1$	$\exists k, p \leq k \leq n, \mathcal{K}, k \models_S \phi_1$
$\diamond^- \phi_1$	$\exists k, 0 \leq k \leq p, \mathcal{K}, k \models_S \phi_1$
$\phi_1 \mathbf{U} \phi_2$	$\exists k, p \leq k \leq n, \mathcal{K}, k \models_S \phi_2$ and $\forall j, p \leq j < k, \mathcal{K}, j \models_S \phi_1$
$\phi_1 \mathbf{S} \phi_2$	$\exists k, 0 \leq k \leq p, \mathcal{K}, k \models_S \phi_2$ and $\forall j, k < j \leq p, \mathcal{K}, j \models_S \phi_1$

$k + 1$, since it would contain both $(T(a), k)$ and $(F(a), k)$ which is not possible. For AR semantics, consider $\phi = T(a) \vee F(a)$ over the TKB \mathcal{K} : while ϕ 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 $\wedge, \circ, \bullet, \circ^-, \bullet^-, \square, \square^-$, 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 [8] for algorithms). Moreover, contrary to the brave semantics, this method still provides a *sound approximation* of AR answers even for unrestricted TCQs, since for all operators, the “if” direction from Table 2 is true.

4 Complexity Analysis for DL-Lite \mathcal{R}

The complexity of TCQ answering under the classical semantics in DL-Lite \mathcal{R} with *negations* in the query has been shown ALOGTIME-complete w.r.t. data complexity and PSPACE-complete w.r.t. combined complexity, rigid concepts and roles being present or not [12]. In our case, i.e., without negations, CQ evaluation over databases provides a NP lower bound for combined complexity and it has been shown in [10, 11] that TCQs in DL-Lite \mathcal{R} are rewritable so that they can be answered over a temporal database—albeit for a restricted setting without rigid roles and with rigid concepts only for TCQs that are rooted. The NP membership of TCQ answering in Case 1 for combined complexity is implied by this latter work as follows: it is possible to guess for each time point i and CQ q from the TCQ either a rewriting q' of q that holds in \mathcal{A}_i together with the rewriting steps that produce q' and the variables assignment that maps q' in \mathcal{A}_i , or to guess “false”. Checking that q' is indeed a rewriting of q and holds in \mathcal{A}_i can be done in polynomial time and there are polynomially many such pairs of

a time point and a CQ to test. Moreover, verifying that the propositional LTL formula obtained by replacing the CQs by propositional variables is satisfied by the sequence of truth assignments that assigns the propositional abstraction of q to **false** at time point i if “false” has been guessed and to **true** otherwise is in P since the formula does not contain negation. It follows that TCQ answering is NP-complete w.r.t. combined complexity. To alleviate the limitations imposed in [10, 11], we first show that TCQ answering without negations is NP-complete w.r.t. combined complexity even in the presence of rigid concepts and roles, with the restriction that a rigid role can only have rigid sub-roles. Indeed, we show that under this restriction, TCQ answering in **Case 3** can be reduced to TCQ answering in **Case 1** by adding to every ABox a set of assertions that models rigid consequences of the TKB and is computable in polynomial time.

For the remainder of this section, $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{0 \leq i \leq n} \rangle$ is a DL-Lite $_{\mathcal{R}}$ TKB and ϕ is a BTCQ. The set of constants of ϕ is denoted by \mathbb{N}_1^ϕ . We make use of the following notations: for a role P and two constants or variables x and y , $P^- := S$ if $P = S^-$ and $P(x, y)$ denotes $S(x, y)$ if $P = S$ and $S(y, x)$ if $P = S^-$. We assume w.l.o.g. that no $x \in \mathbb{N}_1^\mathcal{K}$ is of the form x_w^e where w, e are words built over $\mathbb{N}_1^\mathcal{K} \cup \mathbb{N}_C^\mathcal{K} \cup \mathbb{N}_R^\mathcal{K}$ and \mathbb{N} respectively.

As a first step, we assume that \mathcal{K} is consistent and construct a model $\mathcal{J}_{\mathcal{K}}$ of \mathcal{K} such that for any Boolean *conjunctive query* $q = \exists \mathbf{y} \psi(\mathbf{y})$ such that $\mathbb{N}_1^q \subseteq \mathbb{N}_1^\mathcal{K}$, $\mathcal{K}, p \models q$ iff $\mathcal{J}_{\mathcal{K}}, p \models q$. We build a sequence of (possibly infinite) ABoxes $(\text{chase}_{\text{rig}}^\mathcal{K}(\mathcal{A}_i))_{0 \leq i \leq n}$ similar to the chase presented in [15] for KBs. Let \mathcal{S} be a set of DL-Lite $_{\mathcal{R}}$ assertions. A PI α is applicable in \mathcal{S} to an assertion $\beta \in \mathcal{S}$ if

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

Applying a PI α to an assertion β means adding a new suitable assertion β_{new} to \mathcal{S} such that α is not applicable to β in $\mathcal{S} \cup \{\beta_{\text{new}}\}$.

Definition 5 (Rigid chase of a TKB). Let $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{0 \leq i \leq n} \rangle$ be a DL-Lite $_{\mathcal{R}}$ TKB. Let $(\mathcal{A}'_i)_{0 \leq i \leq n} = (\mathcal{A}_i \cup \{\beta \mid \exists k, \beta \in \mathcal{A}_k \text{ and } \beta \text{ is rigid}\})_{0 \leq i \leq n}$, let \mathcal{T}_p be the set of positive inclusions in \mathcal{T} , and let N_i be the number of assertions in \mathcal{A}'_i . Assume that the assertions of each \mathcal{A}'_i are numbered from $N_1 + \dots + N_{i-1} + 1$ to $N_1 + \dots + N_i$ following their lexicographic order. Consider the sequences of sets of assertions $\mathcal{S}^j = (\mathcal{S}'_i)_{0 \leq i \leq n}$ defined as follows:

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

where \mathcal{S}^{new} is defined in terms of the assertion β_{new} obtained from: let $\beta \in \mathcal{S}'_{i_\beta}$ be the first assertion in \mathcal{S}'_{i_β} such that there is a PI in \mathcal{T}_p applicable in \mathcal{S}'_{i_β} to β and α be the lexicographically first PI applicable in \mathcal{S}'_{i_β} to β . If α, β are of the form

- $\alpha = A_1 \sqsubseteq A_2$ and $\beta = A_1(a)$ then $\beta_{\text{new}} = A_2(a)$

- $\alpha = A \sqsubseteq \exists P$ and $\beta = A(a)$ then $\beta_{new} = P(a, a_{new})$
- $\alpha = \exists P \sqsubseteq A$ and $\beta = P(a, b)$ then $\beta_{new} = A(a)$
- $\alpha = \exists P_1 \sqsubseteq \exists P$ and $\beta = P_1(a, b)$ then $\beta_{new} = P(a, a_{new})$
- $\alpha = P_1 \sqsubseteq P_2$ and $\beta = P_1(a_1, a_2)$ then $\beta_{new} = P_2(a_1, a_2)$

where a_{new} is constructed from α and β as follows:

- if $a \in \mathbf{N}_1^K$ then $a_{new} = x_{aP}^{i_\beta}$
- otherwise $a \notin \mathbf{N}_1^K$, then let $a = x_{a'P_1 \dots P_i}^{i_1 \dots i_i}$ and define $a_{new} = x_{a'P_1 \dots P_i P}^{i_1 \dots i_i i_\beta}$.

If β_{new} is rigid, then $\mathcal{S}^{new} = (\{\beta_{new}\})_{0 \leq i \leq n}$, otherwise, $\mathcal{S}^{new} = (\mathcal{S}_i^{new})_{0 \leq i \leq n}$ with $\mathcal{S}_{i_\beta}^{new} = \{\beta_{new}\}$ and $\mathcal{S}_i^{new} = \emptyset$ for $i \neq i_\beta$. Let N be the total number of assertions in \mathcal{S}^j . If β_{new} is not rigid, β_{new} is numbered by $N + 1$, otherwise for every $0 \leq i \leq n$, the assertion $\beta_{new} \in \mathcal{S}_i^{new}$ added to \mathcal{S}_i^j is numbered by $N + 1 + i$.

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

$$chase_{rig}(\mathcal{K}) = (chase_{rig}^{\mathcal{K}}(\mathcal{A}_i))_{0 \leq i \leq n} = \bigcup_{j \in \mathbb{N}} \mathcal{S}^j = (\bigcup_{j \in \mathbb{N}} \mathcal{S}_i^j)_{0 \leq i \leq n}.$$

If \mathcal{K} is consistent, let $\mathcal{I}_{\mathcal{K}} = (\mathcal{I}_i)_{0 \leq i \leq n}$ where $\mathcal{I}_i = (\Delta, \cdot^{\mathcal{I}_i})$ is defined as follows: $\Delta = \mathbf{N}_1^K \cup \Gamma_N$ where Γ_N is the set of individuals that appear in $chase_{rig}(\mathcal{K})$ but not in \mathcal{K} , $a^{\mathcal{I}_i} = a$ for every $a \in \Delta$, $A^{\mathcal{I}_i} = \{a \mid A(a) \in chase_{rig}^{\mathcal{K}}(\mathcal{A}_i)\}$ for every $A \in \mathbf{N}_{\mathcal{C}}$, and $R^{\mathcal{I}_i} = \{(a, b) \mid R(a, b) \in chase_{rig}^{\mathcal{K}}(\mathcal{A}_i)\}$ for every $R \in \mathbf{N}_{\mathcal{R}}$. Then:

Lemma 1. $\mathcal{I}_{\mathcal{K}}$ is a model of \mathcal{K} that respects the rigid predicates, and for any BCQ $q = \exists \mathbf{y} \psi(\mathbf{y})$ such that $\mathbf{N}_1^q \subseteq \mathbf{N}_1^K$, $\mathcal{K}, p \models q$ iff $\mathcal{I}_{\mathcal{K}}, p \models q$ iff $\mathcal{I}_p \models q$.

We want to construct in polynomial time a set of assertions \mathcal{R} that captures all relevant information about rigid concepts and roles for TCQ answering. Without any restriction on the TBox, \mathcal{R} may be infinite. To see this, consider \mathcal{K} with $\mathcal{T} = \{\exists R^- \sqsubseteq \exists R, R \sqsubseteq S\}$ with S rigid, $\mathcal{A}_0 = \{R(a, b)\}$, $\mathcal{A}_i = \emptyset$ for $1 \leq i \leq n$. Since a model of \mathcal{K} that respects rigid predicates is such that $\phi = \exists x_1 \dots x_{k+1} S(x_1, x_2) \wedge \dots \wedge S(x_k, x_{k+1})$ holds for any $k > 0$ and at any time point, but can be such that no cycle of S , nor $\exists xy R(x, y)$ holds at some time point $i > 0$, \mathcal{R} has to contain an infinite chain of S . Therefore we assume the restriction that rigid roles only have rigid sub-roles, i.e., \mathcal{T} does not entail any role inclusion of the form $P_1 \sqsubseteq P_2$ with $P_1 := R_1 | R_1^-$, $R_1 \in \mathbf{N}_{\mathcal{R}} \setminus \mathbf{N}_{\mathcal{RR}}$ and $P_2 := R_2 | R_2^-$, $R_2 \in \mathbf{N}_{\mathcal{RR}}$.

Proposition 3. Let \mathcal{R} be as follows:

$$\begin{aligned} \mathcal{R} = & \{A(a) \mid A \in \mathbf{N}_{\mathcal{RC}}^{\mathcal{K}}, a \in \mathbf{N}_1^K, \exists i, \langle \mathcal{T}, \mathcal{A}_i \rangle \models_{brave} A(a)\} \cup \\ & \{R(a, b) \mid R \in \mathbf{N}_{\mathcal{RR}}^{\mathcal{K}}, a, b \in \mathbf{N}_1^K, \exists i, \langle \mathcal{T}, \mathcal{A}_i \rangle \models_{brave} R(a, b)\} \cup \\ & \{P(a, x_{aP}) \mid R \in \mathbf{N}_{\mathcal{RR}}^{\mathcal{K}}, P := R | R^-, a \in \mathbf{N}_1^K, \exists i, \langle \mathcal{T}, \mathcal{A}_i \rangle \models_{brave} \exists x P(a, x)\} \cup \\ & \{A(x_{P_1}) \mid S \in \mathbf{N}_{\mathcal{R}}^{\mathcal{K}} \setminus \mathbf{N}_{\mathcal{RR}}^{\mathcal{K}}, P_1 := S | S^-, A \in \mathbf{N}_{\mathcal{RC}}^{\mathcal{K}}, \\ & \quad \exists i, \langle \mathcal{T}, \mathcal{A}_i \rangle \models_{brave} \exists xy P_1(x, y) \text{ and } \mathcal{T} \models \exists P_1^- \sqsubseteq A\} \cup \\ & \{P_2(x_{P_1}, x_{P_1 P_2}) \mid S \in \mathbf{N}_{\mathcal{R}}^{\mathcal{K}} \setminus \mathbf{N}_{\mathcal{RR}}^{\mathcal{K}}, P_1 := S | S^-, R \in \mathbf{N}_{\mathcal{RR}}^{\mathcal{K}}, P_2 := R | R^-, \\ & \quad \exists i, \langle \mathcal{T}, \mathcal{A}_i \rangle \models_{brave} \exists xy P_1(x, y) \text{ and } \mathcal{T} \models \exists P_1^- \sqsubseteq \exists P_2\} \end{aligned}$$

The set \mathcal{R} is computable in polynomial time and such that (i) \mathcal{K} is consistent iff $\mathcal{K}_{\mathcal{R}} = \langle \mathcal{T}, (\mathcal{A}_i \cup \mathcal{R})_{0 \leq i \leq n} \rangle$ is consistent with $\mathbf{N}_{\text{RC}} = \mathbf{N}_{\text{RR}} = \emptyset$, and (ii) for any BTCQ ϕ such that $\mathbf{N}_1^\phi \subseteq \mathbf{N}_1^{\mathcal{K}}$, $\mathcal{K}, p \models \phi$ iff $\mathcal{K}_{\mathcal{R}}, p \models \phi$ with $\mathbf{N}_{\text{RC}} = \mathbf{N}_{\text{RR}} = \emptyset$.

The size of \mathcal{R} is polynomial in the size of $\mathbf{N}_{\mathcal{C}}^{\mathcal{K}}, \mathbf{N}_{\mathcal{R}}^{\mathcal{K}}$, and $\mathbf{N}_1^{\mathcal{K}}$, and since atomic query answering under brave semantics as well as subsumption checking can be done in polynomial time, \mathcal{R} can be computed in P. The first three parts of \mathcal{R} retain information about the participation of individuals of $\mathbf{N}_1^{\mathcal{K}}$ in rigid predicates. The last two witness the participation in rigid predicates of the role-successors w.r.t. non-rigid roles, thus take into account also anonymous individuals that are created in $\text{chase}_{\text{rig}}(\mathcal{K})$ when applying PIs whose right-hand side is an existential restriction with a non-rigid role. Note that the individuals created in $\text{chase}_{\text{rig}}(\mathcal{K})$ when applying such a PI with a rigid role are witnessed by the x_{aP} or $x_{P_1P_2}$ if they do not follow from a rigid role assertion. They do not need to be witnessed otherwise, since the assertion $P_2(x_{P_1}, x_{P_1P_2})$ is sufficient to trigger all the anonymous part implied by the fact that $x_{P_1P_2}$ is in the range of P_2 .

The key point of the proof for Claim (i) in Proposition 3, is that a minimal inconsistent subset of \mathcal{K} of the form $(\alpha, i), (\beta, j)$ with $i \neq j$ entails the violation of a NI that involves a rigid predicate, and that the rigid consequences of α and β are captured by \mathcal{R} . For the other direction, the main idea is that a minimal inconsistent subset of $\mathcal{K}_{\mathcal{R}}$ of the form $(\alpha, i), (\beta, i)$ with $\alpha \in \mathcal{R}$ is such that there is some $\alpha' \in \mathcal{A}_j$ that triggered the addition of α in \mathcal{R} , and a model of \mathcal{K} that respects the rigid predicates should satisfy both β and the rigid consequences of α' at time point i . To prove Claim (ii) of Proposition 3, we first show that for any Boolean conjunctive query $q = \exists \mathbf{y} \psi(\mathbf{y})$ such that $\mathbf{N}_1^q \subseteq \mathbf{N}_1^{\mathcal{K}}$, $\mathcal{J}_{\mathcal{K}}, p \models q$ iff $\mathcal{K}_{\mathcal{R}}, p \models q$ with $\mathbf{N}_{\text{RC}} = \mathbf{N}_{\text{RR}} = \emptyset$ by defining homomorphisms between \mathcal{I}_p and the canonical model of $\langle \mathcal{T}, (\mathcal{A}_p \cup \mathcal{R}) \rangle$.

Lemma 2. *If $q = \exists \mathbf{y} \psi(\mathbf{y})$ is such that $\mathbf{N}_1^q \subseteq \mathbf{N}_1^{\mathcal{K}}$, then $\mathcal{I}_p \models q$ iff $\mathcal{K}_{\mathcal{R}}, p \models q$.*

We then show by induction on the structure of the BTCQ ϕ that if $\mathbf{N}_1^\phi \subseteq \mathbf{N}_1^{\mathcal{K}}$, then $\mathcal{K}, p \models \phi$ iff $\mathcal{K}_{\mathcal{R}}, p \models \phi$ with $\mathbf{N}_{\text{RC}} = \mathbf{N}_{\text{RR}} = \emptyset$, so that TCQ answering over \mathcal{K} in Case 3 can be done by TCQ answering over $\mathcal{K}_{\mathcal{R}}$ in Case 1 and pruning answers that contain individual names not from $\mathbf{N}_1^{\mathcal{K}}$. Note that a model of $\mathcal{K}_{\mathcal{R}}$ 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 $\mathcal{K}_{\mathcal{R}}$ 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. Otherwise a query as $\exists xy \Box (R(a, x) \wedge R(x, y))$ with $\mathcal{T} = \{B \sqsubseteq \exists R, \exists R^- \sqsubseteq \exists R\}$, $R \in \mathbf{N}_{\text{RR}}$ and $\mathcal{A}_i = \{B(a)\}$ would be entailed from \mathcal{K} but not from $\mathcal{K}_{\mathcal{R}}$ with $\mathbf{N}_{\text{RR}} = \emptyset$. Indeed, in this case $\mathcal{R} = \{R(a, x_{aR})\}$, so x_{aR} may have a different R -successor in each interpretation of a model of $\mathcal{K}_{\mathcal{R}}$ and y cannot be mapped to the same object at every time point.

It follows from Proposition 3 and the NP-completeness of TCQ answering in Case 1 that TCQ answering is NP-complete w.r.t. combined complexity with the lower bound coming from the atemporal case. The following theorem summarizes the known complexity results for the classical semantics.

	AR	IAR	brave	AR	IAR	brave
Case 1 ($\mathbf{N}_{RC} = \emptyset, \mathbf{N}_{RR} = \emptyset$)	coNP-c	in P	in P	Π_2^p -c	NP-c	NP-c
Case 2 ($\mathbf{N}_{RC} \neq \emptyset, \mathbf{N}_{RR} = \emptyset$)	coNP-c	in P	NP-c	Π_2^p -c	NP-c	NP-c
Case 3* ($\mathbf{N}_{RC} \neq \emptyset, \mathbf{N}_{RR} \neq \emptyset$)	coNP-c	in P	NP-c	Π_2^p -c	NP-c	NP-c

Fig. 1: Data [left] and combined [right] complexity of BTCQ entailment over DL-Lite \mathcal{R} TKBs under the different semantics. *: only with rigid specializations of rigid roles

Theorem 1. *If \mathcal{T} does not entail any role inclusion of the form $P_1 \sqsubseteq P_2$ with $P_1 := R_1|R_1^-, R_1 \in \mathbf{N}_R \setminus \mathbf{N}_{RR}$ and $P_2 := R_2|R_2^-, R_2 \in \mathbf{N}_{RR}$, then consistency checking is in P w.r.t. combined complexity and TCQ answering is in P w.r.t. data complexity, and NP-complete w.r.t. combined complexity.*

We now turn our attention to the inconsistency-tolerant semantics.

Theorem 2. *The results in Figure 1 hold.*

In what follows, we present the key ideas underlying Theorem 2. First note that verifying that a sequence of ABoxes $(\mathcal{A}'_i)_{0 \leq i \leq n}$ is a repair of \mathcal{K} can be done in P by checking that $\mathcal{A}'_i \subseteq \mathcal{A}_i$ for every i , that $(\mathcal{A}'_i)_{0 \leq i \leq n}$ is consistent, and that adding any other timed-assertion of \mathcal{K} renders it inconsistent.

AR upper bounds. We show that ϕ does not hold under AR semantics by guessing a repair of \mathcal{K} that does not entail ϕ .

IAR upper bounds. We compute the minimal inconsistent subsets of \mathcal{K} in P by checking the consistency of every timed-assertion and pair of timed-assertions, then answer the query over the TKB from which they have been removed. Indeed, if a timed-assertion (α, i) is inconsistent it cannot be in a repair, and if there exists a consistent (β, j) such that $\{(\alpha, i), (\beta, j)\}$ is inconsistent, (α, i) is not in the repairs that contain (β, j) . In the other direction, if (α, i) does not appear in some repair $(\mathcal{A}'_i)_{0 \leq i \leq n}$ of \mathcal{K} , since the repairs are maximal, $(\mathcal{A}'_i)_{0 \leq i \leq n} \cup \{(\alpha, i)\}$ is inconsistent so (α, i) is in some minimal inconsistent subset of \mathcal{K} .

AR and IAR lower bounds. Hardness results come from the atemporal case.

Combined complexity of brave. We show that ϕ holds under brave semantics by guessing a repair of \mathcal{K} that entails ϕ . Hardness comes from the atemporal case.

Data complexity of brave. The data complexity upper bound for brave CQ answering relies on the fact that the size of the minimal sets of assertions that support the query is bounded by the query size, which is not true in the temporal setting (e.g., consider $\phi = \Box A(a)$, which needs n assertions to be entailed). Moreover, while brave BCQ entailment is tractable in the atemporal setting, we show that if rigid concepts are allowed, brave BTCQ entailment is NP-hard.

Proposition 4. *If $\mathbf{N}_{RC} \neq \emptyset$, brave TCQ answering is NP-complete w.r.t. data complexity.*

Proof. We show the lower bound by reduction from SAT. Let $\varphi = C_1 \wedge \dots \wedge C_n$ be a CNF formula over variables x_1, \dots, x_m . We define the following problem of BTCQ entailment under brave semantics over TKB \mathcal{K} with concepts $T, F \in \mathbf{N}_{\text{RC}}$:

$$\begin{aligned} \mathcal{T} &= \{\exists \text{Pos} \sqsubseteq \text{Sat}, \exists \text{Neg} \sqsubseteq \text{Sat}, \exists \text{Pos}^- \sqsubseteq T, \exists \text{Neg}^- \sqsubseteq F, T \sqsubseteq \neg F\} \\ \mathcal{A}_i &= \{\text{Pos}(c, x_j) \mid x_j \in C_i\} \cup \{\text{Neg}(c, x_j) \mid \neg x_j \in C_i\} \text{ for } 1 \leq i \leq n \end{aligned}$$

Let $\phi = \Box^- \text{Sat}(c)$. We show that φ is satisfiable iff $\mathcal{K}, n \models_{\text{brave}} \phi$. Indeed, since T and F are rigid, a repair of \mathcal{K} is such that each x_j has either only *Pos* or *Neg* incoming edges in the whole TKB. Thus, each repair defines a valuation such that x_j is true if x_j has no incoming *Neg*-edge, and false otherwise. A repair of \mathcal{K} that entails ϕ , i.e., that is such that c has an outgoing edge in every ABox, corresponds thus to a valuation of the x_j that satisfies every clause C_i .

It remains to show that in **Case 1**, brave TCQ answering is in P. We describe a method for brave BTCQ entailment when $\mathbf{N}_{\text{RC}} = \mathbf{N}_{\text{RR}} = \emptyset$ that 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 $L(\phi)$ of *leaves* of ϕ , that is, the set of all BCQs in ϕ , and the set $F(\phi)$ of *subformulas* of ϕ . In what follows, we identify the BCQs of $L(\phi)$ and the BTCQs of $F(\phi)$ with their *propositional abstractions*: if we write that a KB or a TKB entails some elements of $L(\phi)$ or $F(\phi)$, we consider them as BCQs or BTCQs, and if we write that some elements of $L(\phi)$ or $F(\phi)$ entail others, we consider the elements of $L(\phi)$ as propositional variables and those of $F(\phi)$ as propositional LTL formulas built over these variables.

Definition 6. A justification structure J for ϕ in \mathcal{K} is a set of tuples of the form $(i, L_{\text{now}}, F_{\text{now}}, F_{\text{prev}}, F_{\text{next}})$, where $0 \leq i \leq n$, $L_{\text{now}} \subseteq L(\phi)$, $F_{\text{now}} \subseteq F(\phi)$, $F_{\text{prev}} \subseteq F(\phi)$, and $F_{\text{next}} \subseteq F(\phi)$.

Note that the size of a justification structure for ϕ in \mathcal{K} 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. $\langle \mathcal{T}, \mathcal{A}_i \rangle \models_{\text{brave}} \bigwedge_{q \in L_{\text{now}}} q$
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 $\psi \in L(\phi)$, 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 $\bigwedge_{q \in L_{\text{now}}} q \wedge \bigcirc^-(\bigwedge_{\chi \in F_{\text{prev}}} \chi) \wedge \bigcirc(\bigwedge_{\chi \in F_{\text{next}}} \chi) \models \psi$, 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 $\diamond \psi \in F_{\text{now}}$, then either $\psi \in F_{\text{now}}$ or $\diamond \psi \in F_{\text{next}}$
 - if $\diamond^- \psi \in F_{\text{now}}$, then either $\psi \in F_{\text{now}}$ or $\diamond^- \psi \in F_{\text{prev}}$
 - if $\psi' \text{U} \psi \in F_{\text{now}}$, then either $\psi \in F_{\text{now}}$ or $\psi' \in F_{\text{now}}$ and $\psi' \text{U} \psi \in F_{\text{next}}$
 - if $\psi' \text{S} \psi \in F_{\text{now}}$, then either $\psi \in F_{\text{now}}$ or $\psi' \in F_{\text{now}}$ and $\psi' \text{S} \psi \in F_{\text{prev}}$

8. If $i = n$,
 - $\forall \psi \in F(\phi)$ of the form $\bullet\varphi, \psi \in F_{\text{now}}$
 - $\forall \psi \in F(\phi)$ of the form $\circ\varphi, \psi \notin F_{\text{now}}$
 - $\forall \psi \in F(\phi)$ of the form $\diamond\varphi, \square\varphi, \varphi'U\varphi, \psi \in F_{\text{now}}$ iff $\varphi \in F_{\text{now}}$
9. If $i = 0$,
 - $\forall \psi \in F(\phi)$ of the form $\bullet^-\varphi, \psi \in F_{\text{now}}$
 - $\forall \psi \in F(\phi)$ of the form $\circ^-\varphi, \psi \notin F_{\text{now}}$
 - $\forall \psi \in F(\phi)$ of the form $\diamond^-\varphi, \square^-\varphi, \varphi'S\varphi, \psi \in F_{\text{now}}$ iff $\varphi \in F_{\text{now}}$

We give the intuition behind the elements of the tuples fulfilling these conditions. The first element i is the time point considered, L_{now} is a set of BCQs whose conjunction is entailed under brave semantics by $\langle \mathcal{T}, \mathcal{A}_i \rangle$ (Condition 1), and F_{now} is the set of formulas that can be entailed together with L_{now} , depending on what is entailed in the previous and next time points, this information being stored in F_{prev} and F_{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_{now} is precisely the set of BCQs contained in F_{now} and Condition 5 that F_{now} is maximal in the sense that it contains its consequences. Condition 7 enforces that $F_{\text{now}}, F_{\text{prev}}$ and F_{next} respect the semantics of LTL operators and Conditions 8 and 9 enforce this semantics at the ends of the finite sequence.

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

The complexity of brave TCQ answering follows from the characterization of brave BTCQ entailment with justification structures.

Proposition 5. *In Case 1, brave TCQ answering is in P w.r.t. data complexity.*

Proof. We start with a justification structure J for ϕ in \mathcal{K} that contains all possible tuples and remove the unjustified tuples 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, then from n to 0 eliminating those which do not satisfy Condition 2. We then check whether the resulting justification structure contains a tuple $(p, L_{\text{now}}, F_{\text{now}}, F_{\text{prev}}, F_{\text{next}})$ such that $\phi \in F_{\text{now}}$. Since the size of J is linear in n , this process requires at most quadratically many steps. Verifying that a given tuple is justified is in P w.r.t. data complexity (checking Condition 3 or Condition 2 is linear in n and only the brave entailment of a BCQ from a DL-Lite $_{\mathcal{R}}$ KB for Condition 1 depends on the size of the ABox), so the complete procedure runs in P w.r.t. data complexity.

5 Conclusion and Future Work

We extended the AR, IAR and brave semantics to the setting of temporal query answering in description logics. We first showed that in the case where rigid predicates are not allowed, TCQ answering under IAR semantics can be achieved by combining algorithms developed for TCQ answering under the classical semantics with algorithms for CQ answering under IAR semantics over atemporal KBs. We also showed that in some cases, the same applies to AR semantics and that in any case, this method provides a sound approximation of AR answers. Since this is not true for brave semantics and we believe that this semantics can be relevant, for instance in the application of situation recognition, it would be useful to characterize the queries for which this method would be correct. Indeed, for many pairs of a TBox and query, the minimal subsets of the TKB such that the query can be mapped into them cannot be inconsistent (e.g., if pairs of predicates that may be needed at the same time point do not appear in any NI entailed by the TBox. If $\mathcal{T} = \{A \sqsubseteq \neg C, B \sqsubseteq \neg C\}$ and $\phi = \exists x A(x) \wedge \diamond(\exists x B(x) \wedge \circ(\exists x C(x)))$, for ϕ being 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 $i + 1 \geq p$. Thus, there cannot be a conflict between the C and the A or B timed-assertions used to satisfy the different CQs).

Our second contribution is a complexity analysis of the three semantics for DL-Lite \mathcal{R} , depending on which predicates are allowed to be rigid. Encouragingly, only brave semantics in the cases with rigid predicates has a higher data complexity than in the atemporal case. In the other cases handling of inconsistencies comes at no extra cost for temporal reasoning in terms of computational complexity. These results rise hope for feasibility of making ontology-based applications in temporal settings resilient against noise in the data.

We also showed that for the classical semantics, rigid predicates can be handled by adding a set of assertions to each ABox of the TKB, proving that disallowing negations in the query makes the combined complexity of TCQ answering drop from PSPACE to NP. However, our approach that adds the set of assertions \mathcal{R} to every ABox to reduce Cases 2 or 3 to Case 1 works only for the classical semantics. Now, practical algorithms still remain to be found for inconsistency-tolerant temporal query answering with rigid predicates.

References

1. Artale, A., Kontchakov, R., Kovtunova, A., Ryzhikov, V., Wolter, F., Zakharyashev, M.: First-order rewritability of temporal ontology-mediated queries. In: Proceedings of IJCAI (2015)
2. Artale, A., Kontchakov, R., Ryzhikov, V., Zakharyashev, M.: A cookbook for temporal conceptual data modelling with description logics. *ACM Trans. Comput. Log.* 15(3), 25:1–25:50 (2014)
3. Baader, F., Bauer, A., Baumgartner, P., Cregan, A., Gabaldon, A., Ji, K., Lee, K., Rajaratnam, D., Schwitter, R.: A novel architecture for situation awareness systems. In: Proceedings of TABLEAUX (2009)
4. Baader, F., Borgwardt, S., Lippmann, M.: Temporalizing ontology-based data access. In: Proceedings of CADE (2013)

5. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press (2003)
6. Bertossi, L.E.: Database Repairing and Consistent Query Answering. Synthesis Lectures on Data Management, Morgan & Claypool Publishers (2011)
7. Bienvenu, M., Bourgaux, C.: Inconsistency-tolerant querying of description logic knowledge bases. In: Reasoning Web, Tutorial Lectures. pp. 156–202 (2016)
8. Bienvenu, M., Bourgaux, C., Goasdoué, F.: Querying inconsistent description logic knowledge bases under preferred repair semantics. In: Proceedings of AAAI (2014)
9. Bienvenu, M., Rosati, R.: Tractable approximations of consistent query answering for robust ontology-based data access. In: Proceedings of IJCAI (2013)
10. Borgwardt, S., Lippmann, M., Thost, V.: Temporal query answering in the description logic DL-Lite. In: Proceedings of FroCoS (2013)
11. Borgwardt, S., Lippmann, M., Thost, V.: Temporalizing rewritable query languages over knowledge bases. Journal Web Sem. 33, 50–70 (2015)
12. Borgwardt, S., Thost, V.: Temporal query answering in DL-Lite with negation. In: Proceedings of GCAI (2015)
13. Bourgaux, C., Turhan, A.Y.: Temporal query answering in DL-Lite over inconsistent data (2017), LTCS-Report 17-06, Chair for Automata Theory, TU Dresden. See <https://lat.inf.tu-dresden.de/research/reports.html>
14. Calbimonte, J., Jeung, H., Corcho, Ó., Aberer, K.: Enabling query technologies for the semantic sensor web. Int. J. Semantic Web Inf. Syst. 8(1), 43–63 (2012)
15. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rodriguez-Muro, M., Rosati, R.: Ontologies and databases: The DL-Lite approach. In: Reasoning Web, Tutorial Lectures. pp. 255–356 (2009)
16. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The DL-Lite family. Journal of Automated Reasoning (JAR) 39(3), 385–429 (2007)
17. Chomicki, J., Toman, D., Böhlen, M.H.: Querying ATSQL databases with temporal logic. ACM Trans. Database Syst. 26(2), 145–178 (2001)
18. Endsley, M.R.: Toward a theory of situation awareness in dynamic systems. Human Factors 37(1), 32–64 (1995)
19. Gutiérrez-Basulto, V., Klarman, S.: Towards a unifying approach to representing and querying temporal data in description logics. In: Proceedings of RR (2012)
20. Klarman, S., Meyer, T.: Querying temporal databases via OWL 2 QL. In: Proceedings of RR (2014)
21. Lembo, D., Lenzerini, M., Rosati, R., Ruzzi, M., Savo, D.F.: Inconsistency-tolerant semantics for description logics. In: Proceedings of RR (2010)
22. Lembo, D., Lenzerini, M., Rosati, R., Ruzzi, M., Savo, D.F.: Inconsistency-tolerant query answering in ontology-based data access. Journal Web Sem. 33, 3–29 (2015)
23. Lutz, C., Wolter, F., Zakharyashev, M.: Temporal description logics: A survey. In: Proceedings of TIME (2008)
24. Motik, B., Cuenca Grau, B., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C.: OWL 2 Web Ontology Language profiles. W3C Recommendation (11 December 2012), available at <http://www.w3.org/TR/owl2-profiles/>
25. Özçep, Ö.L., Möller, R.: Ontology based data access on temporal and streaming data. In: Reasoning Web, Tutorial Lectures. pp. 279–312 (2014)
26. Pnueli, A.: The temporal logic of programs. In: Proceedings of FOCS (1977)
27. Tsalapati, E., Stoilos, G., Stamou, G.B., Koletsos, G.: Efficient query answering over expressive inconsistent description logics. In: Proceedings of IJCAI (2016)