Inconsistency Handling in Ontology-Mediated Query Answering: A Progress Report

Meghyn Bienvenu

CNRS & University of Bordeaux, France

Abstract. This paper accompanies an invited talk on inconsistency handling in OMQA and presents a concise summary of the research that has been conducted in the area.

1 Introduction

It is widely acknowledged that real-world data is plagued with numerous data quality issues, among them the presence of erroneous facts. While already a serious issue for 'plain' databases, the problem of handling imperfect data is even more critical in the setting of ontology-mediated query answering (OMQA), where even a single erroneous fact can provoke an inconsistency, thereby rendering classical OMQA semantics useless. This has motivated DL and KR researchers to study a variety of approaches for handling inconsistent data in OMQA, adapting and extending techniques initially proposed for databases. Now that there has been about a decade of research on inconsistency handling in OMQA, the time is ripe to take a step back and evaluate the progress that has been made and what remains to be done.

In this paper, we will try to summarize what is now quite a large collection of works related to inconsistency handling in OMQA. Our treatment will necessarily be incomplete. We will focus on the case of inconsistencies in the data (i.e., we assume the ontology has been properly debugged) and mainly discuss how inconsistency-tolerant semantics can be used to obtain meaningful information from inconsistent knowledge bases. The material we cover can be mostly found in the survey chapter [9], and we invite interested readers to consult that chapter for a much more detailed presentation, with complete references and further examples and discussion.

Given the target audience, we will assume that the reader is familiar with DLs and the basics of OMQA. Throughout the paper, we assume that $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is a knowledge base (KB) with TBox $\mathcal{T}$ and ABox $\mathcal{A}$, q is a conjunctive query, and $\boldsymbol{a}$ is a tuple of constants from $\mathcal{A}$ of the same arity as q. We say that $\mathcal{A}$ is $\mathcal{T}$*-consistent* if the KB $\langle \mathcal{T}, \mathcal{A} \rangle$ is consistent, i.e. it has at least one model, and otherwise, it is $\mathcal{T}$-inconsistent. A subset $\mathcal{A}' \subseteq \mathcal{A}$ is called a *minimal* $\mathcal{T}$*-inconsistent subset* of $\mathcal{A}$ if (i) $\mathcal{A}'$ is $\mathcal{T}$-inconsistent, and (ii) every $\mathcal{A}'' \subsetneq \mathcal{A}'$ is $\mathcal{T}$-consistent. A subset $C \subseteq \mathcal{A}$ is a *(consistent)* $\mathcal{T}$*-support* of $q(\boldsymbol{a})$ if (i) C is $\mathcal{T}$-consistent, and (ii) $\langle \mathcal{T}, C \rangle \models q(\boldsymbol{a})$, i.e. $\boldsymbol{a}$ is a certain answer to q w.r.t. $\langle \mathcal{T}, C \rangle$.

2 Inconsistency-Tolerant Semantics

As mentioned in the introduction, the usual first-order semantics of DLs does not provide any useful information in the presence of inconsistencies, as everything can be inferred from a contradiction. To address this limitation of classical semantics, several inconsistency-tolerant semantics have been proposed with the aim of returning meaningful answers to queries posed over inconsistent KBs.

A key notion that underlies many of the proposed semantics is that of a repair, which intuitively captures the different ways of restoring consistency while retaining as much of the original information as possible. If we use set inclusion to select the maximal ABoxes, as was proposed in [21] and many subsequent works, then repairs can be formalized as follows.

Definition 1. *An* (ABox) repair *of an ABox $\mathcal{A}$ w.r.t. a TBox $\mathcal{T}$ is an inclusion-maximal subset of $\mathcal{A}$ that is $\mathcal{T}$-consistent. We use $Rep(\mathcal{A}, \mathcal{T})$ to denote the set of repairs of $\mathcal{A}$ w.r.t. $\mathcal{T}$, which we abbreviate to $Rep(\mathcal{K})$ when $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$.*

Each repair is $\mathcal{T}$-consistent, so it is possible to query a repair using classical semantics. The difficulty, however, is that there are typically several different repairs for an inconsistent KB, so we need to decide how to combine the answers obtained from the different repairs. Arguably the most natural approach is to require that a tuple be a certain answer no matter which repair is considered. This idea is captured by the AR semantics, which was first defined in [21] and can be seen as the OMQA analog of the consistent query answering approach long studied in the database literature [1, 16, 5].

Definition 2 (AR semantics). *A tuple $\boldsymbol{a}$ is an answer to q over $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ under the* AR (ABox Repair) semantics*, written $\mathcal{K} \models_{\mathsf{AR}} q(\boldsymbol{a})$, just in the case that $\langle \mathcal{T}, \mathcal{B} \rangle \models q(\boldsymbol{a})$ for every repair $\mathcal{B} \in Rep(\mathcal{K})$.*

A more conservative semantics, termed the IAR semantics [21], is obtained by querying the intersection of the repairs (or equivalently, the set of assertions not participating in any minimal inconsistent subset).

Definition 3 (IAR semantics). *A tuple $\boldsymbol{a}$ is an answer to q over $\mathcal{K}$ under the* IAR (Intersection of ABox Repairs) semantics*, written $\mathcal{K} \models_{\mathsf{IAR}} q(\boldsymbol{a})$, just in the case that $\langle \mathcal{T}, \mathcal{D} \rangle \models q(\boldsymbol{a})$ where $\mathcal{D} = \bigcap_{\mathcal{B} \in Rep(\mathcal{K})} \mathcal{B}$.*

The more adventurous brave semantics, first explored in the OMQA setting in [14], merely requires that an answer hold w.r.t. at least some repair.

Definition 4 (Brave semantics). *A tuple $\boldsymbol{a}$ is an answer to q over $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ under the* brave semantics*, written $\mathcal{K} \models_{\mathsf{brave}} q(\boldsymbol{a})$, just in the case that $\langle \mathcal{T}, \mathcal{B} \rangle \models q(\boldsymbol{a})$ for some repair $\mathcal{B} \in Rep(\mathcal{K})$.*

Before proceeding further, let us illustrate the AR, IAR, and brave semantics on an example (borrowed and adapted from [9]):

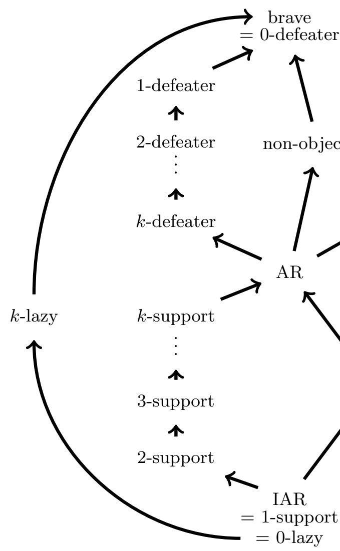

Fig. 1: Relationships between inconsistency-tolerant semantics. An arrow $S \rightarrow S'$ means that S under-approximates S', i.e., $\langle \mathcal{T}, \mathcal{A} \rangle \models_S q(\boldsymbol{a}) \Rightarrow \langle \mathcal{T}, \mathcal{A} \rangle \models_{S'} q(\boldsymbol{a})$.

Example 1. Consider the DL-Lite TBox $\mathcal{T}_{\mathsf{univ}}$ with the following axioms:

$$\begin{array}{llll} \mathsf{Prof} \sqsubseteq \mathsf{Faculty} & \mathsf{Prof} \sqsubseteq \exists \mathsf{Teaches} & \mathsf{Prof} \sqsubseteq \neg \mathsf{Lect} & \mathsf{Faculty} \sqsubseteq \neg \mathsf{Course} \\ \mathsf{Lect} \sqsubseteq \mathsf{Faculty} & \mathsf{Lect} \sqsubseteq \exists \mathsf{Teaches} & \mathsf{Prof} \sqsubseteq \neg \mathsf{Fellow} & \\ \mathsf{Fellow} \sqsubseteq \mathsf{Faculty} & \exists \mathsf{Teaches}^- \sqsubseteq \mathsf{Course} & \mathsf{Lect} \sqsubseteq \neg \mathsf{Fellow} & \end{array}$$

together with the ABox

$$\begin{aligned} \mathcal{A}_{\mathsf{univ}} \quad = \quad & \{\mathsf{Prof}(\mathsf{sam}), \mathsf{Lect}(\mathsf{sam}), \mathsf{Fellow}(\mathsf{sam}), \mathsf{Prof}(\mathsf{kim}), \mathsf{Lect}(\mathsf{kim}), \\ & \mathsf{Fellow}(\mathsf{jane}), \mathsf{Teaches}(\mathsf{cs34}, \mathsf{jane}), \mathsf{Fellow}(\mathsf{alex}), \mathsf{Teaches}(\mathsf{alex}, \mathsf{cs48})\} \end{aligned}$$

The KB $\langle \mathcal{T}_{\mathsf{univ}}, \mathcal{A}_{\mathsf{univ}} \rangle$ is inconsistent and can be shown to have twelve repairs. If we evaluate the query $q(x) = \mathsf{Faculty}(x)$ using the three semantics, we obtain:

- 3 answers for AR semantics: sam, kim, alex
- 1 answer for IAR semantics: alex
- 4 answers for brave semantics: sam, kim, alex, jane

The preceding three semantics can be related as follows (see also Fig. 1):

$$\mathcal{K} \models_{\mathsf{IAR}} q(\boldsymbol{a}) \quad \Rightarrow \quad \mathcal{K} \models_{\mathsf{AR}} q(\boldsymbol{a}) \quad \Rightarrow \quad \mathcal{K} \models_{\mathsf{brave}} q(\boldsymbol{a})$$

In other words, the brave and IAR semantics provide respectively upper and lower bounds on the set of answers w.r.t. AR semantics.

We can also compare these (and later) semantics based upon the properties they satisfy. Three salient properties for an inconsistency-tolerant semantics are:

	Semantics which satisfy the property
Consistent Results	non-objection, CAR, ICAR, AR, ICR, k-support, k-lazy, IAR
Consistent Support	brave, k-defeater, non-objection, AR, ICR, k-support, k-lazy, IAR
Unique Base	IAR, ICR, ICAR

Fig. 2: Properties of inconsistency-tolerant semantics.

Consistent Results Semantics S has the Consistent Support property if for every KB $\langle \mathcal{T}, \mathcal{A} \rangle$, query q, and tuple $\boldsymbol{a}$, if $\langle \mathcal{T}, \mathcal{A} \rangle \models_S q(\boldsymbol{a})$, then there exists a $\mathcal{T}$-support $C \subseteq \mathcal{A}$ of $q(\boldsymbol{a})$.

Consistent Support Semantics S has the Consistent Results property if for every KB $\langle \mathcal{T}, \mathcal{A} \rangle$, there exists a model $\mathcal{I}$ of $\mathcal{T}$ such that $\mathcal{I} \models q(\boldsymbol{a})$ for every $q(\boldsymbol{a})$ with $\langle \mathcal{T}, \mathcal{A} \rangle \models_S q(\boldsymbol{a})$.

Unique Base Semantics S has the Unique Base property if for every KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, there exists a $\mathcal{T}$-consistent ABox $\mathcal{A}'$ such that for every query q and tuple $\boldsymbol{a}$: $\langle \mathcal{T}, \mathcal{A} \rangle \models_S q(\boldsymbol{a})$ iff $\langle \mathcal{T}, \mathcal{A}' \rangle \models q(\boldsymbol{a})$.

The interest of Consistent Results is that it allows users to safely combine the results obtained when querying under semantics S. The Consistent Support property means that every answer can be backed up by exhibiting a consistent subset of the original ABox. Finally, the Unique Base property is a nice feature from the implementation point of view, since it means we compute in an offline phase a consistent ABox and then employ any existing querying algorithms.

As seen in Fig. 2, the brave semantics satisfies only Consistent Support, the AR semantics satisfies both Consistent Support and Consistent Results, while the IAR semantics satisfies all three properties.

Let us now continue on to other semantics that have been proposed in the OMQA literature, starting with the ICR semantics, defined in [7]:

Definition 5 (ICR semantics). *A tuple* $\boldsymbol{a}$ *is an answer to* q *over* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ *under the* ICR (Intersection of Closed Repairs) semantics *just in the case that* $\langle \mathcal{T}, \mathcal{D} \rangle \models q(\boldsymbol{a})$ *where* $\mathcal{D} = \bigcap_{\mathcal{B} \in Rep(\mathcal{K})} \mathsf{close}_{\mathcal{T}}(\mathcal{B})$.

By closing the repairs before intersecting them, the ICR semantics provides a better lower approximation of the AR semantics than the IAR semantics, and it can shown to satisfy the three properties. This semantics coincides with the AR semantics on instance queries, which means that in our example, the ICR semantics would return sam, kim, alex as answers to $q(x) = \mathsf{Faculty}(x)$.

The idea of adding inferred assertions in order to keep more information is also at the heart of the CAR and ICAR semantics proposed in [21]. The key difference is that a modified closure operator is applied to the original inconsistent ABox, and the enriched ABox is then used to define closed ABox repairs:

Definition 6 (Closed ABox repair). *Let* $\mathsf{close}^*_{\mathcal{T}}(\mathcal{A}) = \{\beta \mid \exists S \subseteq \mathcal{A}$ *such that* S *is* $\mathcal{T}$*-consistent and* $\langle \mathcal{T}, S\rangle \models \beta\}$*. A subset* $\mathcal{R} \subseteq \mathsf{close}^*_{\mathcal{T}}(\mathcal{A})$ *is a* closed ABox repair *of* $\mathcal{A}$ *w.r.t.* $\mathcal{T}$ *if (i) it is* $\mathcal{T}$*-consistent, and (ii) there is no* $\mathcal{T}$*-consistent* $\mathcal{R}' \subseteq \mathsf{close}^*_{\mathcal{T}}(\mathcal{A})$ *such that* $\mathcal{R} \cap \mathcal{A} \subsetneq \mathcal{R}' \cap \mathcal{A}$ *or* $\mathcal{R} \cap \mathcal{A} = \mathcal{R}' \cap \mathcal{A}$ *and* $\mathcal{R} \subsetneq \mathcal{R}'$*. If* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$*, the set of closed ABox repairs of* $\mathcal{A}$ *w.r.t.* $\mathcal{T}$ *is denoted* $\mathit{ClosedRep}(\mathcal{K})$*.*

Closed ABox repairs can be seen as maximally 'completing' the (plain) ABox repairs with assertions from $\mathsf{close}^*_{\mathcal{T}}(\mathcal{A}) \setminus \mathcal{A}$. Note that while a closed ABox repair of $\langle \mathcal{T}, \mathcal{A}\rangle$ is always a repair of the KB $\langle \mathcal{T}, \mathsf{close}^*_{\mathcal{T}}(\mathcal{A})\rangle$, repairs of $\langle \mathcal{T}, \mathsf{close}^*_{\mathcal{T}}(\mathcal{A})\rangle$ need not be closed ABox repairs of $\langle \mathcal{T}, \mathcal{A}\rangle$ (see [9] for an example).

Definition 7 (CAR and ICAR semantics). *A tuple* $\boldsymbol{a}$ *is an answer to* q *over* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$ *under the* CAR (Closed ABox Repair) semantics *just in the case that* $\langle \mathcal{T}, \mathcal{R}\rangle \models q(\boldsymbol{a})$ *for every* $\mathcal{R} \in \mathit{ClosedRep}(\mathcal{K})$*. It is an answer under the* ICAR (Intersection of Closed ABox Repairs) semantics *just in the case that* $\langle \mathcal{T}, \mathcal{D}\rangle \models q(\boldsymbol{a})$ *where* $\mathcal{D}$ *is the intersection of the closed ABox repairs of* $\mathcal{A}$ *w.r.t.*$\mathcal{T}$*.*

On the KB from Example 1, the CAR semantics gives the same answers as the AR semantics, and the ICAR semantics coincides with ICR semantics. We provide another example (borrowed from [9]) to show how these semantics differ.

Example 2. Reconsider $\mathcal{T}_{\mathsf{univ}}$ and $\mathcal{A}_{\mathsf{univ}}$ from Example 1, and let $\mathcal{T}'_{\mathsf{univ}}$ be obtained by adding $\exists \mathsf{Teaches} \sqsubseteq \mathsf{Faculty}$ to $\mathcal{T}_{\mathsf{univ}}$. Applying the closure operator yields:

$$\mathsf{close}^*_{\mathcal{T}'_{\mathsf{univ}}}(\mathcal{A}_{\mathsf{univ}}) \;=\; \mathcal{A}_{\mathsf{univ}} \cup \{\mathsf{Faculty(sam)}, \mathsf{Faculty(kim)}, \mathsf{Faculty(alex)}, \mathsf{Course(cs48)}, \mathsf{Faculty(jane)}, \mathsf{Course(jane)}, \mathsf{Faculty(cs34)}\}$$

Since $\mathsf{Faculty(cs34)}$ is not involved in any contradictions, it appears in every closed ABox repair, so $\mathsf{cs34}$ is an answer to $q(x) = \mathsf{Faculty}(x)$ under ICAR and CAR semantics. Note however that $\mathsf{cs34}$ is not an answer under AR semantics since some (standard) repairs do not contain $\mathsf{Teaches(cs34, jane)}$, which is required to infer $\mathsf{Faculty(cs34)}$.

As displayed in Fig. 2, the CAR semantics satisfies Consistent Results, and ICAR semantics further satisfies Unique Base, but neither satisfies Consistent Support (here again we refer to [9] for an example).

We next consider a parameterized family of semantics, called the k-support semantics, that were introduced in [15] in order to provide increasingly more fine-grained lower approximations of the AR semantics (while enjoying certain desirable computational properties, see Section 3).

Definition 8 (k-support semantics). *Tuple* $\boldsymbol{a}$ *is an answer to* q *over* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$ *under the* k-support semantics*, written* $\langle \mathcal{T}, \mathcal{A}\rangle \models_{k\text{-}\mathsf{supp}} q(\boldsymbol{a})$*, if there exist (not necessarily distinct) subsets* $S_1, \ldots, S_k$ *of* $\mathcal{A}$ *that satisfy the following:*

- *each* S_i *is a* $\mathcal{T}$*-support for* $q(\boldsymbol{a})$ *in* $\mathcal{A}$

– *for every $R \in Rep(\mathcal{K})$, there is some S_i with $S_i \subseteq R$*

The intuition for the k-support semantics is to restrict the number of distinct supports of the query that can be used to 'cover' all of the repairs. When $k = 1$, the same support must be present in every repair, so the 1-support semantics coincides with the IAR semantics. By increasing k and allowing larger and larger supports, the set of answers will increase until it coincides with the AR-answers. Like the AR semantics, the k-support semantics satisfy both Consistent Results and Consistent Support.

Example 3. Returning to Example 1, we evaluate $\mathsf{Faculty}(x)$ using the k-support semantics. When $k = 1$, the semantics coincides with IAR, so we only get alex. For $k = 2$, we gain an additional answer, kim, by considering the pair of supports $\{\mathsf{Prof}(\mathsf{kim})\}$ and $\{\mathsf{Lect}(\mathsf{kim})\}$. Finally, for $k \geq 3$, we have one further answer, sam, by considering the supports $\{\mathsf{Prof}(\mathsf{sam})\}$, $\{\mathsf{Lect}(\mathsf{sam})\}$, $\{\mathsf{Fellow}(\mathsf{sam})\}$.

A second parameterized class of semantics, the k-defeater semantics, was introduced in the same work [15] in order to provide increasingly tighter upper approximations of the AR semantics.

Definition 9 (k-defeater semantics). *A tuple $\boldsymbol{a}$ is an answer to q over $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ under the* k-defeater semantics, *written $\mathcal{K} \models_{k\text{-}\mathsf{def}} q(\boldsymbol{a})$, if there does not exist a $\mathcal{T}$-consistent subset S of $\mathcal{A}$ with $|S| \leq k$ such that $\langle \mathcal{T}, S \cup C \rangle \models \bot$ for every inclusion-minimal $\mathcal{T}$-support $C \subseteq \mathcal{A}$ of $q(\boldsymbol{a})$.*

It can be shown that 0-defeater semantics coincides with brave semantics and that the set of answers under k-defeater semantics decreases as the value of k increases, until the set of AR-answers is reached.

We should also mention another parameterized family of inconsistency-tolerant semantics, called k-lazy [24], whose definition involves another notion of repair and will be omitted for lack of space. By taking k large enough, the k-lazy semantics coincides with the AR semantics. However, in contrast to the k-support semantics, the convergence is not monotone, meaning that a tuple might be an answer for $k = \ell$ but no longer an answer when $k = \ell + 1$. Due to this behaviour, the k-lazy semantics are not always under-approximations of AR semantics, though they do satisfy Consistent Results and Consistent Support.

More recently, a new upper approximation of the AR semantics has been proposed, called non-objection inference [4]:

Definition 10. *A tuple $\boldsymbol{a}$ is an answer to q over $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ under the* non-objection (no) semantics *if (i) there is some $\mathcal{B} \in Rep(\mathcal{K})$ with $\langle \mathcal{T}, \mathcal{B} \rangle \models q(\boldsymbol{a})$, and (ii) for every $\mathcal{B} \in Rep(\mathcal{K})$, there is a model of $\langle \mathcal{T}, \mathcal{B} \rangle$ where $q(\boldsymbol{a})$ is satisfied.*

The non-objection semantics lies between the brave and AR semantics, and it satisfies both Consistent Results and Consistent Support. Two further variants are considered in [4], by focusing on cardinality-maximal repairs.

As noted at the beginning of the section, repairs are usually defined using set inclusion. However, in some cases, it can be more appropriate to work with cardinality-maximal repairs, or to select only the most preferred repairs according to some criteria. Several different notions of preferred repair, based cardinality, priority levels or weighted assertions, were explored in [10] and used to define variants of the IAR and AR semantics.

We further remark that the preceding works focused on repairing data given in the form of an ABox, and the definitions need to be adapted to handle the setting of ontology-based data access (OBDA), where existing data sources are linked to a TBox via mappings. This issue is explored in a recent paper [8], where two different approaches (repair-at-source and map-then-repair) are contrasted.

Finally, we should emphasize that there is no single 'best' semantics, and the choice of which to use needs to be based upon the acceptable level of risk and performance requirements. Moreover, it can be fruitful to utilize multiple semantics in combination, either for computational benefit or to identify answers with different levels of reliability.

3 Complexity of Inconsistency-Tolerant Querying

The complexity of query answering under inconsistency-tolerant semantics has been the subject of numerous works. We briefly present what is known and refer to [9] for further details and full references.

Let us start by the most well-studied case, namely, DL-Lite KBs[1]. Figure 3 displays the complexity landscape for querying DL-Lite KBs under a range of inconsistency-tolerant semantics, considering both data and combined complexity and both conjunctive queries (CQs) and instance queries (IQs). We observe that there are several semantics for which query answering is in AC^0 in data complexity. These upper bounds are shown by means of first-order query rewriting. For the IAR semantics, the rough idea is to modify a usual rewriting by adding negated atoms that forbid the use of ABox assertions that do not belong to the intersection of repairs [22, 6]. Subsequent work [15] established general rewritability results that apply to the families of k-support and k-defeater semantics and arbitrary FO-rewritable ontology languages. We note that the AC^0 result for non-objection semantics[2] has not been stated in the literature but can be shown by adapting query rewriting techniques for the brave and IAR semantics. For the AR semantics, which is arguably the most natural, query answering is intractable in data complexity, even in simpler settings, like IQs [21] or very simple TBoxes (a single axiom $T \sqsubseteq F$ suffices [7]). Looking now to combined complexity, we observe that the semantics that are well behaved for data complexity remain so for combined complexity (i.e. their complexity matches that of classical semantics), while the semantics with intractable data complexity exhibit higher combined complexities than classical semantics. Finally, we note that the complexity of querying with variants of AR and IAR based upon

[1] The presented results apply to the most common DL-Lite dialects, such as DL-Lite$_{\mathsf{core}}$, DL-Lite$_{\mathcal{R}}$, and DL-Lite$_{\mathcal{A}}$, see [9] for details.

[2] In [4], only polynomial data complexity is proven, which we improve to AC^0. It is also not too hard to show that the combined complexity matches classical semantics.

Semantics	**Data complexity**		**Combined complexity**	
	CQs	**IQs**	**CQs**	**IQs**
classical	in AC^0	in AC^0	NP	NL
AR	coNP	coNP	Π_2^p	coNP
IAR	in AC^0	in AC^0	NP	NL
brave	in AC^0	in AC^0	NP	NL
CAR	coNP	in AC^0	Π_2^p	NL
ICAR	in AC^0	in AC^0	NP	NL
ICR	coNP	coNP	$\Delta_2^p[O(log\, n)]$	coNP
k**-support** $(k \geq 1)$	in AC^0	in AC^0	NP	NL
k**-defeater** $(k \geq 0)$	in AC^0	in AC^0	NP	NL
k**-lazy** $(k \geq 1)$	coNP	in P	Π_2^p	in P
non-objection	in AC^0	in AC^0	NP	NL

Fig. 3: Complexity of inconsistency-tolerant query answering in DL-Lite. All results are completeness results unless otherwise indicated.

DL	**Semantics**	**Data complexity**		**Combined complexity**	
		CQs	**IQs**	**CQs**	**IQs**
$\mathcal{EL}_\bot$	**classical**	P	P	NP	P
	AR	coNP	coNP	Π_2^p	coNP
	IAR	coNP	coNP	$\Delta_2^p[O(log\, n)]$	coNP
	brave	NP	NP	NP	NP
$\mathcal{ALC}$	**classical**	coNP	coNP	Exp	Exp
	AR	Π_2^p	Π_2^p	Exp	Exp
	IAR	Π_2^p	Π_2^p	Exp	Exp
	brave	Σ_2^p	Σ_2^p	Exp	Exp

Fig. 4: Complexity of inconsistency-tolerant query answering in $\mathcal{EL}_\bot$ and $\mathcal{ALC}$. All results are completeness results unless otherwise indicated.

preferred repairs (cardinality, weights, priorities) has also been studied [10], and the general message is that incorporating preferences leads to higher complexity.

We now briefly consider the situation for DLs beyond DL-Lite. Figure 4 displays complexity results for two representative DLs ($\mathcal{EL}_\bot$ and $\mathcal{ALC}$) and three prominent semantics (AR, IAR, brave). The main observation with regards to $\mathcal{EL}_\bot$ (and other Horn DLs) is that the IAR and brave semantics are no longer tractable in data complexity. Essentially, the reason is that in constrast to DL-Lite and other FO-rewritable languages, it is not possible in general to bound the size of minimal $\mathcal{T}$-inconsistent subsets nor minimal $\mathcal{T}$-supports. For $\mathcal{ALC}$, the adoption of inconsistency-tolerant semantics leads to a rise in data complexity, but leaves the combined complexity unchanged (since the repairs can be enumerated in exponential time).

4 Implementations of Inconsistency-Tolerant Querying

We provide a brief overview of systems that have been implemented and tested for inconsistency-tolerant query answering over DL knowledge bases.

QuID system [26, 23] This system[3] performs conjunctive query answering under the IAR semantics in an extension of DL-Lite$_\mathcal{A}$ with denial and identification constraints. Three different approaches have been implemented and compared: first-order query rewriting, ABox annotation (in which assertions are marked as safe or problematic depending on whether they belong to the intersection of repairs, and the query is modified to only use safe assertions), and ABox cleaning (in which assertions not belonging to the intersection of repairs are removed, and the resulting dataset is queried as usual). The latter two approaches generally proved to be more efficient than the rewriting approach, but they have the downside of involving data modifications.

CQAPri system [10, 13] This system[4] computes answers to CQs over DL-Lite$_\mathcal{R}$ KBs under the IAR, brave, and AR semantics (as well as prioritized versions of AR and IAR). Answers are first computed for the IAR and brave semantics, by evaluating a UCQ-rewriting and filtering the results using a pre-computed set of minimal $\mathcal{T}$-inconsistent subsets. To identify the AR-answers among the remaining tuples (i.e. those holding under brave semantics but not under IAR semantics), CQAPri constructs a (usually quite small) instance of UNSAT for every such tuple, which is passed to an off-the-shelf SAT solver. In addition to using the IAR and brave semantics to reduce the number of calls to the SAT solver, the three semantics are used to partition query answers into three levels of reliability: (Almost) Sure (those answers holding under IAR semantics), Likely (answers holding under AR but not IAR semantics), and Possible (answers only holding under brave semantics). Experiments conducted on the modified LUBM benchmark (which was further augmented with negative inclusions and conflicting assertions) showed that despite its intractable data complexity, it is feasible to compute query answers under the AR semantics, thanks in part to the fact that many AR-answers can be identifed using the tractable IAR semantics.

SaQAI system [28] This system[5] implements the IAR and ICAR semantics for DL-Lite$_\mathcal{R}$ KBs and CQs. For the IAR semantics, the authors follow the ABox cleaning approach from QuID, using query rewriting to identify then remove the assertions that do not appear in the intersection of repairs. For the ICAR semantics, a combination of saturation and query rewriting is employed, together with some optimizations. The experiments conducted using the CQAPri benchmark show a better performance than QuID and CQAPri for the IAR semantics.

[3] QuID system: `http://www.dis.uniroma1.it/~ruzzi/quid/`
[4] CQAPri system and benchmark: `https://www.lri.fr/~bourgaux/CQAPri`.
[5] SaQAI system: `http://www.image.ece.ntua.gr/~etsalap/SaQAI/`

System from [27] This system targets the IAR semantics and currently supports the DL $\mathcal{ELH}^{dr}_{\bot}$. It checks whether sufficient conditions for producing an IAR-rewriting are fulfilled (leveraging the FO-rewritability checker Grind [19]) and constructs such a rewriting when one exists by adding negated conjuncts to a classical rewriting. Experiments were conducted on seven existing ontologies (which sometimes needed to be enriched with negative inclusions) and for six of them, the sufficient conditions were satisfied, suggesting that a rewriting-based approach to IAR may be feasible in practice for ontologies beyond DL-Lite.

System from [4] This system implements the non-objection semantics for ground CQs (i.e. CQs without existentially quantified variables) for DL-Lite$_{\mathcal{R}}$ KBs. Experiments on the CQAPri benchmark confirm that query answers can be efficiently computed (in accordance with the tractable data complexity).

System from[18] This system can be used to query $\mathcal{SHIQ}$ KBs under a variant of the AR semantics with weight on ABox assertions and is restricted to ground CQs. Like CQAPri, it employs SAT solvers, and a form of reachability analysis to identify a query-relevant fragment of the KB.

5 Related Reasoning Services for Inconsistency Handling

We mention some related reasoning and analysis tasks. First, to render inconsistency-tolerant OMQA systems more usable, it is important to be able to explain the results to users. This issue has been taken up in [11], where a formal framework was presented for justifying why a given tuple appears as an answer under the considered inconsistency-tolerant semantics (AR, IAR, or brave) or why it is not part of the results. The approach has been implemented by exploiting different functionalities of SAT solvers and integrated into the CQAPri system. Closely related is a line of work [3, 2] on utilizing argumentation and dialogues with users to explain query answers under various inconsistency-tolerant semantics (ICR, IAR, brave, and AR).

Another important question is how to aid users in repairing their data, in order to improve the quality of the data. An interactive query-driven approach to this question has been presented in [12]. The idea is to allow users may provide feedback on which query results are missing or erroneous, and then interact with the user in order to identify a set of ABox modifications (additions and deletions of assertions) that fix the identified flaws.

Finally, let us mention a formal study of the consistency properties of OBDA specifications [17], in which a database schema is said to protect an OBDA specification if every legal data instance for the database constraints is consistent w.r.t. the ontology and mappings, and it said to be faithful to a specification if the database constraints do not exclude any instance that is allowed by the ontology and mappings. (Un)decidability and complexity results are shown for the analysis tasks of checking the protection and faithfulness properties.

6 Concluding Remarks

We hope to have showcased the large body of research that has been developed over the past decade or so around the issue of inconsistency handling in OMQA. Significant progress has been made on proposing different semantics for querying inconsistent KBs in a principled manner and exploring their computational properties: complexity, algorithms, and implemented prototypes. There remain several interesting challenges to tackle going forward, let us mention just three.

First, while we start to have a reasonable idea of how to approach the issue for DL-Lite KBs, there remains a need to develop practical algorithms for DLs beyond DL-Lite. Indeed, due to the prevalence of data quality issues, *every OMQA system should be equipped with some sort of inconsistency handling mechanism* (beyond simply reporting that the KB is inconsistent!), and the challenge is to find ways of incorporating such features while limiting the impact on performance. Some first steps towards this goal can be found in [28, 27].

Second, a very nice but extremely challenging theoretical question is to classify the complexity of inconsistency-tolerant query answering at the level of ontology-mediated queries (that is, ontology-query pairs). Some preliminary results in this direction have been presented in [6, 7]. We note that this problem is closely related to work on classifying the complexity of consistent query answering in the presence of functional dependencies, where significant progress has been made (see e.g. [20]), but a full classification has proven elusive.

Third, it would also be worthwhile to develop quantitative approaches to inconsistency-tolerant OMQA, both to be able to quantify the confidence in different results, and to be able to take advantage of numeric / probabilistic / statistical information when it is available. For instance, data that results from information extraction systems is often annotated with a confidence value, and mined data quality rules (see e.g. [25]) that act as soft constraints can prove useful in detecting inconsistencies and determining the most likely fixes.

Acknowledgements Let me take this opportunity to thank the steering committee and workshop chairs for asking me to be an invited speaker, and while I'm at it, a big thanks to my many co-authors and the DL community at large for providing a scientifically stimulating and friendly research environment.

References

1. Arenas, M., Bertossi, L.E., Chomicki, J.: Consistent query answers in inconsistent databases. In: Proceedings of the 18th Symposium on Principles of Database Systems (PODS). pp. 68–79. ACM Press (1999)
2. Arioua, A., Croitoru, M.: Dialectical characterization of consistent query explanation with existential rules. In: Proceedings of FLAIRS (2016)
3. Arioua, A., Tamani, N., Croitoru, M.: Query answering explanation in inconsistent datalog +/- knowledge bases. In: Proceedings of DEXA (2015)
4. Benferhat, S., Bouraoui, Z., Croitoru, M., Papini, O., Tabia, K.: Non-objection inference for inconsistency-tolerant query answering. In: Proceedings of the 25th

International Joint Conference on Artificial Intelligence (IJCAI). pp. 3684–3690 (2016)
5. Bertossi, L.E.: Database Repairing and Consistent Query Answering. Synthesis Lectures on Data Management, Morgan & Claypool Publishers (2011)
6. Bienvenu, M.: First-order expressibility results for queries over inconsistent dl-lite knowledge bases. In: Rosati, R., Rudolph, S., Zakharyaschev, M. (eds.) Proceedings of the 24th International Workshop on Description Logics (DL) (2011)
7. Bienvenu, M.: On the complexity of consistent query answering in the presence of simple ontologies. In: Proceedings of AAAI (2012)
8. Bienvenu, M.: Inconsistency-tolerant ontology-based data access revisited: Taking mappings into account. In: Lang, J. (ed.) Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI). pp. 1721–1729 (2018)
9. Bienvenu, M., Bourgaux, C.: Inconsistency-tolerant querying of description logic knowledge bases. In: Reasoning Web: Logical Foundation of Knowledge Graph Construction and Query Answering - 12th International Summer School 2016, Aberdeen, UK, September 5-9, 2016, Tutorial Lectures. pp. 156–202 (2016)
10. Bienvenu, M., Bourgaux, C., Goasdoué, F.: Querying inconsistent description logic knowledge bases under preferred repair semantics. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI) (2014)
11. Bienvenu, M., Bourgaux, C., Goasdoué, F.: Explaining inconsistency-tolerant query answering over description logic knowledge bases. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI) (2016)
12. Bienvenu, M., Bourgaux, C., Goasdoué, F.: Explaining inconsistency-tolerant query answering over description logic knowledge bases. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI) (2016)
13. Bienvenu, M., Bourgaux, C., Goasdoué, F.: Computing and explaining query answers over inconsistent dl-lite knowledge bases. J. Artif. Intell. Res. **64**, 563–644 (2019)
14. Bienvenu, M., Rosati, R.: Tractable approximations of consistent query answering for robust ontology-based data access. In: Proceedings of IJCAI (2013)
15. Bienvenu, M., Rosati, R.: Tractable approximations of consistent query answering for robust ontology-based data access. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI) (2013)
16. Chomicki, J.: Consistent query answering: Five easy pieces. In: Proceedings of ICDT. pp. 1–17 (2007)
17. Console, M., Lenzerini, M.: Data quality in ontology-based data access: The case of consistency. In: Brodley, C.E., Stone, P. (eds.) Proceedings of the 28th AAAI Conference on Artificial Intelligence. pp. 1020–1026 (2014)
18. Du, J., Qi, G., Shen, Y.D.: Weight-based consistent query answering over inconsistent $\mathcal{SHIQ}$ knowledge bases. Knowledge and Information Systems **34**(2), 335–371 (2013)
19. Hansen, P., Lutz, C., Seylan, I., Wolter, F.: Efficient query rewriting in the description logic EL and beyond. In: Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI). pp. 3034–3040 (2015)
20. Koutris, P., Wijsen, J.: Consistent query answering for self-join-free conjunctive queries under primary key constraints. ACM Trans. Database Syst. **42**(2), 9:1–9:45 (2017)
21. Lembo, D., Lenzerini, M., Rosati, R., Ruzzi, M., Savo, D.F.: Inconsistency-tolerant semantics for description logics. In: Proceedings of the 4th International Conference on Web Reasoning and Rule Systems (RR) (2010)

22. Lembo, D., Lenzerini, M., Rosati, R., Ruzzi, M., Savo, D.F.: Query rewriting for inconsistent DL-Lite ontologies. In: Proceedings of the 5th International Conference on Web Reasoning and Rule Systems (RR) (2011)
23. Lembo, D., Lenzerini, M., Rosati, R., Ruzzi, M., Savo, D.F.: Inconsistency-tolerant query answering in ontology-based data access. Journal Web Sem. **33**, 3–29 (2015)
24. Lukasiewicz, T., Martinez, M.V., Simari, G.I.: Inconsistency handling in datalog+/- ontologies. In: Proceedings of ECAI (2012)
25. Ortona, S., Meduri, V.V., Papotti, P.: Rudik: Rule discovery in knowledge bases. PVLDB **11**(12), 1946–1949 (2018)
26. Rosati, R., Ruzzi, M., Graziosi, M., Masotti, G.: Evaluation of techniques for inconsistency handling in OWL 2 QL ontologies. In: Proceedings of ISWC (2012)
27. Trivela, D., Stoilos, G., Vassalos, V.: A framework and positive results for iar-answering. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence. pp. 1973–1980 (2018)
28. Tsalapati, E., Stoilos, G., Stamou, G.B., Koletsos, G.: Efficient query answering over expressive inconsistent description logics. In: Kambhampati, S. (ed.) Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI). pp. 1279–1285 (2016)