Inconsistency-Tolerant Ontology-Based Data Access Revisited:
Taking Mappings into Account

Meghyn Bienvenu

French National Center for Scientific Research (CNRS)
University of Montpellier
French Institute for Research in Computer Science and Automation (Inria)

Abstract

Inconsistency-tolerant query answering in the presence of ontologies has received considerable attention in recent years. However, existing work assumes that the data is expressed using the vocabulary of the ontology and is therefore not directly applicable to ontology-based data access (OBDA), where relational data is connected to the ontology via mappings. This motivates us to revisit existing results in the wider context of OBDA with mappings. After formalizing the problem, we perform a detailed analysis of the data complexity of inconsistency-tolerant OBDA for ontologies formulated in DL-Lite and other data-tractable description logics, considering three different semantics (AR, IAR, and brave), two notions of repairs (subset and symmetric difference), and two classes of global-as-view (GAV) mappings. We show that adding plain GAV mappings does not affect data complexity, but there is a jump in complexity if mappings with negated atoms are considered.

1 Introduction

Ontology-based data access aims to improve access to data (typically stored in a relational database) by using an ontology to provide a conceptual view of the data that describes the semantic relationships holding between different terms [Poggi *et al.*, 2008]. As the ontology usually introduces a more convenient high-level vocabulary that abstracts from the way data is stored, mappings are used to relate the terms in the ontology with the relations in the database. The user query is formulated using the ontology vocabulary, and query answers are computed by combining the information from the data, the mapping, and ontology. Most of the work on OBDA considers ontologies expressed using description logics (DLs) [Baader *et al.*, 2017]. The DL-Lite family of DLs [Calvanese *et al.*, 2007; Artale *et al.*, 2009] has been shown to be especially well suited for OBDA, as it is possible for query answering to be reduced, via query rewriting and unfolding, to query evaluation over relational databases.

The focus of this paper is on handling data inconsistencies in OBDA. It is widely acknowledged that real-world data suffers from numerous data quality issues, and errors in data are frequent. In the OBDA setting, such errors can lead to logical contradictions, in which case standard OBDA semantics (based upon classical first-order logic) trivializes. Fixing the errors by making changes to the underlying data is typically impossible, as we often do not have permission to modify the data (and even if we do, it may not be clear which modifications should be made). A solution is to adopt inconsistency-tolerant semantics, which allow meaningful answers to be obtained from inconsistent data.

The problem of querying inconsistent data using alternative semantics has been extensively studied by the database community, under the name of consistent query answering [Arenas *et al.*, 1999; Bertossi, 2011; Chomicki, 2007]. In the database setting, inconsistencies arise from violations of integrity constraints, and a repair is a database that satisfies the constraints and differs minimally from the original database. Various notions of repairs have been considered, among them, subset repairs ($\subseteq$-repairs), which are maximal consistent subsets of the database, and symmetric difference repairs ($\oplus$-repairs), which may both add and delete facts and minimize the set of such changes. Consistent query answering semantics returns those query answers that hold in every repair.

The topic of inconsistency-tolerant query answering has been recently taken up by the DL community, and the last few years have seen a flurry of activity, with proposals of different inconsistency-tolerant semantics [Lembo *et al.*, 2010; Bienvenu and Rosati, 2013], extensive studies of the complexity of query answering under said semantics [Rosati, 2011; Bienvenu, 2012], and some first implemented systems [Bienvenu *et al.*, 2014; Lembo *et al.*, 2015; Tsalapati *et al.*, 2016]. We refer the reader to the survey [Bienvenu and Bourgaux, 2016] for an overview and further references. However, all of these works focus on a simplified version of OBDA, in which the data is given as a set of ABox facts using the vocabulary of the ontology (aka TBox). The question of how to define repairs in the full OBDA setting with mappings, and how the presence of mappings and type of repair affects the complexity of query answering have not yet been addressed.

The present paper constitutes a first study of full-fledged inconsistency-tolerant OBDA. We begin by proposing a notion of OBDA repair, which is defined at the level of the database, with the mapping and ontology serving to define consistent instances. As the repairs involve modifications of the database, we in fact consider two notions: $\subseteq$-repairs

		AR	IAR	brave
GAV	DL-Lite	coNP-c	in AC^0	in AC^0
	PTIME DLs	coNP-c	coNP-c	NP-c
GAV$^{\neg,\neq}$	DL-Lite	Π_2^p-c	Π_2^p-c	Σ_2^p-c
	PTIME DLs	Π_2^p-c	Π_2^p-c	Σ_2^p-c

Figure 1: Data complexity of CQ entailment under AR, IAR, and brave semantics, for GAV and GAV$^{\neg,\neq}$ mappings. The results hold for both $\subseteq$- and $\oplus$-repairs. Lower bounds for PTIME DLs hold for all DLs extending $\mathcal{EL}_\perp$.

and $\oplus$-repairs. We then show how to transfer existing repair-based semantics to our setting. We consider three such semantics: AR semantics (the DL name for consistent query answering semantics), brave semantics (the dual of AR, which requires a query to hold in *some* repair), and IAR semantics (a strengthening of AR semantics, which queries the intersection of all repairs).

We perform a detailed study of the data complexity of OBDA under these semantics. We show results both for DL-Lite and for the general class of 'data-tractable' DLs, i.e. DLs for which consistency checking and query answering are in PTIME for data complexity. This class includes DLs of the $\mathcal{EL}$ family [Baader *et al.*, 2005] as well as more expressive Horn DLs like Horn-$\mathcal{SHIQ}$ [Eiter *et al.*, 2008]. Mappings in OBDA typically take the form of global-as-view (GAV) mappings. For this initial study, we consider two forms of GAV mappings, one which only allows positive atoms in mapping bodies and a more expressive form of GAV mapping whose bodies can contain negated atoms and inequalities. Mappings with complex bodies (in particular, negated atoms) are supported by existing OBDA systems and have been shown to be useful in applications of OBDA (see discussion in Section 6).

Our data complexity results are summarized in Figure 1. For GAV mappings without negated atoms, we obtain precisely the same complexity as in the simple OBDA setting without mappings, and in particular, the tractability results for DL-Lite under IAR and brave semantics are preserved. By contrast, for the richer form of GAV mappings with negation, there is a sharp increase in complexity, with all problems moving to the second level of the polynomial hierarchy. Interestingly, although they require different techniques, $\subseteq$-repairs and $\oplus$-repairs also yield the same complexity.

2 Preliminaries

We recall basic notions about relational databases, description logics, and ontology-based data access (OBDA).

Databases A database *signature* Σ_{DB} consists of a finite set of relation symbols, each with an associated arity. A database *instance* over signature Σ_{DB} (or Σ_{DB}-instance) is a finite set of *facts* of the form $P(c_1, \ldots, c_n)$, where $P \in \Sigma_{DB}$ is an n-ary relation symbol, and $c_1, \ldots, c_n$ are constants. Every instance $\mathcal{D}$ corresponds to a finite first-order logic interpretation, whose domain contains the constants in $\mathcal{D}$ and which interprets relation P as $\{\vec{c} \mid P(\vec{c}) \in \mathcal{D}\}$.

We consider three types of atoms: relational atoms of the form $P(t_1, \ldots, t_n)$, equality atoms of the form $t_1 = t_2$, and inequality atoms $t_1 \neq t_2$, where the terms t_i are either

constants or variables. A *first-order query* (FO-query) is a first-order formula built up from such atoms using the usual Boolean connectives ($\land, \lor, \neg$) and quantifiers ($\forall, \exists$). We will consider several types of FO-queries. *Conjunctive queries* (CQs) are existentially quantified conjunctions of relational atoms. *Conjunctive queries with inequality* (CQ$^{\neq}$s) additionally allow for inequality atoms. *Conjunctive queries with safe negation* (CQ$^{\neg}$s) extend CQs with negated relational atoms, but require that every variable that appears in a negated relational atom also appears in a positive relational atom. We use CQ$^{\neg,\neq}$ to refer to CQs with both inequalities and safe negation. *Unions of conjunctive queries* (UCQs) are disjunctions of CQs such that every disjunct has the same set of free variables. We can define in the same manner UCQ$^{\neq}$s and UCQ$^{\neg,\neq}$s. A query is *Boolean* if it contains no free variables. A Boolean query q is satisfied by instance $\mathcal{D}$, written $\mathcal{D} \models q$, if q evaluates to true in the interpretation corresponding to $\mathcal{D}$.

Integrity constraints express properties that database instances should satisfy. *Universal constraints* take the form $\forall \vec{x} \, (\alpha_1 \land \ldots \alpha_n \land \varepsilon \rightarrow \beta_1 \lor \ldots \lor \beta_m)$ where the α_i and β_j are relational atoms all of whose terms belong to $\vec{x}$, and ϵ is a (possibly empty) conjunction of inequality atoms. *Denial constraints* are a subclass of universal constraints in which the right-hand side of the implication is empty, denoted $\perp$. Note that denial constraints capture functional dependencies (FDs) and key constraints, two common forms of constraints. An instance $\mathcal{D}$ *satisfies a set of constraints* $\mathcal{C}$, written $\mathcal{D} \models \mathcal{C}$, if every constraint in $\mathcal{C}$ evaluates to true in the interpretation corresponding to $\mathcal{D}$. A database instance $\mathcal{D}$ is *consistent w.r.t.* $\mathcal{C}$ if $\mathcal{D} \models \mathcal{C}$, and inconsistent otherwise.

Description Logics A DL *knowledge base (KB)* consists of an ABox and a TBox, which are constructed from a set N_C of *concept names* (unary relation symbols) and a set of N_R of *role names* (binary relation symbols). The *ABox* (dataset) is a finite set of facts of the forms $A(a)$ and $R(a, b)$, with $A \in N_C$ and $R \in N_R$. The *TBox* (ontology) consists of a finite set of axioms whose form depends on the chosen DL.

We will be particularly interested in DLs from the DL-Lite family. The core DL-Lite language allows only for *concept inclusions* $B_1 \sqsubseteq (\neg B_2)$ where each B_i is either some $A \in N_C$ or $\exists P$, with P either a role name R or inverse role R^-. However, our results will hold for all common DL-Lite dialects, including ones whose TBoxes allow for role inclusions, or functionality assertions [Calvanese *et al.*, 2007].

Another DL we will consider is $\mathcal{EL}_\perp$ [Baader *et al.*, 2005], whose TBoxes consists of concept inclusions $C \sqsubseteq D$, where C, D are complex concepts formed according to the grammar $C := \top \mid \perp \mid A \mid \exists R.C \mid C \sqcap C$, where $A \in N_C$ and $R \in N_R$. We observe that $\mathcal{EL}_\perp$ can express *atomic concept disjointness* using inclusions of the form $A \sqcap B \sqsubseteq \perp$ (which is equivalent to the DL-Lite axiom $A \sqsubseteq \neg B$).

The semantics of DL KBs is given by first-order logic *interpretations* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where $\Delta^\mathcal{I}$ is a non-empty domain and $\cdot^\mathcal{I}$ maps every $A \in N_C$ to a unary relation $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$ and every $R \in N_R$ to a binary relation $R^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$. The function $\cdot^\mathcal{I}$ is extended to complex concepts and roles, e.g. $\perp^\mathcal{I} = \emptyset$ and $(C \sqcap D)^\mathcal{I} = C^\mathcal{I} \cap D^\mathcal{I}$. A concept inclusion $C \sqsubseteq D$ is satisfied in $\mathcal{I}$ if $C^\mathcal{I} \subseteq D^\mathcal{I}$. A *model* of a KB $\langle \mathcal{A}, \mathcal{T} \rangle$ is an interpretation that satisfies all axioms in $\mathcal{T}$ and

facts in $\mathcal{A}$. We say that $\langle \mathcal{A}, \mathcal{T} \rangle$ is *consistent* if it has a model, and otherwise, it is inconsistent, written $\langle \mathcal{A}, \mathcal{T} \rangle \models \bot$.

Querying DL KBs Conjunctive queries are the main query language used to query DL KBs. To simplify the presentation, we focus w.l.o.g. on Boolean CQs. We say that a *Boolean CQ q is entailed from a KB* $\langle \mathcal{A}, \mathcal{T} \rangle$, written $\langle \mathcal{A}, \mathcal{T} \rangle \models q$, if q is satisfied in every model of $\langle \mathcal{A}, \mathcal{T} \rangle$.

Query rewriting is a technique that can be used to reduce KB consistency and CQ entailment to database query evaluation. Formally, we say that an FO-query $q_\bot$ is a *rewriting of inconsistency w.r.t.* $\mathcal{T}$ if $\langle \mathcal{A}, \mathcal{T} \rangle \models q$ iff $\mathcal{A}_{\mathsf{DB}} \models q'$, for every ABox $\mathcal{A}$. Note that here $\mathcal{A}_{\mathsf{DB}}$ signifies that $\mathcal{A}$ is treated like a database instance (i.e. facts not present in $\mathcal{A}$ are treated as false). An FO-query q' is a *rewriting of a Boolean CQ q w.r.t. TBox* $\mathcal{T}$ if $\langle \mathcal{A}, \mathcal{T} \rangle \models q$ iff $\mathcal{A}_{\mathsf{DB}} \models q'$, for every ABox $\mathcal{A}$.

It is well known that for every DL-Lite TBox $\mathcal{T}$, we can construct a $\mathrm{UCQ}^{\neq}$ that is a rewriting of inconsistency, and for every Boolean CQ q, we can construct a $\mathrm{UCQ}^{\neq}$ q' that is a rewriting of q w.r.t. $\mathcal{T}$ [Calvanese *et al.*, 2007].

Ontology-based Data Access In OBDA, we have a TBox $\mathcal{T}$, but data is stored in a relational database $\mathcal{D}$, rather than an ABox. Typically, $\mathcal{D}$ is formulated using a different signature than $\mathcal{T}$, so a *mapping* is used to relate $\mathcal{D}$ and $\mathcal{T}$. In this paper, we consider two forms of global-as-view (GAV) mappings. A (plain) *GAV mapping* consists of a set of rules $\forall \vec{x} \varphi \rightarrow \gamma$, where φ is a CQ all of whose variables are free and among $\vec{x}$, and γ is either a concept atom $A(x)$ or role atom $R(x, x')$ with $x, x' \in \vec{x}$. $\mathrm{GAV}^{\neg, \neq}$ mappings are defined similarly, except that φ can be a $\mathrm{CQ}^{\neg, \neq}$. To simplify the presentation, we will typically omit the initial $\forall \vec{x}$ when writing mappings.

We use the term *OBDA specification* to refer to a triple $\langle \Sigma_{\mathsf{DB}}, \mathcal{M}, \mathcal{T} \rangle$, consisting of a database signature Σ_{DB}, mapping $\mathcal{M}$, and TBox $\mathcal{T}$; by restricting to mappings $\mathcal{M}$ from the mapping language $\mathcal{L}_M$ and TBoxes $\mathcal{T}$ from the DL $\mathcal{L}_O$, we obtain the class of $(\mathcal{L}_M, \mathcal{L}_O)$ OBDA specifications. An *OBDA instance* is also a triple, $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle$, but with an instance $\mathcal{D}$ in the first position. Given an instance $\mathcal{D}$ and mapping $\mathcal{M}$, the *induced ABox* $\mathcal{M}(\mathcal{D})$ is the ABox obtained by applying the mapping to $\mathcal{D}$ and collecting the resulting facts. More precisely, for every $\varphi \rightarrow \gamma \in \mathcal{M}$ and variable assignment ν such that $\nu(\varphi)$ holds in $\mathcal{D}$, the ABox $\mathcal{M}(\mathcal{D})$ contains the fact $\nu(\gamma)$ (and $\mathcal{M}(\mathcal{D})$ contains only such facts). We say that a CQ q is entailed by an OBDA instance $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle$, written $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle \models q$, if $\langle \mathcal{M}(\mathcal{D}), \mathcal{T} \rangle \models q$.

When $\mathcal{T}$ is a DL-Lite TBox, we can perform query entailment as follows: (1) compute a rewriting q' of the input query q w.r.t. $\mathcal{T}$, (2) *unfold* q' w.r.t. $\mathcal{M}$ to get q'', and (3) evaluate q'' over $\mathcal{D}$. Intuitively, the unfolding of q' w.r.t. $\mathcal{M}$ works by replacing each DL atom $A(t)$ or $R(t, t')$ by a disjunction of the mapping bodies that can create the atom. We refer the reader to [Poggi *et al.*, 2008] for more details of the construction of the induced ABox $\mathcal{M}(\mathcal{D})$ and the unfolding operation.

Remark 1. When computing the induced ABox $\mathcal{M}(\mathcal{D})$, mapping bodies are evaluated using standard (closed-world) database semantics: each database instance corresponds to a single finite interpretation that is intended to provide be (locally) complete. For example, hospital data could be expected to contain a complete list of all diagnoses made within the

hospital. The use of a closed-world semantics on the underlying relational data is entirely compatible with the open-world interpretation of the induced ABox. Indeed, ontologies typically introduce new terms, or give a wider meaning to existing terms. For example, diagnoses from the hospital data will be used to populate ontology terms about medical conditions, but will generally only provide a partial view of a patient's medical conditions (as some diagnoses may have made elsewhere, or the patient may have undiagnosed conditions). Importantly, however, because the database relations are intended to be complete, it can be meaningful to use negation in the bodies of mappings, e.g. to identify patients that are *not* on the opt-in organ donation list.

Complexity We focus on *data complexity*, which is measured with respect to the size of the data (either a database instance or ABox). We consider the following standard complexity classes: PTIME (solvable in deterministic polynomial time), NP (solvable in non-deterministic polynomial time), coNP (complement is in NP), Σ_2^p (solvable in non-deterministic polynomial time with access to an NPoracle), and Π_2^p (complement is in Σ_2^p). We will also mention a circuit complexity class $\mathrm{AC}^0 \subseteq$ PTIME. It is known that FO-query evaluation over database instances is in AC^0 w.r.t. data complexity.

3 Inconsistency-Tolerant OBDA

As previously mentioned, there is a large body of literature on inconsistency-tolerant query answering over relational databases and DL knowledge bases. In this section, we review the key notions from this body of work and then show how to adapt them to the OBDA setting.

3.1 Database and ABox Repairs

Intuitively, a repair of a database $\mathcal{D}$ w.r.t. integrity constraints $\mathcal{C}$ is a database that satisfies $\mathcal{C}$ and is as close as possible to $\mathcal{D}$. In this paper, we consider two well-studied forms of repairs: *symmetric difference repairs* ($\oplus$-repairs) and *subset repairs* ($\subseteq$-repairs). Symmetric difference repairs are the most commonly used and are appropriate in settings where both fact insertions and fact deletions can be used to fix constraint violations (i.e., when facts may be both missing or erroneous). Formally, a $\oplus$-*repair* of an instance $\mathcal{D}$ w.r.t. $\mathcal{C}$ is an instance $\mathcal{R}$ such that (i) $\mathcal{R} \models \mathcal{C}$, and (ii) there is no $\mathcal{R}'$ such that $\mathcal{R}' \oplus \mathcal{D} \subsetneq \mathcal{R} \oplus \mathcal{D}$ and $\mathcal{R}' \models \mathcal{C}$. Here $\oplus$ denotes the symmetric difference operation: $S_1 \oplus S_2 = (S_1 \setminus S_2) \cup (S_2 \setminus S_1)$. If only fact deletions are permitted (i.e. constraint violations are due to the presence of erroneous facts, rather than missing facts), we can use $\subseteq$-repairs instead: $\mathcal{R}$ is a $\subseteq$-*repair* of $\mathcal{D}$ w.r.t. $\mathcal{C}$ iff (i) $\mathcal{R} \subseteq \mathcal{D}$, (ii) $\mathcal{R} \models \mathcal{C}$, and (iii) there is no $\mathcal{R}'$ such that $\mathcal{R} \subsetneq \mathcal{R}'$ and $\mathcal{R}' \models \mathcal{C}$. We use $Rep_\oplus(\mathcal{D}, \mathcal{C})$ and $Rep_\subseteq(\mathcal{D}, \mathcal{C})$ to denote the sets of $\oplus$- and $\subseteq$-repairs, respectively.

In the DL setting, we have the corresponding notion of ABox repair, first proposed in [Lembo *et al.*, 2010]. Formally, an *ABox repair* of an ABox $\mathcal{A}$ w.r.t. a TBox $\mathcal{T}$ is an ABox $\mathcal{R} \subseteq \mathcal{A}$ such that (i) $\langle \mathcal{R}, \mathcal{T} \rangle \not\models \bot$ and (ii) there is no $\mathcal{R}'$ such that $\mathcal{R} \subsetneq \mathcal{R}'$ and $\langle \mathcal{R}', \mathcal{T} \rangle \not\models \bot$. Note that in contrast to the database setting, adding facts to an ABox can never restore consistency, which is why it only makes sense to con-

sider repairs that are subsets of $\mathcal{A}$. We will use $ARep(\mathcal{A}, \mathcal{T})$ to refer to the set of repairs of $\mathcal{A}$ w.r.t. $\mathcal{T}$.

In this paper, we consider three natural repair-based semantics that have been proposed for querying DL KBs. A query is entailed under *AR semantics* if it is entailed from *every* ABox repair. The AR semantics essentially corresponds to the standard semantics used for consistent query answering over databases [Arenas *et al.*, 1999] but translated to the DL setting [Lembo *et al.*, 2010]. Under the stronger *IAR semantics*, put forth in [Lembo *et al.*, 2010], a query is entailed if it is entailed from the *intersection of the repairs*, i.e. from the 'surest' ABox facts. The weakest reasonable semantics is the so-called *brave semantics* [Bienvenu and Rosati, 2013], which only requires that a query be entailed from *some* repair. These semantics can be profitably used together, either to identify query results of different levels of confidence, or to improve efficiency, as the IAR and brave semantics can sometimes be used as tractable upper and lower approximations of AR semantics, see [Bienvenu and Bourgaux, 2016] for discussion. For $S \in \{AR, IAR, brave\}$, we will use $\langle \mathcal{A}, \mathcal{T} \rangle \models_S q$ for query entailment over DL KB $\langle \mathcal{A}, \mathcal{T} \rangle$ under S semantics.

To more easily relate semantics based upon ABox, database, and OBDA repairs, we will adopt the notation[1] $\langle \mathcal{D}, \mathcal{C} \rangle \models_S^r q$, with $r \in \{\oplus, \subseteq\}$ indicating the type of repair, to refer to query entailment using the database analog of semantics $S \in \{AR, IAR, brave\}$. For $S = AR$ and $S = brave$, the definition is the same but uses database repairs in place of ABox repairs (when $S = AR$, this is just the standard consistent query answering semantics). Suitably translating the IAR semantics to the database setting proves more subtle and will be addressed later in the section (Definition 10).

3.2 OBDA Repairs

In order to transfer the preceding repair-based semantics to the OBDA setting, we need to formalize what we mean by a repair of an OBDA instance $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle$. There are two natural options: either we consider the ABox repairs of $\mathcal{M}(\mathcal{D})$ w.r.t. $\mathcal{T}$ ('*map then repair*' approach), or we compute repairs of $\mathcal{D}$ using the pair $(\mathcal{M}, \mathcal{T})$ to determine consistent database instances ('*repair at source*' approach).

The 'map then repair' approach has some appealing characteristics, as it leads to repairs preserving more ABox facts and allows us to more easily reuse techniques for querying ABox repairs. Moreover, similar strategies have been employed in the data integration setting [Bravo and Bertossi, 2003; Calì *et al.*, 2003]. However, it has one major disadvantage: as the approach does not keep track of the origins of the mapped facts, it is possible for repairs to contain ABox facts that originate from conflicting database facts.

Example 2. Consider the instance $\mathcal{D}^* = \{T(a, b, c), U(a, c)\}$, TBox $\mathcal{T}^* = \{\exists P \sqsubseteq \neg A\}$, and mapping $\mathcal{M}^* = \{U(x, y) \rightarrow A(x), U(x, y) \rightarrow B(y), T(x, y, z) \rightarrow P(x, z), T(x, y, z) \rightarrow R(y, z)\}$. The ABox $\mathcal{M}^*(\mathcal{D})$ contains four facts: $P(a, c), R(b, c), A(a), B(b)$. There are two

ABox repairs of $\mathcal{M}(\mathcal{D})$: $\mathcal{R}_1$ drops $P(a, c)$, and $\mathcal{R}_2$ drops $A(a)$. Note that $B(b) \wedge R(b, c)$ holds in both repairs, even though $B(b)$ and $R(b, c)$ originate from conflicting facts. If we adopt the 'repair at source' approach, we will have two repairs: $\mathcal{S}_1 = \{T(a, b, c)\}$ and $\mathcal{S}_2 = \{U(a, c)\}$, with $\mathcal{M}(\mathcal{S}_1) = \{P(a, c), R(b, c)\}$ and $\mathcal{M}(\mathcal{S}_2) = \{A(a), B(b)\}$.

In a recent work on inconsistency-tolerant data exchange [ten Cate *et al.*, 2016], the authors similarly faced a decision as to whether to define repairs on the source dataset, or to perform the repair on the mapped data. They opt for the former in order to avoid the semantic anomalies that can arise when combining facts derived from conflicting facts.

The preceding considerations lead us to investigate the 'repair at source' approach, which can be formalized as follows:

Definition 3. A database instance $\mathcal{R}$ is an $\oplus$-*repair of an OBDA instance* $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle$ if (a) $\langle \mathcal{R}, \mathcal{M}, \mathcal{T} \rangle \not\models \bot$, and (b) there is no $\mathcal{R}'$ with $\mathcal{R}' \oplus \mathcal{D} \subsetneq \mathcal{R} \oplus \mathcal{D}$ with $\langle \mathcal{R}', \mathcal{M}, \mathcal{T} \rangle \not\models \bot$. An instance $\mathcal{R}$ is a $\subseteq$-*repair* of $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle$ if (a) $\mathcal{R} \subseteq \mathcal{D}$, (b) $\langle \mathcal{R}, \mathcal{M}, \mathcal{T} \rangle \not\models \bot$, and (c) there is no $\mathcal{R}'$ with $\mathcal{R} \subsetneq \mathcal{R}'$ such that $\langle \mathcal{R}', \mathcal{M}, \mathcal{T} \rangle \not\models \bot$. For $r \in \{\oplus, \subseteq\}$, we use $Rep_r(\mathcal{D}, \mathcal{M}, \mathcal{R})$ to refer to the set of r-repairs of $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle$.

Remark 4. Another advantage of defining repairs at the level of the database instances is that it is straightforward to integrate database constraints. Indeed, we can easily adapt Definition 3 to the case where the instance comes with a set of integrity constraints $\mathcal{C}$ by simply requiring that possible repairs $\mathcal{R}$ satisfy $\mathcal{R} \models \mathcal{C}$ in addition to $\langle \mathcal{R}, \mathcal{M}, \mathcal{T} \rangle \not\models \bot$.

The following easy lemma, an immediate consequence of Def. 3, shows that every $\subseteq$-repair is also a $\oplus$-repair:

Lemma 5. *For every OBDA instance* $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle$: $Rep_{\subseteq}(\mathcal{D}, \mathcal{M}, \mathcal{T}) \subseteq Rep_{\oplus}(\mathcal{D}, \mathcal{M}, \mathcal{T})$.

In general, however, some $\oplus$-repairs are not $\subseteq$-repairs, as illustrated by the following example:

Example 2 (continued). Reconsider $\mathcal{D}^*$ and $\mathcal{T}^*$, but now with the mapping $\mathcal{M}' = \{T(x, y, z) \wedge \neg N(y) \rightarrow P(x, y), U(x, y) \rightarrow A(x)\}$. There are two $\subseteq$-repairs of $\langle \mathcal{D}^*, \mathcal{M}', \mathcal{T}^* \rangle$: $\mathcal{S}_1$ and $\mathcal{S}_2$. If we consider $\oplus$-repairs instead, then we have a third repair $\mathcal{S}_3 = \mathcal{D}^* \cup \{N(b)\}$. Indeed, by adding $N(b)$, we block the derivation of $P(a, b)$.

3.3 Repair-based Semantics for OBDA

With a notion of repair in hand, we can now define variants of the AR, brave, and IAR semantics for OBDA instances.

Definition 6. Consider an OBDA instance $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle$, CQ q, and $r \in \{\oplus, \subseteq\}$. Then:

- $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle \models_{AR}^r q$ iff $\langle \mathcal{M}(\mathcal{R}), \mathcal{T} \rangle \models q$ for every $\mathcal{R} \in Rep_r(\mathcal{D}, \mathcal{M}, \mathcal{T})$

- $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle \models_{brave}^r q$ iff $\langle \mathcal{M}(\mathcal{R}), \mathcal{T} \rangle \models q$ for some $\mathcal{R} \in Rep_r(\mathcal{D}, \mathcal{M}, \mathcal{T})$

Example 7. Consider the OBDA instance with $\mathcal{D}^\dagger = \{P(a), R(a), S(a)\}$, $\mathcal{M}^\dagger = \{P(x) \rightarrow A(x), R(x) \rightarrow B(x), S(x) \rightarrow D(x), S(x) \wedge \neg P(x) \rightarrow E(x)\}$, and $\mathcal{T}^\dagger = \{A \sqsubseteq \neg B, A \sqsubseteq C, B \sqsubseteq C\}$. The two ($\oplus$- and $\subseteq$-) repairs are: $\mathcal{R}_1^\dagger = \{P(a), S(a)\}$ and $\mathcal{R}_2^\dagger = \{R(a), S(a)\}$.

---

[1] To facilitate comparisons, we are committing an abuse of notation and terminology, as AR and IAR stand for 'ABox Repair' and 'Intersection of ABox Repairs' (conveniently, they also abbreviate the mnemonic 'All Repairs' and 'Intersection of All Repairs').

Their induced ABoxes are: $\mathcal{M} \dagger (\mathcal{R}_1^\dagger) = \{A(a), D(a)\}$ and $\mathcal{M} \dagger (\mathcal{R}_2^\dagger) = \{B(a), D(a), E(a)\}$. Under brave semantics, $A(a), B(a), C(a), D(a)$, and $E(a)$ can all be derived, while under AR semantics, only $C(a)$ and $D(a)$ are entailed.

The most obvious way of defining the IAR semantics in the OBDA setting would be to consider a query q to be entailed if $\langle \mathcal{M}(\mathcal{R}_\cap), \mathcal{T} \rangle \models q$ where $\mathcal{R}_\cap = \bigcap_{\mathcal{R} \in Rep_r(\mathcal{D}, \mathcal{M}, \mathcal{T})} \mathcal{R}$. Note however that $\mathcal{R}_\cap$ satisfies a negated fact if the fact is absent from even a single repair, whereas a positive fact must appear in all repairs. To correct this asymmetry and stay closer to the spirit of the original IAR definition, we propose to compute the induced ABox using the negated facts common to all repairs. This can be formalized as follows, where $\mathcal{F}_s$ and $\mathcal{F}_p$ are intended to contain respectively the set of sure and possible positive facts (i.e. those holding in all / some repair(s)).

Definition 8. Fix two sets of facts $\mathcal{F}_s \subseteq \mathcal{F}_p$. A CQ$^{\neg,\neq}$ q is *satisfied in* $(\mathcal{F}_s, \mathcal{F}_p)$ *under variable assignment* ν (written $(\mathcal{F}_s, \mathcal{F}_p) \models_\mu q$) if μ satisfies all inequality atoms in q and

- for every positive atom $\alpha \in q$: $\nu(\alpha) \in \mathcal{F}_s$
- for every negated atom $\neg\beta \in q$: $\nu(\beta) \notin \mathcal{F}_p$.

Given a GAV$^{\neg,\neq}$ mapping $\mathcal{M}$, we define $\mathcal{M}(\mathcal{F}_s, \mathcal{F}_p)$ as the ABox consisting of those facts $\nu(\gamma)$ for which there exists $\varphi \to \gamma \in \mathcal{M}$ such that $(\mathcal{F}_s, \mathcal{F}_p) \models_\mu \varphi$.

Definition 9. Consider an OBDA instance $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle$ with $\mathcal{M}$ a GAV$^{\neg,\neq}$ mapping, CQ q, and r $\in \{\oplus, \subseteq\}$. Then $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle \models_{IAR}^r q$ iff $\langle \mathcal{M}(\mathcal{R}_\cap, \mathcal{R}_\cup), \mathcal{T} \rangle \models q$ where:

$$\mathcal{R}_\cap = \bigcap_{\mathcal{R} \in Rep_r(\mathcal{D}, \mathcal{M}, \mathcal{T})} \mathcal{R} \qquad \mathcal{R}_\cup = \bigcup_{\mathcal{R} \in Rep_r(\mathcal{D}, \mathcal{M}, \mathcal{T})} \mathcal{R}$$

We return to our example to illustrate the IAR semantics.

Example 7 (continued)**.** By intersecting and unioning the repairs of $\langle \mathcal{D}^\dagger, \mathcal{M}^\dagger, \mathcal{T}^\dagger \rangle$, we get $\mathcal{R}_\cap^\dagger = \{S(a)\}$ and $\mathcal{R}_\cup^\dagger = \{P(a), R(a), S(a)\}$. The ABox $\mathcal{M}^\dagger(\mathcal{R}_\cap^\dagger, \mathcal{R}_\cup^\dagger)$ consists of a single fact, $D(a)$. Note that the mapping $S(x) \wedge \neg P(x) \to E(x)$ cannot be applied since $P(a) \in \mathcal{R}_\cup^\dagger$, whereas $E(a)$ would be entailed with the naïve definition of IAR semantics.

Similar considerations apply when translating the IAR semantics to the database setting, leading us to propose the following definition:

Definition 10. Consider an instance $\mathcal{D}$, set of integrity constraints $\mathcal{C}$, r $\in \{\oplus, \subseteq\}$, and UCQ$^{\neg,\neq}$ $Q = q_1 \vee \ldots \vee q_n$. Then $\langle \mathcal{D}, \mathcal{C} \rangle \models_{IAR}^r Q$ iff $(\mathcal{R}_\cap, \mathcal{R}_\cup) \models_\mu q_i$ for some $1 \leq i \leq n$ and variable assignment μ, where $\mathcal{R}_\cap$ (resp. $\mathcal{R}_\cup$) denotes the intersection (resp. union) of all r-repairs of $\langle \mathcal{D}, \mathcal{C} \rangle$.

The next theorem shows that, as one would hope, the semantics we have defined based upon OBDA repairs obey the same relationships as those for ABox repairs. Note that this result crucially depends on the way the IAR semantics is defined; indeed, it does not hold if one adopts the 'naïve' definition (to see why, consider $\langle \mathcal{D}^\dagger, \mathcal{M}^\dagger, \mathcal{T}^\dagger \rangle$ from Example 7).

Theorem 11. *For every OBDA instance Γ, CQ q, and r $\in \{\oplus, \subseteq\}$:* $\Gamma \models_{IAR}^r q \Rightarrow \Gamma \models_{AR}^r q \Rightarrow \Gamma \models_{brave}^r q$. *The converse implications do not hold in general.*

Proof. The second implication is immediate from Def. 6, and Example 7 prove that the converse implications fail. To show the first implication, suppose $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle \models_{IAR}^r q$, i.e. $\langle \mathcal{M}(\mathcal{R}_\cap, \mathcal{R}_\cup), \mathcal{T} \rangle \models q$, with $\mathcal{R}_\cap$ and $\mathcal{R}_\cup$ defined as in Def. 9. To prove $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle \models_{AR}^r q$, it suffices to show that $\mathcal{M}(\mathcal{R}_\cap, \mathcal{R}_\cup) \subseteq \mathcal{M}(\mathcal{R})$ for every $\mathcal{R} \in Rep_r(\mathcal{D}, \mathcal{M}, \mathcal{T})$. Take some $\mathcal{R} \in Rep_r(\mathcal{D}, \mathcal{M}, \mathcal{T})$ and $\theta \in \mathcal{M}(\mathcal{R}_\cap, \mathcal{R}_\cup)$. By Def. 9, there exists $\varphi \to \gamma \in \mathcal{M}$ and a variable assignment ν that satisfies the inequality atoms in φ and is such that $\theta = \nu(\gamma)$ and $\nu(\alpha) \in \mathcal{R}_\cap$ (resp. $\nu(\beta) \notin \mathcal{R}_\cup$) for every positive atom $\alpha \in \varphi$ (resp. negated atom $\neg\beta$) in φ. It follows from the definition of $\mathcal{R}_\cap$ and $\mathcal{R}_\cup$ that $\nu(\alpha_i) \in \mathcal{R}$ for all $1 \leq i \leq n$ and $\nu(\beta_i) \notin \mathcal{R}$ for all $1 \leq i \leq m$. We can therefore apply the same rule to $\mathcal{R}$ to get $\theta \in \mathcal{M}(\mathcal{R})$. □

The next result shows that the semantics based upon $\oplus$-repairs are more conservative (i.e. entail fewer queries) than those based upon $\subseteq$-repairs.

Theorem 12. *For every OBDA instance $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle$, CQ q, and* S $\in \{AR, IAR\}$: $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle \models_S^\oplus q$ *implies* $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle \models_S^\subseteq q$. *The converse implication do not hold.*

Proof. The implication follows from Def. 6 and Lemma 5. To see why the converse implication does not hold, take $\mathcal{D} = \{P(a,a), T(a)\}$, $\mathcal{M} = \{P(x,y) \wedge \neg S(x,y) \to A(x), P(x,y) \wedge \neg S(x,y) \to B(y), T(x) \wedge \neg P(x,x) \to E(x)\}$, $\mathcal{T} = \{A \sqsubseteq \neg B\}$, and $q = E(a)$. There is a single $\subseteq$-repair $\mathcal{R}_1 = \{T(a)\}$ with induced ABox $\{E(a)\}$, but an additional $\oplus$-repair $\mathcal{R}_2 = \{P(a,a), T(a), S(a,a)\}$ with an empty induced ABox. Thus, for every S $\in \{AR, IAR, brave\}$, $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle \models_S^\subseteq E(a)$, but $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle \not\models_S^\oplus E(a)$. □

It follows from Example 2 that the 'map then repair' approach would lead to some queries being entailed that are not entailed w.r.t. our 'repair at source' approach. In the next example, show the opposite, namely that there are queries entailed according to our approach that would not be entailed in the 'map-then-repair' approach, thereby showing that the two approaches are incomparable w.r.t. the set of entailed queries.

Example 13. Consider the OBDA instance with $\mathcal{D}^\ddagger = \{P(a,a), S(a), T(a)\}$, $\mathcal{M}^\ddagger = \{P(x,y) \to A(x), P(x,y) \to B(y), S(x) \wedge T(x) \wedge \neg P(x,y) \to D(x)\}$, and $\mathcal{T}^\ddagger = \{A \sqsubseteq \neg B\}$. As $\mathcal{M}^\ddagger(\mathcal{D}^\ddagger) = \{A(a), B(a)\}$, we have $\langle \mathcal{M}^\ddagger(\mathcal{D}^\ddagger), \mathcal{T}^\ddagger \rangle \not\models_S D(a)$ for S $\in \{AR, IAR, brave\}$. However, the single OBDA repair $\mathcal{R}^\ddagger = \{S(a), T(a)\}$ gives rise to the induced ABox $\{D(a)\}$, so $\langle \mathcal{D}^\ddagger, \mathcal{M}^\ddagger, \mathcal{T}^\ddagger \rangle \models_S^r D(a)$ for r $\in \{\subseteq, \oplus\}$ and S $\in \{AR, IAR, brave\}$. We argue that it is reasonable to infer $D(a)$ from this inconsistent OBDA instance. Indeed, as the fact $P(a,a)$ contradicts the specification, it must be false. Thus, the negated fact $\neg P(a,a)$ holds in the correct version of the database, and it can be combined with the (presumed correct) facts $S(a)$ and $T(a)$ to fire the mapping rule $S(x) \wedge T(x) \wedge \neg P(x,y) \to D(x)$.

In the next two sections, we will provide algorithms and data complexity results for inconsistency-tolerant OBDA, considering the impact of varying the semantics (AR, IAR, or brave), the type of repair ($\oplus$ or $\subseteq$) and the form of the

mapping (GAV or $GAV^{\neg,\neq}$). In addition to deciding query entailment under the three semantics, we will consider the repair checking problem, which as its name suggests, is to check whether a given instance is a repair.

4 Results for GAV Mappings

In this section, we describe how we obtain the results for GAV mappings. We begin by observing that when GAV mappings are used, it is sufficient to study $\subseteq$-repairs, as every $\oplus$-repair is a $\subseteq$-repair. This is essentially a rephrasing of an analogous result for databases with denial constraints and exploits the fact that consistency can never be recovered by adding facts.

Lemma 14. *For every OBDA instance $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle$ with $\mathcal{M}$ a GAV mapping, $Rep_{\subseteq}(\mathcal{D}, \mathcal{M}, \mathcal{T}) = Rep_{\oplus}(\mathcal{D}, \mathcal{M}, \mathcal{T})$.*

Another easy observation is that when GAV mappings are considered, our way of defining the IAR semantics (Def. 9) coincides with the 'naïve' definition.

Lemma 15. *For every OBDA instance $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle$ with $\mathcal{M}$ a GAV mapping, $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle \models^{\subseteq}_{\mathsf{IAR}} q$ iff $\langle \mathcal{M}(\mathcal{R}_{\cap}), \mathcal{T} \rangle \models q$ where $\mathcal{R}_{\cap} = \bigcap_{\mathcal{R} \in Rep_{\subseteq}(\mathcal{D}, \mathcal{M}, \mathcal{T})} \mathcal{R}$.*

We next establish the upper bounds in Table 1 for 'PTIME DLs', i.e. DLs for which both consistency and CQ answering are in PTIME for data complexity.

Theorem 16. *Let $\mathcal{L}$ be a PTIME description logic. Then CQ answering w.r.t. (GAV, $\mathcal{L}$) OBDA specifications is:*

- *coNP in data complexity, for AR and IAR semantics*

- *NP in data complexity, for brave semantics.*

Proof. The upper bounds for AR and brave semantics are easy and lift the corresponding results for DL KBs.

We first observe that it is in PTIME to recognize $\subseteq$-repairs. Indeed, to show that $\mathcal{R}$ is *not* a repair, it suffices to iterate over all $\tau \in \mathcal{D} \setminus \mathcal{R}$, and show that some $\mathcal{R} \cup \{\tau\}$ is consistent.

For AR semantics, we can show that $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle \not\models^{\subseteq}_{\mathsf{AR}} q$ by guessing $\mathcal{R} \subseteq \mathcal{D}$ and verifying that (i) $\mathcal{R} \in Rep_{\subseteq}(\mathcal{D}, \mathcal{M}, \mathcal{T})$, and (ii) $\langle \mathcal{M}(\mathcal{R}), \mathcal{T} \rangle \not\models q$. Both checks are in PTIME.

For brave semantics, we can show that $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle \models^{\subseteq}_{\mathsf{brave}} q$ by guessing $\mathcal{R} \subseteq \mathcal{D}$ and verifying in PTIME that (i) $\mathcal{R} \in Rep_{\subseteq}(\mathcal{D}, \mathcal{M}, \mathcal{T})$,, and (ii) $\langle \mathcal{M}(\mathcal{R}), \mathcal{T} \rangle \models q$.

For IAR semantics, we adapt an upper bound from [Rosati, 2011]. By Lemma 15, $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle \models^{\subseteq}_{\mathsf{IAR}} q$ iff $\langle \mathcal{M}(\mathcal{R}_{\cap}), \mathcal{T} \rangle \models q$ where $\mathcal{R}_{\cap} = \bigcap_{\mathcal{R} \in Rep_r(\mathcal{D}, \mathcal{M}, \mathcal{T})} \mathcal{R}$. To show $\langle \mathcal{M}(\mathcal{R}_{\cap}), \mathcal{T} \rangle \not\models q$, we guess a subset $\mathcal{N} \subseteq \mathcal{D}$ as well as a subset $\mathcal{R}_{\neg\tau} \subseteq \mathcal{D}$ for every $\tau \notin \mathcal{D} \setminus \mathcal{N}$. We then verify that (i) every $\mathcal{R}_{\neg\tau}$ is a repair such that $\tau \notin \mathcal{R}_{\neg\tau}$, and (ii) $\langle \mathcal{M}(\mathcal{N}), \mathcal{T} \rangle \not\models q$. It follows from (i) that $\mathcal{R}_{\cap} \subseteq \mathcal{N}$, which implies $\mathcal{M}(\mathcal{R}_{\cap}) \subseteq \mathcal{M}(\mathcal{N})$, since $\mathcal{M}$ is a GAV mapping. We then use (ii) to get $\langle \mathcal{M}(\mathcal{R}_{\cap}), \mathcal{T} \rangle \not\models q$. $\qquad\square$

The preceding upper bounds match existing lower bounds for inconsistency-tolerant query answering over KBs expressed in $\mathcal{EL}_{\perp}$ [Rosati, 2011; Bienvenu and Bourgaux, 2016], one of the simplest PTIME DLs. The next result allows us to transfer these lower bounds to the OBDA setting.

Lemma 17. *For every DL $\mathcal{L}$ and $\mathsf{S} \in \{\mathsf{AR}, \mathsf{IAR}, \mathsf{brave}\}$, there is a data-independent reduction from CQ answering under S semantics over $\mathcal{L}$ knowledge bases to CQ answering under S semantics w.r.t. (GAV, $\mathcal{L}$) OBDA specifications.*

Proof. Consider a TBox $\mathcal{T}$ with signature Σ, and let $\mathcal{M}_{\mathsf{copy}} = \{P(\vec{z}) \to P(\vec{z}) \mid P \in \Sigma\}$. As $\mathcal{M}_{\mathsf{copy}}(\mathcal{A}) = \mathcal{A}$, we have $ARep(\mathcal{A}, \mathcal{T}) = Rep_{\subseteq}(\mathcal{A}, \mathcal{M}_{\mathsf{copy}}, \mathcal{T})$. It follows that $\langle \mathcal{A}, \mathcal{T} \rangle \models_{\mathsf{S}} q$ iff $\langle \mathcal{A}, \mathcal{M}, \mathcal{T}' \rangle \models_{\mathsf{S}} q$ for every query q and $\mathsf{S} \in \{\mathsf{AR}, \mathsf{IAR}, \mathsf{brave}\}$ (for IAR, we again use Lemma 15). $\qquad\square$

It is known that CQ answering under IAR and brave semantics is tractable for *DL-Lite* knowledge bases [Lembo *et al.*, 2015; Bienvenu and Rosati, 2013], and we will show that these positive results hold also in the OBDA setting with GAV mappings. We start by giving a reduction to inconsistency-tolerant query answering in the database setting.

Theorem 18. *For every (DL-Lite, GAV) OBDA specification $\langle \Sigma_{\mathsf{DB}}, \mathcal{M}, \mathcal{T} \rangle$ and every CQ q, there exists a set $\mathcal{C}_{\mathcal{M},\mathcal{T}}$ of denial constraints and a $UCQ^{\neq}$ Q_q such that*

$$\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle \models^{\subseteq}_{\mathsf{S}} q \qquad \Leftrightarrow \qquad \langle \mathcal{D}, \mathcal{C}_{\mathcal{M},\mathcal{T}} \rangle \models^{\subseteq}_{\mathsf{S}} Q_q$$

for every Σ_{DB}-instance $\mathcal{D}$ and $\mathsf{S} \in \{\mathsf{AR}, \mathsf{IAR}, \mathsf{brave}\}$.

Proof. Given a (GAV, *DL-Lite*) OBDA specification $\langle \Sigma_{\mathsf{DB}}, \mathcal{M}, \mathcal{T} \rangle$, we proceed as follows:

1. Construct a $UCQ^{\neq}$ Q_q such that for every Σ_{DB}-instance $\mathcal{D}$ such that $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle \not\models \perp$: $\mathcal{D} \models Q_q$ iff $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle \models q$.

2. Construct a $UCQ^{\neq}$ Q_{unsat} such that for every Σ_{DB}-instance $\mathcal{D}$: $\mathcal{D} \models Q_{\mathsf{unsat}}$ iff $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle$ is inconsistent.

3. Let $\mathcal{C}_{\mathcal{M},\mathcal{T}} = \{q_i \to \perp \mid q_i \in Q_{\mathsf{unsat}}\}$.

Steps 1 and 2 can be performed by applying existing query rewriting and unfolding procedures (as described in [Poggi *et al.*, 2008; Kontchakov and Zakharyaschev, 2014]), followed by standard equivalence-preserving transformations. To show correctness of the reduction, we use the facts that $Rep_{\subseteq}(\mathcal{D}, \mathcal{M}, \mathcal{T}) = Rep_{\subseteq}(\mathcal{D}, \mathcal{C}_{\mathcal{M},\mathcal{T}})$, and that $\langle \mathcal{M}(\mathcal{R}), \mathcal{T} \rangle \models q$ iff $\mathcal{R} \models Q_q$, for every repair $\mathcal{R}$. $\qquad\square$

Next we show that $UCQ^{\neq}$ answering with denial constraints under IAR and brave semantics can be achieved by query rewriting. We do this by adapting a result from [Bienvenu and Rosati, 2013] on CQ answering under brave and IAR semantics for first-order rewritable ontology languages.

Theorem 19. *For every data signature Σ_{DB}, set of denial constraints $\mathcal{C}$, $\mathsf{S} \in \{\mathsf{IAR}, \mathsf{brave}\}$, and $UCQ^{\neq}$ q, there exists a first-order query q_{S} such that $\langle \mathcal{D}, \mathcal{C} \rangle \models^{\subseteq}_{\mathsf{S}} q$ iff $\mathcal{D} \models q_{\mathsf{S}}$, for every Σ_{DB}-instance $\mathcal{D}$.*

Proof. A result from [Bienvenu and Rosati, 2013] shows that CQ answering under IAR and brave semantics can be reduced to FO-query evaluation for all known FO-rewritable ontology languages. The construction starts with a $UCQ^{\neq}$ Q_{unsat} that identifies inconsistent datasets and a $UCQ^{\neq}$ Q_q that is a rewriting of the input query q w.r.t. the ontology. These two queries are then manipulated to obtain FO-queries implementing the IAR and brave semantics. Every set $\mathcal{C}$ of denial constraints gives rise to a $UCQ^{\neq}$ $Q_{\mathcal{C}}$ that holds on instances

violating $\mathcal{C}$. We can thus apply the same construction to the input UCQ$^{\neq}q$ and the UCQ$^{\neq}$ $Q_{\mathcal{C}}$ to obtain FO-queries q_{IAR} and q_{brave} such that $\langle \mathcal{D}, \mathcal{C} \rangle \models^{\subseteq}_{\mathsf{S}} q$ iff $\mathcal{D} \models q_{\mathsf{S}}$, for every Σ_{DB}-instance $\mathcal{D}$ and $\mathsf{S} \in \{\mathsf{IAR}, \mathsf{brave}\}$. $\qquad\square$

Combining Theorems 18 and 19, we obtain the following.

Corollary 20. *CQ answering under IAR and brave semantics is in* AC^0 *for (GAV, DL-Lite) OBDA specifications.*

We point out that all upper bounds in this section hold also in the presence of denial constraints over the data signature.

5 Results for GAV$^{\neg,\neq}$ Mappings

We next turn to GAV$^{\neg,\neq}$ mappings, starting by establishing the complexity of repair checking:

Theorem 21. *For every* PTIME *DL $\mathcal{L}$ that can express atomic concept disjointness and every* $\mathsf{r} \in \{\subseteq, \oplus\}$, r-*repair checking for (GAV$^{\neg,\neq}$, $\mathcal{L}$) specifications is co*NP-*complete.*

Proof. Membership in coNP can be shown similarly to the GAV case (proof of Theorem 16), and coNP-hardness for $\oplus$-repairs can be adapted from a similar result for universal constraints [Staworko and Chomicki, 2010].

We thus focus on the lower bound for $\subseteq$-repairs, which is proven by reduction from unsatisfiability of 3CNF formulas. Consider a 3SAT instance $\varphi = \lambda_1 \wedge \ldots \wedge \lambda_n$ over variables $v_1, \ldots, v_m$, where $\lambda_i = \ell_{i,1} \vee \ell_{i,2} \vee \ell_{i,3}$. We use $\mathsf{var}(\ell)$ to refer to the variable of literal ℓ and set $\mathsf{sgn}(\ell) = \mathsf{t}$ (resp. $\mathsf{sgn}(\ell) = \mathsf{f}$) if ℓ is a positive (resp. negative) literal. We consider the TBox $\mathcal{T}_{\mathsf{sat}} = \{T \sqsubseteq V, F \sqsubseteq V, T \sqsubseteq \neg F, V \sqsubseteq \neg V', C \sqsubseteq \neg U\}$ and the mapping $\mathcal{M}$ that contains:

- $A^{\mathsf{t}}(x) \rightarrow T(x)$ and $A^{\mathsf{f}}(x) \rightarrow F(x)$

- for $s \in \{\mathsf{t}, \mathsf{f}\}$: $A^s(x) \wedge \neg Z \rightarrow V'(x)$

- for every $(s_1, s_2, s_3) \in \{\mathsf{t}, \mathsf{f}\}^3$:
  - $Z \wedge S^{s_1 s_2 s_3}(x, y_1, y_2, y_3) \rightarrow C(x)$
  - $S^{s_1 s_2 s_3}(x, y_1, y_2, y_3) \wedge \neg A^{s_1}(y_1)$
    $\wedge \neg A^{s_2}(y_2) \wedge \neg A^{s_3}(y_3) \rightarrow U(x)$

We take the instance $\mathcal{D}_{\varphi}$ consisting of the tuples

$$\{Z\} \cup \{A^{\mathsf{t}}(v_j), A^{\mathsf{f}}(v_j) \mid 1 \leq j \leq m\} \cup$$
$$\{S^{s_1 s_2 s_3}(c_i, u_1, u_2, u_3) \mid \exists 1 \leq i \leq n \, \forall l \in \{1, 2, 3\}:$$
$$u_l = \mathsf{var}(\ell_{i,l}) \text{ and } \mathsf{sgn}(\ell_{i,l}) = s_l\}$$

and let $\mathcal{D}'$ be the restriction of $\mathcal{D}_{\varphi}$ to the relations $S^{s_1 s_2 s_3}$.

We claim that φ is satisfiable iff $\mathcal{D}' \notin Rep_{\subseteq}(\mathcal{D}_{\varphi}, \mathcal{M}, \mathcal{T}_{\mathsf{sat}})$. Indeed, if φ has a satisfying valuation ν, then $\mathcal{D}' \cup \{Z\} \cup \{A^{\mathsf{t}}(v_j) \mid \nu(v_j) = \mathsf{t}\} \cup \{A^{\mathsf{f}}(v_j) \mid \nu(v_j) = \mathsf{f}\}$ is consistent, so $\mathcal{D}'$ is not a repair. Conversely, suppose there is a repair $\mathcal{R}$ with $\mathcal{D}' \subsetneq \mathcal{R}'$. Consistency of $\mathcal{R}$ implies that for every j, either $A^{\mathsf{t}}(v_j) \notin \mathcal{R}$ or $A^{\mathsf{f}}(v_j) \notin \mathcal{R}$. There exists thus a valuation $\nu_{\mathcal{R}}$ to $v_1, \ldots, v_m$ such that $A^s(v_j) \in \mathcal{R}$ implies $\nu(v_j) = s$. We claim that $\nu_{\mathcal{R}}$ satisfies φ. Indeed, $\mathcal{R} \setminus \mathcal{D}' \neq \emptyset$, so $\mathcal{R}$ contains either Z or a fact of the form $A^s(v_j)$. If $Z \notin \mathcal{R}$ and $A^s(v_j) \in \mathcal{R}$, then $\langle \mathcal{M}(\mathcal{R}), \mathcal{T}_{\mathsf{sat}} \rangle$ entails $V(v_j)$ and $V'(v_j)$, contradicting $V \sqsubseteq \neg V'$. Thus, $Z \in \mathcal{R}$, which implies that $C(c_i) \in \mathcal{M}(\mathcal{R})$ for every $1 \leq i \leq n$. It follows that $\mathcal{M}(\mathcal{R})$ cannot contain any $U(c_i)$, and thus for every

$1 \leq i \leq m$, there must exist $1 \leq l \leq 3$, such that $\mathcal{R}$ contains $A^s(v_k)$, where $v_k = \mathsf{var}(\ell_{i,l})$ and $s = \mathsf{sgn}(\ell_{i,l})$. The latter shows that $\nu_{\mathcal{R}}$ satisfies λ_i. $\qquad\square$

We use the preceding result to derive the remaining upper bounds in Figure 1.

Theorem 22. *Let $\mathcal{L}$ be a* PTIME *description logic. Then CQ answering w.r.t. (GAV$^{\neg,\neq}$, $\mathcal{L}$) OBDA specifications is:*

- Π_2^p *in data complexity, under AR and IAR semantics*

- Σ_2^p *in data complexity, under brave semantics,*

for both $\subseteq$- and $\oplus$-repairs.

Proof. By adapting a result about $\oplus$-repairs w.r.t. universal constraints [Staworko and Chomicki, 2010], we can show that $\oplus$-repairs of $\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle$ use only constants from $\mathcal{D}$ and are thus of polynomial size in $|\mathcal{D}|$. This makes it possible for us to reuse / adapt the algorithms from the proof of Theorem 16. As repair checking is now coNP-complete (Theorem 21), these algorithms yield membership in Π_2^p (resp. Σ_2^p) for AR (resp. brave semantics). For the IAR semantics, we need to modify the algorithm as follows. We guess two sets $\mathcal{N}_{\cap}$ and $\mathcal{N}_{\cup}$, together with a set $\mathcal{R}_{\neg\tau}$ (resp. $\mathcal{R}_{\tau}$) of facts for each $\tau \in \mathcal{D} \setminus \mathcal{N}_{\cap}$ (resp. $\tau \in \mathcal{N}_{\cup}$). The guessed sets only use constants from $\mathcal{D}$, so are of polynomial size w.r.t. $\mathcal{D}$. We then check that (i) every $\mathcal{R}_{\neg\tau}$ is a repair such that $\tau \notin \mathcal{R}_{\neg\tau}$, (ii) every $\mathcal{R}_{\tau}$ is a repair such that $\tau \in \mathcal{R}_{\tau}$, and (iii) $\langle \mathcal{M}(\mathcal{N}_{\cap}, \mathcal{N}_{\cup}), \mathcal{T} \rangle \not\models q$. It follows from (i) and (ii) that $\mathcal{R}_{\cap} \subseteq \mathcal{N}_{\cap}$ and $\mathcal{N}_{\cup} \subseteq \mathcal{R}_{\cup}$, which implies that $\mathcal{M}(\mathcal{R}_{\cap}, \mathcal{R}_{\cup}) \subseteq \mathcal{M}(\mathcal{N}_{\cap}, \mathcal{N}_{\cup})$. Combining this with (iii), we obtain $\langle \mathcal{M}(\mathcal{R}_{\cap}, \mathcal{R}_{\cup}), \mathcal{T} \rangle \not\models q$. $\qquad\square$

For *DL-Lite* ontologies coupled with GAV$^{\neg,\neq}$ mappings, it is still possible to proceed by reduction to the database setting if we accept to move to a more expressive class of constraints:

Theorem 23. *For every (DL-Lite, GAV$^{\neg,\neq}$) OBDA specification $\langle \Sigma_{\mathsf{DB}}, \mathcal{M}, \mathcal{T} \rangle$ and every CQ q, there exists a set $\mathcal{C}_{\mathcal{M},\mathcal{T}}$ of universal constraints and a UCQ$^{\neg,\neq}$ q' such that*

$$\langle \mathcal{D}, \mathcal{M}, \mathcal{T} \rangle \models^{\mathsf{r}}_{\mathsf{S}} q \qquad \Leftrightarrow \qquad \langle \mathcal{D}, \mathcal{C}_{\mathcal{M},\mathcal{T}} \rangle \models^{\mathsf{r}}_{\mathsf{S}} q'$$

for every Σ_{DB}-instance $\mathcal{D}$, semantics $\mathsf{S} \in \{\mathsf{AR}, \mathsf{IAR}, \mathsf{brave}\}$, and $\mathsf{r} \in \{\subseteq, \oplus\}$.

Proof. We proceed similarly to the proof of Theorem 18. The key difference is that because the mapping bodies can contain negated atoms, the queries Q_q and Q_{unsat} from Steps 1 and 2 will take the form of UCQ$^{\neg,\neq}$s rather than UCQ$^{\neq}$s. Each disjunct of Q_{unsat} can be written as $\bigwedge_j \alpha_j \wedge \bigwedge_k \neg\beta_k$, with each α_j a positive atom or inequality atom, and each β_k a positive atom. In Step 3, such a disjunct will be translated into the universal constraint $\bigwedge_j \alpha_j \rightarrow \bigvee_i \beta_i$. Another difference is that for the IAR semantics, we need to work with $\mathcal{M}(\mathcal{R}_{\cap}, \mathcal{R}_{\cup})$, so the proof of correctness is a bit more involved. $\qquad\square$

Although the preceding theorem does not lead to improved upper bounds, it is relevant from the practical viewpoint as it suggests an alternative algorithmic approach.

We now establish matching lower bounds:

Theorem 24. *For every* PTIME *DL $\mathcal{L}$ that can express atomic concept disjointness, CQ answering w.r.t. (GAV$^{\neg,\neq}$, $\mathcal{L}$) OBDA specifications is*

- Π_2^p*-hard in data complexity, for AR and IAR semantics*
- Σ_2^p*-hard in data complexity, for brave semantics,*

for both $\subseteq$- and $\oplus$-repairs.

Proof. For $\oplus$-repairs, we can adapt an existing lower bound for query answering under AR semantics w.r.t. universal constraints [Staworko and Chomicki, 2010]. As the reduction uses a ground atomic query, it directly applies to IAR semantics and can be slightly modified to handle brave semantics.

For $\subseteq$-repairs, we give a reduction from $\forall\exists$-QBF validity. Consider a $\forall\exists$-QBF instance $\Phi = \forall v_1, \ldots, v_k \exists v_{k+1} \ldots v_m \, \varphi$ where $\varphi = \lambda_1 \wedge \ldots \wedge \lambda_n$ is a 3CNF formula, defined as in the proof of Theorem 21. We use the TBox $\mathcal{T}_{\mathsf{sat}}$ from before, and consider the mapping $\mathcal{M}$ consisting of:

- for $p \in \{\forall, \exists\}$: $A^{\mathsf{t},\mathsf{q}}(x) \to T(x)$ and $A^{\mathsf{f},\mathsf{q}}(x) \to F(x)$
- for $s \in \{\mathsf{t}, \mathsf{f}\}$: $A^{s,\exists}(x) \wedge \neg Z \to V'(x)$
- for every $(s_1, s_2, s_3) \in \{\mathsf{t}, \mathsf{f}\}^3$ and $(p_1, p_2, p_3) \in \{\forall, \exists\}^3$:
  - $Z \wedge S^{s_1 p_1 s_2 p_2 s_3 p_3}(x, y_1, y_2, y_3) \to C(x)$
  - $S^{s_1 p_1 s_2 p_2 s_3 p_3}(x, y_1, y_2, y_3) \wedge \neg A^{s_1 p_1}(y_1)$
    $\wedge \neg A^{s_2 p_2}(y_2) \wedge \neg A^{s_3 p_3}(y_3) \to U(x)$
- $Z_0(x) \wedge Z \to Q(x)$

We consider the following instance $\mathcal{D}_\Phi$:

$$\{Z, Z_0(a)\} \cup \{A^{s,\forall}(v_i) \mid 1 \leq i \leq k, s \in \{\mathsf{t}, \mathsf{f}\}\} \cup$$

$$\{A^{s,\exists}(v_i) \mid k+1 \leq i \leq m, s \in \{\mathsf{t}, \mathsf{f}\}\} \cup$$

$$\{S^{s_1 p_1 s_2 p_2 s_3 p_3}(c_i, u_1, u_2, u_3) \mid \exists 1 \leq i \leq n \, \forall l \in \{1, 2, 3\}:$$

$$u_l = \mathsf{var}(\ell_{i,l}), s_l = \mathsf{sgn}(\ell_{i,l}), \text{ and } p_l = \mathsf{quant}(u_l)\}$$

where $\mathsf{quant}(v_i) = \forall$ when $1 \leq i \leq k$ and $\mathsf{quant}(v_i) = \exists$ when $k+1 \leq i \leq m$.

We claim that Φ is valid iff every $\mathcal{R} \in Rep_\subseteq(\mathcal{D}_\varphi, \mathcal{M}, \mathcal{T}_{\mathsf{sat}})$ contains Z. First suppose for a contradiction that Φ is valid, but there is a repair $\mathcal{R}$ such that $Z \notin \mathcal{R}$. As $\mathcal{R}$ is a $\subseteq$-maximal consistent subset, it must contain $Z_0(a)$ and all of the $S^{s_1 p_1 s_2 p_2 s_3 p_3}$-facts, and cannot contain any $A^{\mathsf{t},\exists}(v_j)$ or $A^{\mathsf{t},\exists}(v_j)$ facts. Moreover, for every $1 \leq j \leq k$, exactly one of $A^{\mathsf{t},\forall}(v_j)$ and $A^{\mathsf{t},\forall}(v_j)$ belongs to $\mathcal{R}$. We can thus define an assignment $\nu_\forall$ to $v_1, \ldots, v_k$ by setting $\nu(v_i) = \mathsf{t}$ if $A^{\mathsf{t},\forall}(v_i) \in \mathcal{R}$ and $\nu(v_i) = \mathsf{f}$ if $A^{\mathsf{f},\forall}(v_i) \in \mathcal{R}$. As Φ is valid, there exists an extension ν' of ν to the variables $v_{k+1}, \ldots, v_m$ that satisfies all clauses in Φ. Let $\mathcal{R}'$ be obtained from $\mathcal{R}$ by adding Z and all facts $A^{s,\exists}(v_i)$ such that $\nu(v_i) = s$ and $k+1 \leq i \leq m$. It can be verified that $\mathcal{R}'$ is consistent, contradicting the maximal consistency of $\mathcal{R}$. For the other direction, suppose every $\mathcal{R} \in Rep_\subseteq(\mathcal{D}_\varphi, \mathcal{M}, \mathcal{T}_{\mathsf{sat}})$ contains Z, and consider a valuation ν of the universal variables. Then there exists a repair $\mathcal{R}$ such that $A^{\mathsf{t},\forall}(v_i) \in \mathcal{R}$ if $\nu(v_i) = \mathsf{t}$, and $A^{\mathsf{f},\forall}(v_i) \in \mathcal{R}$ if $\nu(v_i) = \mathsf{f}$. As $\mathcal{R}$ is consistent, it contains at most one of $A^{\mathsf{t},\exists}(v_i)$ and $A^{\mathsf{t},\exists}(v_i)$ for every $k+1 \leq v_i \leq m$. We let ν' be a variable assignment to $v_1, \ldots, v_k$ 'induced' by $\mathcal{R}$ (observe that ν' extends ν). Consider some clause λ_i. As $\mathcal{R}$ contains Z and $S^{s_1 p_1 s_2 p_2 s_3 p_3}(c_i, u_1, u_2, u_3)$, $C(c_i) \in$

$\mathcal{M}(\mathcal{R})$, hence $U(c_i) \notin \mathcal{M}(\mathcal{R})$. It follows that $\mathcal{R}$ contains at least one of $A^{s_1 p_1}(u_1)$, $A^{s_2 p_2}(u_2)$, and $A^{s_3 p_3}(u_3)$, from which we can infer that ν' satisfies λ_i.

To complete the proof, we note that for $\mathsf{S} \in \{\mathsf{AR}, \mathsf{IAR}\}$, $\langle \mathcal{D}_\Phi, \mathcal{M}, \mathcal{T} \rangle \models_\mathsf{S}^\subseteq Q(a)$ iff every $\mathcal{R} \in Rep_\subseteq(\mathcal{D}_\varphi, \mathcal{M}, \mathcal{T}_{\mathsf{sat}})$ contains Z. Indeed, $Q(a)$ holds in the induced ABox just in the case that the repair (or positive intersection of repairs) contains $Z_0(a)$ and Z (and $Z_0(a)$ holds in every repair). For brave semantics, a similar reduction can be used: it suffices to replace $Z_0(x) \wedge Z \to Q(x)$ by $Z_0(x) \wedge \neg Z \to Q(x)$. $\square$

We remark that all upper bounds in this section hold also in the presence of universal constraints over the data signature.

6 Discussion and Future Work

This paper provides a first study of inconsistency-tolerant querying in the OBDA setting, where data are stored in relational databases and linked to the ontology via mappings. Interestingly, while our framework builds upon existing work on querying inconsistent DL KBs, the addition of mappings introduces significant conceptual and computational challenges. Indeed, it is not immediately clear how to appropriately transfer the notion of repairs to the OBDA setting, nor how to define a suitable OBDA analog of the IAR semantics. To justify our proposed definitions, we have exhibited desirable properties that our definitions satisfy but which alternative definitions do not. Our complexity study showed that the inconsistency-tolerant semantics are highly sensitive to the shape of mappings (unlike standard OBDA semantics): while plain GAV mappings do not increase the complexity compared to the mapping-free setting, our three repair-based semantics jump to the second level of the polynomial hierarchy for mappings with negated atoms.

The experience gained in deploying the OBDA paradigm in practical applications has highlighted the need for rich mappings (allowing for complex mapping bodies, with negation, subqueries, and even aggregation), in order to bridge the (often large) semantic gap between the database and ontology vocabularies[2]. For example, in a project on the Italian public debt, the data contained a relation storing all loans and another with all shared loans, and a mapping with negation was required to be able to access single-borrower loans [Antonioli *et al.*, 2014]. It is also relevant to note that the R2RML mapping language[3] (a W3C standard for mapping relational data to RDF), allows for arbitrary SQL expressions in mapping bodies, and R2RML mappings is used by the well-known OBDA systems Ontop [Calvanese *et al.*, 2017] and Mastro[4].

In future work, we plan to extend our investigation to a wider range of mapping languages (in particular, full R2RML) and to develop practical algorithms. Initially, we plan to focus on DL-Lite ontologies and leverage the reduction to the database setting. We expect that we will be able

---

[2]Here we cite the experience of researchers at DIAG, La Sapienza that have been involved in several industrial OBDA projects in which complex mappings proved indispensable. Riccardo Rosati, personal communication, April 2018.

[3]https://www.w3.org/TR/r2rml/

[4]http://www.obdasystems.com

to exploit some existing algorithms and optimizations from consistent query answering, but we will also need to develop new techniques, in particular, for handling the IAR and brave semantics with constraints, as these semantics are as of yet little explored in the database setting.

Acknowledgments

References

[Antonioli *et al.*, 2014] Natalia Antonioli, Francesco Castanò, Spartaco Coletta, Stefano Grossi, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Emanuela Virardi, and Patrizia Castracane. Ontology-based data management for the italian public debt. In *Proceedings of FOIS*, pages 372–385, 2014.

[Arenas *et al.*, 1999] Marcelo Arenas, Leopoldo E. Bertossi, and Jan Chomicki. Consistent query answers in inconsistent databases. In *Proceedings of PODS*, 1999.

[Artale *et al.*, 2009] Allessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyaschev. The DL-Lite family and relations. *Journal Artif. Intell. Res. (JAIR)*, 36:1–69, 2009.

[Baader *et al.*, 2005] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proceedings of IJCAI*, pages 364–369, 2005.

[Baader *et al.*, 2017] Franz Baader, Ian Horrocks, Carsten Lutz, and Ulrike Sattler. *An Introduction to Description Logic*. Cambridge University Press, 2017.

[Bertossi, 2011] Leopoldo E. Bertossi. *Database Repairing and Consistent Query Answering*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2011.

[Bienvenu and Bourgaux, 2016] Meghyn Bienvenu and Camille Bourgaux. Inconsistency-tolerant querying of description logic knowledge bases. In *Reasoning Web Tutorial Lectures*, LNCS 9885, pages 156–202. Springer, 2016.

[Bienvenu and Rosati, 2013] Meghyn Bienvenu and Riccardo Rosati. Tractable approximations of consistent query answering for robust ontology-based data access. In *Proceedings of IJCAI*, 2013. Long version with proofs can be obtained from the author's website.

[Bienvenu *et al.*, 2014] Meghyn Bienvenu, Camille Bourgaux, and François Goasdoué. Querying inconsistent description logic knowledge bases under preferred repair semantics. In *Proceedings of AAAI*, 2014.

[Bienvenu, 2012] Meghyn Bienvenu. On the complexity of consistent query answering in the presence of simple ontologies. In *Proceedings of AAAI*, 2012.

[Bravo and Bertossi, 2003] Loreto Bravo and Leopoldo E. Bertossi. Logic programs for consistently querying data integration systems. In *Proceedings of IJCAI*, pages 10–15, 2003.

[Calì *et al.*, 2003] Andrea Calì, Domenico Lembo, and Riccardo Rosati. Query rewriting and answering under constraints in data integration systems. In *Proceedings of IJCAI*, pages 16–21, 2003.

[Calvanese *et al.*, 2007] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *Journal of Automated Reasoning (JAR)*, 39(3):385–429, 2007.

[Calvanese *et al.*, 2017] Diego Calvanese, Benjamin Cogrel, Sarah Komla-Ebri, Roman Kontchakov, Davide Lanti, Martin Rezk, Mariano Rodriguez-Muro, and Guohui Xiao. Ontop: Answering SPARQL queries over relational databases. *Semantic Web*, 8(3):471–487, 2017.

[Chomicki, 2007] Jan Chomicki. Consistent query answering: Five easy pieces. In *Proceedings of ICDT*, 2007.

[Eiter *et al.*, 2008] Thomas Eiter, Georg Gottlob, Magdalena Ortiz, and Mantas Simkus. Query answering in the description logic Horn-$\mathcal{SHIQ}$. In *Proceedings of JELIA*, 2008.

[Kontchakov and Zakharyaschev, 2014] Roman Kontchakov and Michael Zakharyaschev. An introduction to description logics and query rewriting. In *Reasoning Web Tutorial Lectures*, LNCS 8714, pages 195–244. Springer, 2014.

[Lembo *et al.*, 2010] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. Inconsistency-tolerant semantics for description logics. In *Proceedings of RR*, 2010.

[Lembo *et al.*, 2015] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. Inconsistency-tolerant query answering in ontology-based data access. *Journal of Web Semantics*, 33:3–29, 2015.

[Poggi *et al.*, 2008] Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. Linking data to ontologies. *Journal of Data Semantics*, 10:133–173, 2008.

[Rosati, 2011] Riccardo Rosati. On the complexity of dealing with inconsistency in description logic ontologies. In *Proceedings of IJCAI*, 2011.

[Staworko and Chomicki, 2010] Slawomir Staworko and Jan Chomicki. Consistent query answers in the presence of universal constraints. *Information Systems*, 35(1):1–22, 2010.

[ten Cate *et al.*, 2016] Balder ten Cate, Richard L. Halpert, and Phokion G. Kolaitis. Exchange-repairs - managing inconsistency in data exchange. *Journal of Data Semantics*, 5(2):77–97, 2016.

[Tsalapati *et al.*, 2016] Eleni Tsalapati, Giorgos Stoilos, Giorgos B. Stamou, and George Koletsos. Efficient query answering over expressive inconsistent description logics. In *Proceedings of IJCAI*, 2016.