

Temporal Conjunctive Queries in Expressive Description Logics with Transitive Roles^{*}

Franz Baader, Stefan Borgwardt, and Marcel Lippmann

Theoretical Computer Science, TU Dresden, Germany
firstname.lastname@tu-dresden.de

Abstract. In Ontology-Based Data Access (OBDA), user queries are evaluated over a set of facts under the open world assumption, while taking into account background knowledge given in the form of a Description Logic (DL) ontology. In order to deal with dynamically changing data sources, temporal conjunctive queries (TCQs) have recently been proposed as a useful extension of OBDA to support the processing of temporal information. We extend the existing complexity analysis of TCQ entailment to very expressive DLs underlying the OWL 2 standard, and in contrast to previous work also allow for queries containing transitive roles.

1 Introduction

Given a (man-made or natural) dynamical system that changes its states over time, it is sometimes useful to monitor the behavior of the system in order to detect and then react to critical situations [2]. To achieve this, one can monitor the running system using sensors (e.g., heart rate and blood pressure sensors for a patient) and store the (possibly aggregated and preprocessed) values in a database. Critical situations (such as “blood pressure too high”) can then be described by database queries, and detecting them can be realized through query answering. However, such a pure database solution is unsatisfactory for several reasons. First, one cannot assume that the sensors provide a complete description of the current state of the system, which clashes with the closed world assumption used by database systems. Second, though one usually does not have a complete specification of the system’s behavior, one may have some background knowledge restricting the possible states of the system, which can help to detect more situations.

These two problems are addressed by so-called ontology-based data access (OBDA) [14,27], where (i) the preprocessed and aggregated data are stored in a Description Logic (DL) ABox, which is interpreted with open world assumption, and (ii) the background knowledge is represented in a TBox (ontology) expressed in an appropriate DL. DLs [5] can be used to formalize knowledge using concepts, which represent sets of elements of an application domain, and roles, which describe binary relations between elements. For example, the concept *Patient* can

^{*} This work was partially supported by the DFG in the CRC 912 (HAEC).

be used to model the set of all patients in a hospital, while `isTreatedWith` represents a relationship between patients and treatments. Concept constructors can then be used to build complex concepts out of atomic concepts and roles. For example, `Patient ⊓ ∃isTreatedWith.Antibiotics` describes patients treated with antibiotics. In the TBox, one can state subconcept-superconcept relationships, such as `∃isTreatedWith.Antibiotics ⊑ ∃finding.BacterialInfection`, which says that antibiotics treatment is given only if there is a bacterial infection. In the ABox, one can state specific facts about individuals, such as `isTreatedWith(BOB, PENICILLIN)`.

When monitoring a dynamical system, the situation to be recognized may also depend on states of the system at different points in time (such as “fluctuating heart rate”). For this reason, OBDA was extended to the temporal case in [1, 4]. In [4] the complexity of answering temporal conjunctive queries (TCQs) w.r.t. TBoxes was investigated for TBoxes expressed in DLs between \mathcal{ALC} and \mathcal{SHQ} . The results are concerned both with data complexity (which is measured only in the size of the data) and with combined complexity (which additionally takes the size of the query and the TBox into account). In addition, the paper considers rigid concepts and roles, whose interpretations must not change over time.

We extend the results of [4] in two directions. First, while being quite expressive, \mathcal{SHQ} does not contain the constructors nominals and inverse roles, which are quite useful in many applications. Here, we also consider logics that have these two constructors. However, the main difference is that, though \mathcal{SHQ} can express transitivity of roles and sub-role relationships, transitive roles and roles with transitive subroles must not occur in queries in [4]. In the present paper, we dispense with this restriction, which unfortunately leads to a dramatic increase in complexity that reflects the results for standard (atemporal) queries (see [15, 21] and Table 2).

As an example that illustrates the benefit of transitive roles in queries, assume that we want to recognize patients who have previously had myocarditis, i.e., an inflammation of the heart muscle. This can be expressed using the TCQ

$$\text{Patient}(x) \wedge \bigcirc^- \diamond^- (\exists y, z. \text{partOf}(y, x) \wedge \text{Heart}(y) \wedge \text{partOf}(z, y) \wedge \text{Muscle}(z) \wedge \text{Inflamed}(z))$$

This query is looking for a patient that, at some past time point, had (as part) a heart that itself had as part a muscle that was inflamed. In this example, we assume that the role `partOf` is transitive and rigid. Transitivity implies that the inflamed muscle was also part of the patient and rigidity ensures that the heart is not part of different patients at different points in time. In addition, we assume that `Heart` and `Muscle` are rigid (hearts and muscles stay hearts and muscles over time), but `Patient` and `Inflamed` are non-rigid (the muscle may, e.g., cease to be inflamed and the patient may be discharged).

In the next section, we introduce the DLs investigated in this paper, as well as TCQs and their semantics. We also give an overview over the already known and the new complexity results (see Table 2). Section 3 investigates the complexity of answering certain atemporal queries in a fine-grained way. The reason is that,

Table 1. Syntax and Semantics of DLs

	syntax	semantics
inverse role	r^-	$\{(e, d) \mid (d, e) \in r^{\mathcal{I}}\}$
negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
existential restriction	$\exists r.C$	$\{d \in \Delta^{\mathcal{I}} \mid \text{there is } e \in C^{\mathcal{I}} \text{ with } (d, e) \in r^{\mathcal{I}}\}$
nominal	$\{a\}$	$\{a^{\mathcal{I}}\}$
at-most restriction	$\leq n r.C$	$\{d \in \Delta^{\mathcal{I}} \mid \#\{e \in C^{\mathcal{I}} \mid (d, e) \in r^{\mathcal{I}}\} \leq n\}$
general concept inclusion	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
concept assertion	$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
role assertion	$r(a, b)$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$
role inclusion	$r \sqsubseteq s$	$r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$
transitivity axiom	$\text{trans}(r)$	$r^{\mathcal{I}} = (r^{\mathcal{I}})^+$

similar to [4], we split the task of answering TCQs into propositional temporal reasoning on the one hand, and answering atemporal queries on the other hand. In Section 4, we then determine the combined complexity of answering TCQs whereas in Section 5 we deal with the data complexity. Full proofs of our results can be found in an accompanying technical report [3].

2 Preliminaries

In this section, we recall the basic notions of DLs and TCQs. Throughout the paper, let \mathbf{N}_C , \mathbf{N}_R , and \mathbf{N}_I be non-empty, pairwise disjoint sets of *concept names*, *role names*, and *individual names*, respectively.

Definition 1 (Syntax of DLs). *A role is either a role name $r \in \mathbf{N}_R$ or an inverse role r^- . The set of concepts is inductively defined starting from concept names $A \in \mathbf{N}_C$ using the constructors in the second part of Table 1, where r, s are roles, $a, b \in \mathbf{N}_I$, $n \in \mathbb{N}$, and C, D are concepts. The third part of Table 1 shows how axioms are defined. A TBox is a finite set of general concept inclusions (GCIs), an RBox is a finite set of role inclusions and transitivity axioms, and an ABox is a finite set of concept and role assertions. A knowledge base (KB) $\mathcal{K} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$ consists of an ABox \mathcal{A} , a TBox \mathcal{T} , and an RBox \mathcal{R} .*

In the DL \mathcal{ALC} , negation, conjunction, and existential restriction are the only allowed constructors. Also, no inverse roles, role inclusions and transitivity axioms are allowed in \mathcal{ALC} . Additional letters denote different concept constructors or types of axioms: \mathcal{I} means inverse roles, \mathcal{O} means nominals, \mathcal{Q} means at-most restrictions, and \mathcal{H} means role inclusions. For example, the DL \mathcal{ALCHI} extends \mathcal{ALC} by role inclusions and inverse roles. The extension of \mathcal{ALC} with transitivity axioms is denoted by \mathcal{S} . Hence, the DL allowing for all the constructors and types of axioms introduced here is called \mathcal{SHOIQ} . We sometimes write \mathcal{L} -concept (\mathcal{L} -KB, ...) for some DL \mathcal{L} to make clear which DL is used.

Definition 2 (Semantics of DLs). An interpretation is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a non-empty domain, and $\cdot^{\mathcal{I}}$ is a mapping assigning a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ to every $A \in \mathbf{N}_{\mathbf{C}}$, a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ to every $r \in \mathbf{N}_{\mathbf{R}}$, and a domain element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ to every $a \in \mathbf{N}_{\mathbf{I}}$, such that $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ for all $a, b \in \mathbf{N}_{\mathbf{I}}$ with $a \neq b$ (unique name assumption (UNA)). This function is extended to roles and concepts as shown in Table 1, where $\#S$ denotes the cardinality of the set S .

Moreover, \mathcal{I} is a model of the axiom α (written $\mathcal{I} \models \alpha$) if the condition in the third part of Table 1 is satisfied, where \cdot^+ denotes the transitive closure. Furthermore, \mathcal{I} is a model of a set of axioms \mathcal{X} (written $\mathcal{I} \models \mathcal{X}$) if it is a model of all axioms $\alpha \in \mathcal{X}$, and \mathcal{I} is a model of a KB $\mathcal{K} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$ (written $\mathcal{I} \models \mathcal{K}$) if it is a model of \mathcal{A} , \mathcal{T} , and \mathcal{R} . We call \mathcal{K} consistent if it has a model.

For an RBox \mathcal{R} , we call a role name $r \in \mathbf{N}_{\mathbf{R}}$ *transitive* (w.r.t. \mathcal{R}) if every model of \mathcal{R} is a model of $\text{trans}(r)$. Moreover, r is a *subrole* of a role name $s \in \mathbf{N}_{\mathbf{R}}$ (w.r.t. \mathcal{R}) if every model of \mathcal{R} is a model of $r \sqsubseteq s$. Finally, r is *simple* w.r.t. \mathcal{R} if it has no transitive subrole. Deciding whether $r \in \mathbf{N}_{\mathbf{R}}$ is simple can be done in time polynomial in the size of \mathcal{R} by a simple reachability test. Unfortunately, the problem of deciding whether a given *SHQ*-KB $\mathcal{K} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$ is consistent is undecidable in general [19]. To regain decidability, we need to make the following syntactic restriction: if $\leq n r.C$ occurs in \mathcal{K} , then r must be simple w.r.t. \mathcal{R} .

To better separate the influence the ABox has on the complexity of reasoning, we assume in the following that assertions use only *names* that must also occur in the TBox or the RBox. One can still simulate a complex concept assertion $C(a)$ using $A(a)$ and $A \equiv C$, where the latter stands for $A \sqsubseteq C$ and $C \sqsubseteq A$.

Before we can define temporal queries, we need to lift the notions of knowledge bases and interpretations to a temporal setting. We assume that there are designated sets $\mathbf{N}_{\mathbf{RC}} \subseteq \mathbf{N}_{\mathbf{C}}$ of *rigid concept names* and $\mathbf{N}_{\mathbf{RR}} \subseteq \mathbf{N}_{\mathbf{R}}$ of *rigid role names*, whose interpretation does not change over time. All individual names are implicitly rigid. A concept or role name that is not rigid is called *flexible*.

Definition 3 (TKB). A tuple $\mathcal{K} = ((\mathcal{A}_i)_{0 \leq i \leq n}, \mathcal{T}, \mathcal{R})$, consisting of a finite sequence of ABoxes \mathcal{A}_i , a TBox \mathcal{T} , and an RBox \mathcal{R} , is called a temporal knowledge base (TKB). Let $\mathfrak{J} = (\mathcal{I}_i)_{i \geq 0}$ be an infinite sequence of interpretations $\mathcal{I}_i = (\Delta, \cdot^{\mathcal{I}_i})$ over a fixed domain Δ . Then \mathfrak{J} is a model of \mathcal{K} (written $\mathfrak{J} \models \mathcal{K}$) if

- $\mathcal{I}_i \models \mathcal{A}_i$ for all $i, 0 \leq i \leq n$,
- $\mathcal{I}_i \models \mathcal{T}$ and $\mathcal{I}_i \models \mathcal{R}$ for all $i \geq 0$, and
- \mathfrak{J} respects rigid names, i.e., $x^{\mathcal{I}_i} = x^{\mathcal{I}_j}$ for all $x \in \mathbf{N}_{\mathbf{I}} \cup \mathbf{N}_{\mathbf{RC}} \cup \mathbf{N}_{\mathbf{RR}}$ and $i, j \geq 0$.

We denote the set of all individual names occurring in a TKB \mathcal{K} by $\text{Ind}(\mathcal{K})$. TCQs are defined by combining conjunctive queries via the operators of LTL [4, 26].

Definition 4 (Syntax of TCQs). Let $\mathbf{N}_{\mathbf{V}}$ be a set of variables. A conjunctive query (CQ) is of the form $\exists y_1, \dots, y_m. \psi$, where $y_1, \dots, y_m \in \mathbf{N}_{\mathbf{V}}$ and ψ is a finite conjunction of atoms of the form $A(z_1)$ (concept atom), $r(z_1, z_2)$ (role atom), or $z_1 \approx z_2$ (equality atom), where $A \in \mathbf{N}_{\mathbf{C}}$, $r \in \mathbf{N}_{\mathbf{R}}$, and $z_1, z_2 \in \mathbf{N}_{\mathbf{V}} \cup \mathbf{N}_{\mathbf{I}}$.

Temporal conjunctive queries (TCQs) are built inductively from CQs, using the constructors $\neg \phi_1$ (negation), $\phi_1 \wedge \phi_2$ (conjunction), $\bigcirc \phi_1$ (next), $\bigcirc^- \phi_1$ (previous), $\phi_1 \cup \phi_2$ (until), and $\phi_1 \text{ S } \phi_2$ (since), where ϕ_1 and ϕ_2 are TCQs.

In contrast to [4], we allow non-simple roles to occur in CQs. A *union of conjunctive queries (UCQ)* is a disjunction of CQs, defined as $\phi_1 \vee \phi_2 := \neg(\neg\phi_1 \wedge \neg\phi_2)$. A *CQ-literal* is either a CQ or a negated CQ. We denote the set of individual names occurring in a TCQ ϕ by $\text{Ind}(\phi)$, the set of variables occurring in ϕ by $\text{Var}(\phi)$, and the set of free variables of ϕ by $\text{FVar}(\phi)$. If $\text{FVar}(\phi) = \emptyset$, we call ϕ *Boolean*. As in [4], we assume without loss of generality that all CQs are connected, i.e., all variables and individual names are related (transitively) by roles.

Definition 5 (Semantics of TCQs). *An interpretation $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$ is a model of a Boolean CQ ϕ (written $\mathcal{I} \models \phi$) if there is a homomorphism of ϕ into \mathcal{I} , which is a mapping $\pi: \text{Var}(\phi) \cup \text{Ind}(\phi) \rightarrow \Delta$ with $\pi(a) = a^{\mathcal{I}}$ for all $a \in \text{Ind}(\phi)$; $\pi(z) \in A^{\mathcal{I}}$ for all concept atoms $A(z)$ in ϕ ; $(\pi(z_1), \pi(z_2)) \in r^{\mathcal{I}}$ for all role atoms $r(z_1, z_2)$ in ϕ ; and $\pi(z_1) = \pi(z_2)$ for all equality atoms $z_1 \approx z_2$ in ϕ .*

An infinite sequence of interpretations $\mathfrak{I} = (\mathcal{I}_i)_{i \geq 0}$ over a common domain Δ is a model of a Boolean TCQ ϕ at time point $i \geq 0$ iff $\mathfrak{I}, i \models \phi$ holds, where

$$\begin{array}{ll}
\mathfrak{I}, i \models \exists y_1, \dots, y_m. \psi & \text{iff } \mathcal{I}_i \models \exists y_1, \dots, y_m. \psi \\
\mathfrak{I}, i \models \neg\phi_1 & \text{iff } \mathfrak{I}, i \not\models \phi_1 \\
\mathfrak{I}, i \models \phi_1 \wedge \phi_2 & \text{iff } \mathfrak{I}, i \models \phi_1 \text{ and } \mathfrak{I}, i \models \phi_2 \\
\mathfrak{I}, i \models \bigcirc\phi_1 & \text{iff } \mathfrak{I}, i+1 \models \phi_1 \\
\mathfrak{I}, i \models \bigcirc^-\phi_1 & \text{iff } i > 0 \text{ and } \mathfrak{I}, i-1 \models \phi_1 \\
\mathfrak{I}, i \models \phi_1 \bigcup \phi_2 & \text{iff there is } k \geq i \text{ with } \mathfrak{I}, k \models \phi_2 \\
& \text{and } \mathfrak{I}, j \models \phi_1 \text{ for all } j, i \leq j < k \\
\mathfrak{I}, i \models \phi_1 \text{S} \phi_2 & \text{iff there is } k, 0 \leq k \leq i \text{ with } \mathfrak{I}, k \models \phi_2 \\
& \text{and } \mathfrak{I}, j \models \phi_1 \text{ for all } j, k < j \leq i
\end{array}$$

*Given a TKB $\mathcal{K} = ((\mathcal{A}_i)_{0 \leq i \leq n}, \mathcal{T}, \mathcal{R})$, we say that \mathfrak{I} is a model of ϕ w.r.t. \mathcal{K} if $\mathfrak{I} \models \mathcal{K}$ and $\mathfrak{I}, n \models \phi$. We call ϕ *satisfiable w.r.t. \mathcal{K}* if it has a model w.r.t. \mathcal{K} , and it is *entailed by \mathcal{K}* (written $\mathcal{K} \models \phi$) if every model \mathfrak{I} of \mathcal{K} satisfies $\mathfrak{I}, n \models \phi$.*

For a TCQ ϕ , $\mathbf{a}: \text{FVar}(\phi) \rightarrow \text{Ind}(\mathcal{K})$ is a certain answer to ϕ w.r.t. \mathcal{K} if $\mathcal{K} \models \mathbf{a}(\phi)$, where $\mathbf{a}(\phi)$ is obtained from ϕ by replacing the free variables using \mathbf{a} .

As usual [4], in the following we consider only the TCQ entailment problem, which can be used to compute all certain answers. For this purpose, we analyze the satisfiability problem, which has the same complexity as non-entailment. We examine both the *combined complexity* where the whole TKB and the TCQ are considered as the input, and the *data complexity*, where TBox, RBox, and TCQ are fixed, i.e., the complexity is measured only w.r.t. the sequence of ABoxes.

In the remainder of this section, we recall the basic approach from [4] to decide satisfiability by splitting it into two separate satisfiability problems, one for the temporal component and one for the DL component. In the following, let $\mathcal{K} = ((\mathcal{A}_i)_{0 \leq i \leq n}, \mathcal{T}, \mathcal{R})$ be a TKB and ϕ be a TCQ to be checked for satisfiability. The *propositional abstraction* ϕ^p of ϕ is the propositional LTL-formula obtained from ϕ by replacing all CQs by propositional variables. We assume that $\alpha_1, \dots, \alpha_m$ are the CQs occurring in ϕ , and that each α_i is replaced by the propositional variable p_i , $1 \leq i \leq m$. We now consider a set $\mathcal{S} = \{X_1, \dots, X_k\} \subseteq 2^{\{p_1, \dots, p_m\}}$, which intuitively specifies the worlds that are

Table 2. Summary of known and new complexity results for TCQ entailment, where contributions of this paper are highlighted in boldface. Settings: (i) no rigid names are allowed, (ii) only rigid concept names are allowed, and (iii) arbitrary rigid names are allowed.

	data complexity			combined complexity		
	(i)	(ii)	(iii)	(i)	(ii)	(iii)
<i>ALC</i> – <i>ALCHQ</i> [4]	co-NP	co-NP	\leq EXP	EXP	co-NEXP	2-EXP
<i>ALCO</i> – <i>ALCHOQ/ALCHOI</i>	co-NP	co-NP	\leqEXP	\geq co-NEXP	?	2-EXP
<i>S</i> – <i>SQ</i>	co-NP	co-NP	\leqEXP	\geq co-NEXP	?	2-EXP
<i>SO</i> – <i>SOQ</i>	\geq co-NP	?	\leq EXP	\geq co-NEXP	?	2-EXP
<i>SH/ALCI</i> – <i>SHIQ</i>	co-NP	co-NP	\leqEXP	2-EXP	2-EXP	2-EXP
<i>SHO</i> – <i>SHOQ/SHOI</i>	\geq co-NP	?	\leq EXP	2-EXP	2-EXP	2-EXP
<i>ALCOIQ</i> – <i>ALCHOIQ</i>	\geq co-NP	?	decidable	\geq co-2-NEXP	?	decidable
<i>SOIQ</i> – <i>SHOIQ</i>	\geq co-NP	?	?	\geq co-2-NEXP	?	?

allowed to occur in an LTL-structure satisfying ϕ^P at time point n , and a mapping $\iota: \{0, \dots, n\} \rightarrow \{1, \dots, k\}$, which assigns a world $X_{\iota(i)}$ to each input ABox \mathcal{A}_i .

Definition 6 (t-satisfiability). *The LTL-formula ϕ^P is t-satisfiable w.r.t. \mathcal{S} and ι if there exists an LTL-structure $\mathfrak{J} = (w_i)_{i \geq 0}$ such that $\mathfrak{J}, n \models \phi^P$, $w_i \in \mathcal{S}$ for all $i \geq 0$, and $w_i = X_{\iota(i)}$ for all i , $0 \leq i \leq n$.*

However, finding \mathcal{S} and ι and then testing t-satisfiability is not sufficient for checking whether ϕ has a model w.r.t. \mathcal{K} . We must also check whether \mathcal{S} can indeed be induced by some sequence of interpretations that is a model of \mathcal{K} , in the following sense.

Definition 7 (r-satisfiability). *The set \mathcal{S} is r-satisfiable w.r.t. ι and \mathcal{K} if there exist interpretations $\mathcal{J}_1, \dots, \mathcal{J}_k, \mathcal{I}_0, \dots, \mathcal{I}_n$ that share the same domain, respect rigid names, are models of \mathcal{T} and \mathcal{R} , and additionally each \mathcal{J}_i , $1 \leq i \leq k$, is a model of $\chi_i := \bigwedge_{p_j \in X_i} \alpha_j \wedge \bigwedge_{p_j \notin X_i} \neg \alpha_j$, and each \mathcal{I}_i , $0 \leq i \leq n$, is a model of \mathcal{A}_i and $\chi_{\iota(i)}$.*

The following was shown in [4] for *SHQ*, but is actually independent of any specific DL.

Proposition 8. *ϕ is satisfiable w.r.t. \mathcal{K} iff there are a set \mathcal{S} and a mapping ι such that \mathcal{S} is r-satisfiable w.r.t. ι and \mathcal{K} , and ϕ^P is t-satisfiable w.r.t. \mathcal{S} and ι .*

The complexity of the t-satisfiability problem is obviously also DL-agnostic, and hence we can reuse another result from [4].

Proposition 9. *Deciding t-satisfiability of ϕ^P w.r.t. \mathcal{S} and ι can be done in EXP w.r.t. combined complexity, and in P w.r.t. data complexity.*

Table 2 gives an overview over all known complexity results for TCQ entailment. We distinguish the cases that (i) no rigid names are allowed ($N_{RC} = N_{RR} = \emptyset$); (ii) only rigid concept names are allowed, but no rigid role names ($N_{RR} = \emptyset$); and (iii) arbitrary rigid names are allowed. The first row of the table contains the known results for *ALC/ALCHQ* [4]¹, and in this paper we derive the upper

¹ Actually, that paper considers *SHQ*, but restricts the roles in CQs to be simple.

bounds for cases (ii) and (iii) marked in bold font. Unfortunately, we leave open the precise data complexity for case (iii), as was the case in [4]. A question mark indicates that the precise complexity is unknown even for the atemporal CQ entailment problem. For SHOIQ , it is not even known whether this problem is decidable, while for $\mathit{ALCHOIQ}$ it is only known to be decidable, but no better upper bound has been found so far [25, 28]. The shown lower bounds follow from the complexity of satisfiability of ALC -LTL formulae [4, 6] and the complexity of atemporal CQ entailment. More precisely, the latter problem is CO-NP-hard in data complexity already for $\mathit{AL\mathcal{E}}$ [29]. Under combined complexity, it is CO-NEXP-hard for ALCO [23] and S [15], 2-EXP-hard for SH [15] and ALCI [21], and CO-2-NEXP-hard for ALCOIQ [18].

3 Atemporal Queries in SHIQ , SHOQ , and SHOI

Since our results about TCQ entailment are based on reductions to conjunctions of CQ-literals, we first analyze in more detail the case of such atemporal queries. In a nutshell, we reduce the satisfiability of such a conjunction to UCQ non-entailment and exploit existing algorithms for this problem. We consider only the logics SHIQ , SHOQ , and SHOI that have the *quasi-forest model property* [12], which means that every consistent KB formulated in one of these logics has a model that basically consists of several tree-shaped structures whose roots are arbitrarily interconnected by roles (disregarding role connections due to nominals or transitive roles).

To show the results in the following sections, however, we need to conduct a more fine-grained analysis of the complexity of the atemporal query entailment algorithms. The main insight is that, while UCQ entailment in SHIQ , SHOQ , and SHOI is in 2-EXP w.r.t. combined complexity, the *number* of CQs in the UCQ only has an exponential influence on the complexity of this decision problem. Likewise, for data complexity, assuming that the number of CQs in the UCQ is linear instead of constant usually has no influence on the complexity. Unfortunately, to the best of our knowledge, the precise data complexity of UCQ entailment is known only for SHIQ , ALCHOQ , and ALCHOI , while for SHOQ and SHOI it is still open [25].

In the following, we consider the *size* of a CQ ψ (written $|\psi|$) to be the number of symbols in ψ , ignoring constant expressions like ‘(’ and ‘^’, considering each name and variable to be of size 1, and further ignoring the prefix $\exists y_1, \dots, y_m$ since these variables also occur in the atoms of ψ . For example, $\exists x, y. r(x, y) \wedge A(x)$ has size 5. We could also assume that each name or variable is represented by a binary string denoting its name, and hence of size logarithmic in the size of ψ , but this would not affect our complexity results. Similarly, the size of a knowledge base is computed by ignoring the concept constructors, and hence considers only the number of occurrences of names in the axioms.

Lemma 10. *Let $\psi = \rho_1 \wedge \dots \wedge \rho_\ell \wedge \neg\sigma_1 \dots \wedge \neg\sigma_o$ be a Boolean conjunction of CQ-literals, $\mathcal{K} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$ be a KB formulated in SHIQ , SHOQ , or SHOI ,*

and $\|\psi\| := \max\{|\rho_1|, \dots, |\rho_\ell|, |\sigma_1|, \dots, |\sigma_o|\}$. Then the satisfiability of ψ w.r.t. \mathcal{K} can be decided by a deterministic algorithm in time bounded by $2^{p(\ell, o, |\mathcal{K}|)^{p'(\|\psi\|)}}$, for two polynomials p and p' .

In the case of *SHIQ*, *ALCHOQ*, or *ALCHOI*, if \mathcal{T} , \mathcal{R} , and $\|\psi\|$ are fixed, then satisfiability of ψ w.r.t. \mathcal{K} can be decided by a nondeterministic algorithm in time bounded by $p(\ell, o, |\mathcal{A}|)$ for some polynomial p .

Proof. As in [4], we reduce the decision whether ψ has a model w.r.t. \mathcal{K} to a UCQ non-entailment problem. We instantiate the positive CQs ρ_1, \dots, ρ_ℓ by omitting the existential quantifiers and replacing all variables by fresh individual names. The set \mathcal{A}_ρ of all resulting assertions can be viewed as an additional ABox. To ensure that the UNA is satisfied, we additionally consider equivalence relations \approx on $\text{Ind}(\mathcal{A} \cup \mathcal{A}_\rho)$ with the additional restriction that no two names from $\text{Ind}(\mathcal{A})$ may be equivalent. We denote by \mathcal{A}_\approx the ABox resulting from \mathcal{A}_ρ by replacing each new individual name by a fixed representative of its equivalence class, where this representative is an element of $\text{Ind}(\mathcal{A})$ whenever possible. It can be shown as in [4] that ψ is satisfiable w.r.t. \mathcal{K} iff there is an equivalence relation \approx with

$$(\mathcal{A} \cup \mathcal{A}_\approx, \mathcal{T}, \mathcal{R}) \not\models \sigma_1 \vee \dots \vee \sigma_o. \quad (1)$$

Note that the number of equivalence relations \approx is exponential in the total number of variables in ρ_1, \dots, ρ_ℓ , which is bounded by $\ell \cdot \|\psi\|$, but each is of size polynomial in $\ell \cdot \|\psi\|$. Hence, one can either enumerate all such equivalences in time exponential in $\ell \cdot \|\psi\|$, or guess one of them in time polynomial in $\ell \cdot \|\psi\|$.

We now consider the case that \mathcal{K} is formulated in *SHIQ*. By [16, Lemma 23], for the non-entailment test (1), it suffices to find a so-called *extended knowledge base* $\mathcal{K}' = (\mathcal{A} \cup \mathcal{A}_\approx \cup \mathcal{A}', \mathcal{T} \cup \mathcal{T}', \mathcal{R})$, where \mathcal{A}' and \mathcal{T}' are formulated in *SHIQ*[□], i.e., *SHIQ* extended by role conjunctions, such that \mathcal{K}' is consistent. By [16, Lemma 20 and Definition 21], the size of each $(\mathcal{A}', \mathcal{T}')$ is bounded by $p(o \cdot (|\mathcal{K}| + \ell \cdot \|\psi\|))^{p(\|\psi\|)}$ for some polynomial p , where the term $\ell \cdot \|\psi\|$ represents the size of the additional ABox \mathcal{A}_\approx . The bound given in [16] is exponential in the total size of the UCQ, i.e., $o \cdot \|\psi\|$, but the exponential blowup comes only from the rewriting of each individual CQ σ_i . Moreover, all pairs $(\mathcal{A}', \mathcal{T}')$ can be enumerated in time bounded by $2^{p(o \cdot (|\mathcal{K}| + \ell \cdot \|\psi\|))^{p(\|\psi\|)}}$. It is important to note that the size of the longest role conjunction occurring in $(\mathcal{A}', \mathcal{T}')$ is bounded by a polynomial in $\|\psi\|$. Hence, by [16, Lemma 28], one can check the consistency of \mathcal{K}' in time $2^{p'(o \cdot (|\mathcal{K}| + \ell \cdot \|\psi\|))^{p'(\|\psi\|)}}$ for some polynomial p' . Thus, we can decide satisfiability of ψ w.r.t. \mathcal{K} by enumerating all relations \approx and extended KBs as above and testing each of them for consistency within the claimed time bound.

If \mathcal{T} , \mathcal{R} , and $\|\psi\|$ are fixed, then one can guess \approx in time polynomial in ℓ . Following the proof of [16, Theorem 35], one can also guess \mathcal{K}' in time $p(o \cdot (|\mathcal{A}| + \ell))$, and the following consistency test can be done in (deterministic) polynomial time in the size of the ABox $\mathcal{A} \cup \mathcal{A}_\approx \cup \mathcal{A}'$, which is polynomial in $o \cdot (|\mathcal{A}| + \ell)$. This establishes the second bound for the case of *SHIQ*.

The proof of the remaining cases can be found in the technical report. For *SHOQ*, we use algorithms developed in [17, 20]. For *SHOI*, we analyze the

automata-based construction from [12,13] based on fully enriched automata [9]. For \mathcal{ALCHOQ} and \mathcal{ALCHOI} under the assumption that \mathcal{T} , \mathcal{R} , and $\|\psi\|$ are fixed, we obtain the claimed results using a tableaux algorithm introduced in [24]. \square

4 Combined Complexity of TCQ Entailment

Let $\mathcal{K} = ((\mathcal{A}_i)_{0 \leq i \leq n}, \mathcal{T}, \mathcal{R})$ be a TKB, ϕ be a TCQ, and assume for now that a set $\mathcal{S} = \{X_1, \dots, X_k\} \subseteq 2^{\{p_1, \dots, p_m\}}$ and a mapping $\iota: \{0, \dots, n\} \rightarrow \{1, \dots, k\}$ are given. For our complexity results, we employ the *copying* technique from [4,6] for deciding whether \mathcal{S} is r-satisfiable w.r.t. ι and \mathcal{K} . The idea is to introduce enough copies of all flexible names in order to combine the separate satisfiability tests of Def. 7 into one big atemporal satisfiability test.

Formally, for all i , $1 \leq i \leq k+n+1$, and every *flexible* concept name A (every *flexible* role name r) occurring in \mathcal{T} or \mathcal{R} , we introduce a copy $A^{(i)}$ ($r^{(i)}$). We call $A^{(i)}$ ($r^{(i)}$) the i -th copy of A (r). The conjunctive query $\alpha^{(i)}$ (the axiom $\beta^{(i)}$) is obtained from a CQ α (an axiom β) by replacing every flexible name by its i -th copy. Similarly, for $1 \leq \ell \leq k$, the conjunction of CQ-literals $\chi_\ell^{(i)}$ is obtained from χ_ℓ (see Definition 7) by replacing each CQ α_j by $\alpha_j^{(i)}$. Finally, we define

$$\begin{aligned} \chi_{\mathcal{S},\iota} &:= \bigwedge_{1 \leq i \leq k} \chi_i^{(i)} \wedge \bigwedge_{0 \leq i \leq n} \left(\chi_{i^{(i)}}^{(k+i+1)} \wedge \bigwedge_{\alpha \in \mathcal{A}_i} \alpha^{(k+i+1)} \right), \\ \mathcal{T}_{\mathcal{S},\iota} &:= \{\beta^{(i)} \mid \beta \in \mathcal{T} \text{ and } 1 \leq i \leq k+n+1\}, \\ \mathcal{R}_{\mathcal{S},\iota} &:= \{\gamma^{(i)} \mid \gamma \in \mathcal{R} \text{ and } 1 \leq i \leq k+n+1\}. \end{aligned}$$

The following result, which reduces r-satisfiability to an atemporal satisfiability problem, was shown in [4] for \mathcal{SHQ} with simple roles in queries, but it remains valid in our setting since it does not depend on the DL under consideration.

Proposition 11. *The set \mathcal{S} is r-satisfiable w.r.t. ι and \mathcal{K} iff $\chi_{\mathcal{S},\iota}$ is satisfiable w.r.t. $(\mathcal{T}_{\mathcal{S},\iota}, \mathcal{R}_{\mathcal{S},\iota})$.*

Together with Lemma 10, this allows us to show our first complexity results.

Theorem 12. *Let \mathcal{L} be a DL that contains \mathcal{ALCI} or \mathcal{SH} and is contained in \mathcal{SHIQ} , \mathcal{SHOQ} , or \mathcal{SHOI} . Then TCQ entailment in \mathcal{L} is 2-EXP-complete w.r.t. combined complexity, and in EXP w.r.t. data complexity.*

Proof. The lower bound directly follows from 2-EXP-hardness of CQ entailment in \mathcal{SH} [15] and \mathcal{ALCI} [21]. To check a TCQ ϕ for satisfiability w.r.t. a TKB \mathcal{K} , we first enumerate all possible sets \mathcal{S} and mappings ι , which can be done in 2-EXP. For each of these double exponentially many pairs (\mathcal{S}, ι) , we then check t-satisfiability of $\phi^{\mathcal{P}}$ w.r.t. \mathcal{S} and ι in exponential time (see Proposition 9) and test \mathcal{S} for r-satisfiability w.r.t. ι and \mathcal{K} (using Proposition 11). By Proposition 8, ϕ has a model w.r.t. \mathcal{K} iff at least one pair passes both tests.

For the r-satisfiability test, observe that the conjunction of CQ-literals $\chi_{\mathcal{S},\iota}$ contains exponentially many (negated) CQs, each of size polynomial in the size of ϕ , and that $\mathcal{T}_{\mathcal{S},\iota}$ and $\mathcal{R}_{\mathcal{S},\iota}$ are of exponential size in the size of \mathcal{K} . By Lemma 10, the satisfiability of $\chi_{\mathcal{S},\iota}$ w.r.t. $(\mathcal{T}_{\mathcal{S},\iota}, \mathcal{R}_{\mathcal{S},\iota})$ can thus be checked in double exponential time in the size of ϕ and \mathcal{K} . For the data complexity, observe that the number of CQs in $\chi_{\mathcal{S},\iota}$ is linear in the size of the input ABoxes, and their size only depends on ϕ (the size of a single assertion is constant). Moreover, $\mathcal{T}_{\mathcal{S},\iota}$ and $\mathcal{R}_{\mathcal{S},\iota}$ are of size linear in n . Lemma 10 thus yields the claimed upper bound. \square

By the same arguments, it is easy to see that TCQ entailment in $\mathcal{ALCHOIQ}$ is decidable since this is the case for UCQ (non-)entailment [28].

5 Data Complexity without Rigid Roles

To obtain a tight bound on the data complexity if we disallow rigid role names, we follow a different approach from [4]. Similarly to the previous section, we decide r-satisfiability of \mathcal{S} w.r.t. ι and \mathcal{K} by constructing conjunctions of CQ-literals of which we want to check satisfiability. However, we do not compile the whole r-satisfiability check into just one conjunction. More precisely, we define the conjunctions of CQ-literals $\gamma_i \wedge \chi_{\mathcal{S}}$, $0 \leq i \leq n$, w.r.t. $(\mathcal{T}_{\mathcal{S}}, \mathcal{R}_{\mathcal{S}})$, where

$$\begin{aligned} \gamma_i &:= \bigwedge_{\alpha \in \mathcal{A}_i} \alpha^{(\iota(i))}, & \chi_{\mathcal{S}} &:= \bigwedge_{1 \leq i \leq k} \chi_i^{(i)}, \\ \mathcal{T}_{\mathcal{S}} &:= \{\beta^{(i)} \mid \beta \in \mathcal{T} \text{ and } 1 \leq i \leq k\}, \\ \mathcal{R}_{\mathcal{S}} &:= \{\gamma^{(i)} \mid \gamma \in \mathcal{R} \text{ and } 1 \leq i \leq k\}. \end{aligned}$$

This separates the consistency checks for the individual ABoxes \mathcal{A}_i , $1 \leq i \leq n$, from each other. For r-satisfiability, we additionally have to make sure that rigid consequences of the form $A(a)$ for a rigid concept name $A \in \mathbf{N}_{\text{RC}}$ and an individual name $a \in \mathbf{N}_I$ are shared between all the conjunctions $\gamma_i \wedge \chi_{\mathcal{S}}$. It suffices to do this for the set $\text{RCon}(\mathcal{T})$ of rigid concept names occurring in \mathcal{T} since those that occur only in ABox assertions cannot affect the entailment of the TCQ ϕ .

For this purpose, we guess a set $\mathcal{D} \subseteq 2^{\text{RCon}(\mathcal{T})}$ that fixes the combinations of rigid concept names that are allowed to occur in the models of $\gamma_i \wedge \chi_{\mathcal{S}}$, and a function $\tau: \text{Ind}(\phi) \cup \text{Ind}(\mathcal{K}) \rightarrow \mathcal{D}$ that assigns to each individual name one such combination. To express this formally, we extend the TBox by the axioms in

$$\mathcal{T}_{\mathcal{D}} := \{A_Y \equiv C_Y \mid Y \in \mathcal{D}\},$$

where A_Y are fresh rigid concept names and, for every $Y \subseteq \text{RCon}(\mathcal{T})$,

$$C_Y := \prod_{A \in Y} A \sqcap \prod_{A \in \text{RCon}(\mathcal{T}) \setminus Y} \neg A.$$

The size of $\mathcal{T}_{\mathcal{D}}$ is bounded polynomially in the sizes of \mathcal{D} and $\text{RCon}(\mathcal{T})$, which are constant w.r.t. data complexity. We now extend the conjunctions $\gamma_i \wedge \chi_{\mathcal{S}}$ by

$$\rho_{\tau} := \bigwedge_{a \in \text{Ind}(\phi) \cup \text{Ind}(\mathcal{K})} A_{\tau(a)}(a)$$

in order to fix the behavior of the rigid concept names on the named individuals.

We need one more definition to formulate the main lemma of this section. We say that an interpretation \mathcal{I} *respects* \mathcal{D} if

$$\mathcal{D} = \{Y \subseteq \text{RCon}(\mathcal{T}) \mid \text{there is a } d \in \Delta^{\mathcal{I}} \text{ with } d \in (C_Y)^{\mathcal{I}}\},$$

which means that every combination of rigid concept names in \mathcal{D} is realized by a domain element of \mathcal{I} , and conversely, the domain elements of \mathcal{I} may only realize those combinations that occur in \mathcal{D} .

Lemma 13. *Let the DL \mathcal{L} be contained in SHIQ , ALCHOQ , or ALCHOI . If $\text{N}_{\text{RR}} = \emptyset$, then \mathcal{S} is *r-satisfiable* w.r.t. ι and \mathcal{K} iff there exist $\mathcal{D} \subseteq 2^{\text{RCon}(\mathcal{T})}$ and $\tau: \text{Ind}(\phi) \cup \text{Ind}(\mathcal{K}) \rightarrow \mathcal{D}$ such that each $\gamma_i \wedge \chi_{\mathcal{S}} \wedge \rho_{\tau}$, $0 \leq i \leq n$, has a model w.r.t. $(\mathcal{T}_{\mathcal{S}} \cup \mathcal{T}_{\mathcal{D}}, \mathcal{R}_{\mathcal{S}})$ that respects \mathcal{D} . \square*

The restriction imposed by \mathcal{D} can be expressed as the conjunction of CQ-literals

$$\sigma_{\mathcal{D}} := (\neg \exists x. A_{\mathcal{D}}(x)) \wedge \bigwedge_{Y \in \mathcal{D}} \exists x. A_Y(x),$$

where $A_{\mathcal{D}}$ is a fresh concept name that is restricted by adding the axiom $A_{\mathcal{D}} \equiv \prod_{Y \in \mathcal{D}} \neg A_Y$ to the TBox. We denote by $\mathcal{T}'_{\mathcal{S}}$ the resulting extension of $\mathcal{T}_{\mathcal{S}} \cup \mathcal{T}_{\mathcal{D}}$, and have now reduced the r-satisfiability of \mathcal{S} w.r.t. ι and \mathcal{K} to the consistency of $\gamma_i \wedge \chi_{\mathcal{S}} \wedge \rho_{\tau} \wedge \sigma_{\mathcal{D}}$ w.r.t. $(\mathcal{T}'_{\mathcal{S}}, \mathcal{R}_{\mathcal{S}})$.

Theorem 14. *Let \mathcal{L} be a DL that contains $\text{AL}\mathcal{E}$ and is contained in SHIQ , ALCHOQ , or ALCHOI . Then TCQ entailment in \mathcal{L} is CO-NP-complete w.r.t. data complexity.*

Proof. The lower bound follows from CO-NP-hardness of instance checking in $\text{AL}\mathcal{E}$ [29]. To test satisfiability of a TCQ ϕ w.r.t. a TKB \mathcal{K} , we employ the same approach as before, but instead guess \mathcal{S} and ι . Since \mathcal{S} is of constant size in the size of the ABoxes and ι is of linear size, this can be done in nondeterministic polynomial time. The t-satisfiability test for Proposition 8 can be done in polynomial time by Proposition 9, and for the r-satisfiability test, we use Lemma 13.

Following the reduction described above, we guess a set $\mathcal{D} \subseteq 2^{\text{RCon}(\mathcal{T})}$ and a function $\tau: \text{Ind}(\phi) \cup \text{Ind}(\mathcal{K}) \rightarrow \mathcal{D}$, which can be done in nondeterministic polynomial time since \mathcal{D} only depends on \mathcal{T} and τ is of size linear in the size of the input ABoxes. Next, we check the satisfiability of the polynomially many conjunctions $\gamma_i \wedge \chi_{\mathcal{S}} \wedge \rho_{\tau} \wedge \sigma_{\mathcal{D}}$ w.r.t. $(\mathcal{T}'_{\mathcal{S}}, \mathcal{R}_{\mathcal{S}})$. Note that $\chi_{\mathcal{S}}$, $\sigma_{\mathcal{D}}$, $\mathcal{T}'_{\mathcal{S}}$, and $\mathcal{R}_{\mathcal{S}}$ do not depend on the input ABoxes, while γ_i and ρ_{τ} are of polynomial size. Furthermore, the size of the CQs in γ_i and ρ_{τ} is constant. Hence, Lemma 10 yields the desired NP upper bound for these satisfiability tests. \square

6 Conclusions

Query answering w.r.t. DL ontologies is currently a very active research area. We have extended complexity results for very expressive DLs underlying the

web ontology language OWL 2 to the case of temporal queries. Our results show that, w.r.t. worst-case complexity, adding a temporal dimension often comes for free. In fact, in all sublogics of \mathcal{SHOIQ} , the upper bounds for the combined complexity of TCQ entailment obtained in this paper for the temporal case coincide with the best known upper bounds for atemporal query entailment (even in the presence of rigid roles). From the application point of view, data complexity is more important since the amount of data is often very large, and in comparison the size of the background knowledge and the user query is small. We have shown that, in many cases, the atemporal data complexity of CO-NP does not increase if we consider TCQs with rigid concepts (specifically, in \mathcal{ALCHOQ} , \mathcal{ALCHOI} , \mathcal{SHIQ} , and sublogics). For the remaining logics of Table 2, it is an open problem to find a CO-NP algorithm even in the atemporal case.

As part of future work, we will try to obtain CO-NP upper bounds even in the presence of rigid roles, and study extensions of TCQs with concrete domains and inconsistency-tolerant semantics. Since CO-NP is already a rather negative result for data complexity, we could also try to find restricted formalisms with lower data complexity. On the one hand, one could take a less expressive DL to formulate the background ontology, which has already been investigated for \mathcal{EL} [11] and $DL-Lite_{horn}^H$ [10], but only the latter choice reduces the data complexity (to ALOGTIME). On the other hand, one could investigate whether the data complexity can be reduced by imposing additional restrictions on the TBox or CQs, as has been done in the atemporal case [7, 8, 22].

References

1. Artale, A., Kontchakov, R., Wolter, F., Zakharyashev, M.: Temporal description logic for ontology-based data access. In: Proc. IJCAI'13. pp. 711–717 (2013)
2. Baader, F.: Ontology-based monitoring of dynamic systems. In: Proc. KR'14. pp. 678–681 (2014)
3. Baader, F., Borgwardt, S., Lippmann, M.: Temporal conjunctive queries in expressive DLs with non-simple roles. LTCSS-Report 15-17, Chair of Automata Theory, TU Dresden (2015), <http://lat.inf.tu-dresden.de/research/reports.html>
4. Baader, F., Borgwardt, S., Lippmann, M.: Temporal query entailment in the description logic \mathcal{SHQ} . J. Web Semant. 33, 71–93 (2015)
5. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. 2 edn. (2007)
6. Baader, F., Ghilardi, S., Lutz, C.: LTL over description logic axioms. ACM T. Comput. Log. 13(3), 21:1–21:32 (2012)
7. Bienvenu, M., Ortiz, M., Šimkus, M., Xiao, G.: Tractable queries for lightweight description logics. In: Rossi, F. (ed.) Proc. IJCAI'13. pp. 768–774 (2013)
8. Bienvenu, M., ten Cate, B., Lutz, C., Wolter, F.: Ontology-based data access: A study through disjunction Datalog, CSP, and MMSNP. ACM T. Database Syst. 39(4), 33:1–33:44 (2014)
9. Bonatti, P.A., Lutz, C., Murano, A., Vardi, M.Y.: The complexity of enriched μ -calculi. Log. Meth. Comput. Sci. 4(3) (2008)
10. Borgwardt, S., Thost, V.: Temporal query answering in $DL-Lite$ with negation. In: Proc. GCAI'15 (2015), to appear.

11. Borgwardt, S., Thost, V.: Temporal query answering in the description logic \mathcal{EL} . In: Proc. IJCAI'15. pp. 2819–2825 (2015)
12. Calvanese, D., Eiter, T., Ortiz, M.: Regular path queries in expressive description logics with nominals. In: Proc. IJCAI'09. pp. 714–720 (2009)
13. Calvanese, D., Eiter, T., Ortiz, M.: Answering regular path queries in expressive description logics via alternating tree-automata. *Inf. Comput.* 237, 12–55 (2014)
14. Decker, S., Erdmann, M., Fensel, D., Studer, R.: Ontobroker: Ontology based access to distributed and semi-structured information. In: Proc. DS-8. pp. 351–369 (1999)
15. Eiter, T., Lutz, C., Ortiz, M., Šimkus, M.: Query answering in description logics with transitive roles. In: Proc. IJCAI'09. pp. 759–764 (2009)
16. Glimm, B., Horrocks, I., Lutz, C., Sattler, U.: Conjunctive query answering for the description logic SHIQ . *J. Artif. Intell. Res.* 31(1), 157–204 (2008)
17. Glimm, B., Horrocks, I., Sattler, U.: Unions of conjunctive queries in SHOQ . In: Proc. KR'08. pp. 252–262 (2008)
18. Glimm, B., Kazakov, Y., Lutz, C.: Status QIO : An update. In: Proc. DL'11. pp. 136–146 (2011)
19. Horrocks, I., Sattler, U., Tobies, S.: Practical reasoning for very expressive description logics. *L. J. IGPL* 8(3), 239–263 (2000)
20. Lippmann, M.: Temporalised Description Logics for Monitoring Partially Observable Events. Ph.D. thesis, TU Dresden, Germany (2014)
21. Lutz, C.: The complexity of conjunctive query answering in expressive description logics. In: Proc. IJCAR'08. pp. 179–193 (2008)
22. Lutz, C., Wolter, F.: Non-uniform data complexity of query answering in description logics. In: Brewka, G., Eiter, T., McIlraith, S.A. (eds.) Proc. KR'12. pp. 297–307 (2012)
23. Ngo, N., Ortiz, M., Šimkus, M.: The combined complexity of reasoning with closed predicates in description logics. In: Proc. DL'15. pp. 249–261 (2015)
24. Ortiz, M., Calvanese, D., Eiter, T.: Data complexity of query answering in expressive description logics via tableaux. *J. Autom. Reasoning* 41(1), 61–98 (2008)
25. Ortiz, M., Šimkus, M.: Reasoning and query answering in description logics. In: Reasoning Web. 8th Int. Summer School, chap. 1, pp. 1–53 (2012)
26. Pnueli, A.: The temporal logic of programs. In: Proc. SFCS'77. pp. 46–57 (1977)
27. Poggi, A., Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Linking data to ontologies. *J. Data Semant.* X, 133–173 (2008)
28. Rudolph, S., Glimm, B.: Nominals, inverses, counting, and conjunctive queries or: Why infinity is your friend! *J. Artif. Intell. Res.* 39(1), 429–481 (2010)
29. Schaerf, A.: On the complexity of the instance checking problem in concept languages with existential quantification. *J. Intell. Inf. Syst.* 2(3), 265–278 (1993)