

Inconsistency-Tolerant Querying of Description Logic Knowledge Bases*

Meghyn Bienvenu¹ and Camille Bourgaux²

¹ LIRMM - CNRS & Université de Montpellier & INRIA

`meghyn@lirmm.fr`

² LRI - CNRS & Université Paris Sud

`bourgaux@lri.fr`

Abstract. An important issue that arises when querying description logic (DL) knowledge bases is how to handle the case in which the knowledge base is inconsistent. Indeed, while it may be reasonable to assume that the TBox (ontology) has been properly debugged, the ABox (data) will typically be very large and subject to frequent modifications, both of which make errors likely. As standard DL semantics is useless in such circumstances (everything is entailed from a contradiction), several alternative inconsistency-tolerant semantics have been proposed with the aim of providing meaningful answers to queries in the presence of such data inconsistencies. In the first part of this chapter, we present and compare these inconsistency-tolerant semantics, which can be applied to any DL (or ontology language). The second half of the chapter summarizes what is known about the computational properties of these semantics and gives an overview of the main algorithmic techniques and existing systems, focusing on DLs of the DL-Lite family.

* This work has been supported by ANR project PAGODA (ANR-12-JS02-007-01).

Table of Contents

Inconsistency-Tolerant Querying of Description Logic Knowledge Bases . . .	1
<i>Meghyn Bienvenu and Camille Bourgaux</i>	
1 Introduction	2
2 Preliminaries	4
2.1 Description Logic Knowledge Bases	4
2.2 Querying DL KBs	6
2.3 Query Rewriting	7
2.4 Complexity of Query Answering	8
3 Inconsistency-Tolerant Semantics	9
3.1 Repairs	11
3.2 AR Semantics	12
3.3 IAR Semantics	14
3.4 Brave Semantics	15
3.5 k -Support and k -Defeater Semantics	16
3.6 ICR Semantics	19
3.7 CAR and ICAR Semantics	20
3.8 k -Lazy Semantics	23
3.9 Preferred Repair Semantics	25
3.10 Summary and Discussion	28
4 Complexity of Inconsistency-Tolerant Query Answering	29
4.1 Complexity Landscape for DL-Lite	29
4.2 Beyond DL-Lite	40
5 Systems for Inconsistency-Tolerant Query Answering	42
5.1 The QuID System	43
5.2 The CQAPri System	44
6 Summary and Outlook	45

1 Introduction

Ontology-mediated query answering (OMQA) is a promising approach to data access that leverages the semantic knowledge provided by an ontology to improve query answering (see [12] for a recent survey and references). Much of the work on OMQA considers focuses on ontologies formulated using description logics (DLs) [5], a well-known class of decidable fragments of first-order logic that provide the logically underpinnings of the W3C-standardized OWL 2 ontology language [40]. Over the past decade, significant research efforts have been devoted to understanding how the complexity of OMQA varies depending on the choice of DL and query language, which has led to the identification of DLs with favourable computational properties. The DL-Lite family of DLs [15], which was specifically designed with OMQA in mind, has gained particular prominence, due to the fact that query answering can be reduced via first-order query rewriting to

database query evaluation. In addition to theoretical work, there has been a lot of practically-oriented research aimed at developing and implementing efficient algorithms for OMQA, particularly for DL-Lite ontologies.

An important practical issue that arises in the context of DL-based OMQA is how to handle the case in which the dataset (or ABox, in DL parlance) is inconsistent with the ontology (TBox). Indeed, while it may be reasonable to assume that the TBox has been properly debugged, the ABox will typically be very large and/or subject to frequent modifications, which makes errors likely. Unfortunately, standard DL semantics is next to useless in such circumstances, as everything is entailed from a contradiction. It is therefore essential to devise robust methods for handling inconsistent data if OMQA is to be widely adopted in practice. Modifying the ABox to restore consistency may seem like the ideal solution, as it allows us to use existing query answering algorithms. However, this approach is not always feasible. First, it can be difficult and time-consuming to identify the erroneous parts of the data, and removing all potentially erroneous assertions will typically lead to unacceptable loss of information. Second, even if it can be determined which ABox assertions should be removed, the OMQA system may lack the authorization to make data modifications (e.g., in information integration applications involving external data sources).

As inconsistencies cannot always be eliminated, it is important to provide principled methods for obtaining meaningful answers to queries posed over inconsistent DL KBs. In this chapter, we will present a number of different ways of approaching this problem, based upon using different kinds of inconsistency-tolerant semantics for defining what tuples should be counted as query answers. Probably the most well-known and arguably the most natural such semantics is the AR semantics, introduced in [32]. The semantics is inspired by work on consistent query answering in relational databases [1, 2, 17, 7], where the standard approach is to define a set of data repairs (which correspond to the different ways of minimally changing the dataset to restore consistency), and to define the set of query results as those answers that can be obtained from each of the data repairs. In the DL setting, an analogous notion of ABox repair can be defined by considering the inclusion-maximal subsets of the ABox that are consistent w.r.t. the TBox. The AR semantics amounts to considering those answers that can be derived from each of the ABox repairs using the axioms in the TBox. As we shall see in this chapter, there are in fact many other inconsistency-tolerant semantics that can be defined in terms of ABox repairs, such as the IAR semantics [32], which defines query answers w.r.t. to the intersection of the ABox repairs, and the brave semantics [13] that only requires that an answer hold w.r.t. at least one repair.

The aim of this chapter is to provide an introduction to inconsistency-tolerant query answering in the DL setting. After some preliminaries, we will introduce in Section 3 a variety of different inconsistency-tolerant semantics and examine the relationships that hold between the semantics. In the following section, we will explore the computational properties of these semantics, providing a detailed complexity landscape for DLs of the DL-Lite family, as well as a short discussion

of what happens when one considers other DLs. In Section 5, we will briefly describe the systems that have been implemented for inconsistency-tolerant query answering, before concluding in Section 6 with a discussion of current and future research directions.

2 Preliminaries

2.1 Description Logic Knowledge Bases

A DL *knowledge base* (KB) consists of an ABox and a TBox, which are constructed from a set \mathbf{N}_C of *concept names* (unary predicates), a set of \mathbf{N}_R of *role names* (binary predicates), and a set \mathbf{N}_I of *individual names* (constants). The *ABox* (dataset) is a finite set of *concept assertions* of the form $A(a)$, with $A \in \mathbf{N}_C$ and $a \in \mathbf{N}_I$, and *role assertions* of the form $R(a, b)$, with $R \in \mathbf{N}_R, a, b \in \mathbf{N}_I$. The *TBox* (ontology) consists of a finite set of axioms whose form depends on the DL in question.

In this chapter, we will mainly focus on description logics from the DL-Lite family [15]. In the DL-Lite $_{\mathcal{R}}$ dialect (which underlies the OWL 2 QL profile [39]), TBoxes are composed of *concept inclusions* $B \sqsubseteq C$ and *role inclusions* $Q \sqsubseteq S$, where B and C (resp. Q and S) are complex concepts (resp. roles) formed according to the following syntax:

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

where $A \in \mathbf{N}_C$ and $R \in \mathbf{N}_R$. Inclusions that have \neg on the right-hand side are called *negative inclusions*; all other inclusions are called *positive inclusions*. In the DL-Lite $_{\text{core}}$ dialect, only concept inclusions are allowed in the TBox.

The next example introduces the DL-Lite $_{\text{core}}$ TBox that we will use throughout the chapter, which is adapted from the one in [13]:

Example 1. We consider the TBox $\mathcal{T}_{\text{univ}}$ consisting of the following axioms:

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

The inclusions in the first column state that professors, lecturers, and research fellows are three classes of faculty members. Due to the negative inclusions in the third column, we know that these are disjoint classes. In the second column, we state that professors and lecturers must teach something (that is, they must occur in the first argument of relation *Teaches*), and everything that is taught (that is, appears in the second argument of *Teaches*) is of type *Course*. The rightmost axiom states that *Faculty* and *Course* are disjoint.

We will also briefly mention the description logics \mathcal{EL}_{\perp} and \mathcal{ALC} . In both DLs, the TBox consists of concept inclusions $C_1 \sqsubseteq C_2$, where C_1, C_2 are concepts built according to the syntax of the DL. In \mathcal{EL}_{\perp} , we use the grammar:

$$C := A \mid \top \mid \perp \mid C \sqcap C \mid \exists R.C$$

and in \mathcal{ALC} , concepts are constructed as follows:

$$C := A \mid \top \mid \perp \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \exists R.C \mid \forall R.C$$

where as before, $A \in \mathbf{N}_C$ and $R \in \mathbf{N}_R$. Thus, in \mathcal{EL}_\perp , we can build concepts using the top and bottom concepts (\top, \perp), conjunction (\sqcap), and qualified existential restrictions ($\exists R.C$), and in \mathcal{ALC} , we may additionally use unrestricted negation (\neg), disjunction (\sqcup), and qualified universal restrictions ($\forall R.C$).

The semantics of DL knowledge bases is defined using interpretations. An *interpretation* takes the form $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a non-empty set and $\cdot^{\mathcal{I}}$ maps each $a \in \mathbf{N}_I$ to $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, each $A \in \mathbf{N}_C$ to $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and each $R \in \mathbf{N}_R$ to $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The function $\cdot^{\mathcal{I}}$ is straightforwardly extended to general concepts and roles:

$$\begin{aligned} \top^{\mathcal{I}} &= \Delta^{\mathcal{I}} & \perp^{\mathcal{I}} &= \emptyset & (\neg B)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus B^{\mathcal{I}} \\ (C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} & (C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}} \\ (\exists Q)^{\mathcal{I}} &= \{c \mid \exists d : (c, d) \in Q^{\mathcal{I}}\} & (\exists R.C)^{\mathcal{I}} &= \{c \mid \exists d : (c, d) \in R^{\mathcal{I}}, d \in C^{\mathcal{I}}\} \\ (R^-)^{\mathcal{I}} &= \{(c, d) \mid (d, c) \in R^{\mathcal{I}}\} & (\forall R.C)^{\mathcal{I}} &= \{c \mid \forall d : (c, d) \in R^{\mathcal{I}} \Rightarrow d \in C^{\mathcal{I}}\} \end{aligned}$$

An interpretation \mathcal{I} satisfies an inclusion $G \sqsubseteq H$ if $G^{\mathcal{I}} \subseteq H^{\mathcal{I}}$; it satisfies $A(a)$ (resp. $R(a, b)$) if $a^{\mathcal{I}} \in A^{\mathcal{I}}$ (resp. $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$). An interpretation \mathcal{I} is a *model* of $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ if \mathcal{I} satisfies all inclusions in \mathcal{T} and assertions in \mathcal{A} . A KB \mathcal{K} is *consistent* if it has a model, and we say that an ABox \mathcal{A} is \mathcal{T} -*consistent* if the KB $\langle \mathcal{T}, \mathcal{A} \rangle$ is consistent. 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.

Example 2. In our running example, we will consider the following ABox:

$$\begin{aligned} \mathcal{A}_{\text{univ}} &= \{\text{Prof}(\text{sam}), \text{Lect}(\text{sam}), \text{Fellow}(\text{sam}), \text{Prof}(\text{kim}), \text{Lect}(\text{kim}), \\ &\quad \text{Fellow}(\text{julie}), \text{Teaches}(\text{csc343}, \text{julie}), \text{Fellow}(\text{alex}), \text{Teaches}(\text{alex}, \text{csc486})\} \end{aligned}$$

Observe that $\mathcal{A}_{\text{univ}}$ is $\mathcal{T}_{\text{univ}}$ -inconsistent. Indeed, we have the following minimal $\mathcal{T}_{\text{univ}}$ -inconsistent subsets:

- $\{\text{Prof}(\text{sam}), \text{Lect}(\text{sam})\}$ which contradicts $\text{Prof} \sqsubseteq \neg \text{Lect}$
- $\{\text{Prof}(\text{sam}), \text{Fellow}(\text{sam})\}$ which contradicts $\text{Prof} \sqsubseteq \neg \text{Fellow}$
- $\{\text{Lect}(\text{sam}), \text{Fellow}(\text{sam})\}$ which contradicts $\text{Lect} \sqsubseteq \neg \text{Fellow}$
- $\{\text{Prof}(\text{kim}), \text{Lect}(\text{kim})\}$ which contradicts $\text{Prof} \sqsubseteq \neg \text{Lect}$
- $\{\text{Fellow}(\text{julie}), \text{Teaches}(\text{csc343}, \text{julie})\}$ which contradicts $\text{Faculty} \sqsubseteq \neg \text{Course}$, due to the positive inclusions $\text{Fellow} \sqsubseteq \text{Faculty}$ and $\exists \text{Teaches}^- \sqsubseteq \text{Course}$

We say that an assertion or axiom α is *entailed* from a KB \mathcal{K} , written $\mathcal{K} \models \alpha$, if every model of \mathcal{K} satisfies α . If \mathcal{A} is \mathcal{T} -consistent, then the \mathcal{T} -*closure* of \mathcal{A} , denoted $\text{cl}_{\mathcal{T}}(\mathcal{A})$, consists of all assertions α such that $\langle \mathcal{T}, \mathcal{A} \rangle \models \alpha$. Note that since $\langle \mathcal{T}, \mathcal{A} \rangle$ is finite and consistent, the closure $\text{cl}_{\mathcal{T}}(\mathcal{A})$ is necessarily finite.

2.2 Querying DL KBs

We will generally assume that the user query is given as a *conjunctive query* (CQ), which takes the form $q(\mathbf{x}) = \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y})$, where \mathbf{x} and \mathbf{y} are tuples of variables, and ψ is a conjunction of atoms of the forms $A(t)$ or $R(t, t')$, where t, t' are variables from $\mathbf{x} \cup \mathbf{y}$ or individual names. The variables \mathbf{x} are called the *answer variables* of $q(\mathbf{x})$. A CQ without answer variables is called *Boolean*, a CQ without any existentially quantified variables is called a *ground CQ*, and a ground CQ consisting of a single atom is called an *instance query* (IQ).

We say that a Boolean CQ q is *entailed* from \mathcal{K} , written $\mathcal{K} \models q$, just in the case that q holds in all models of \mathcal{K} . For a non-Boolean CQ q with answer variables $\mathbf{x} = (x_1, \dots, x_k)$, a tuple of individuals $\mathbf{a} = (a_1, \dots, a_k)$ is a *certain answer* for q w.r.t. \mathcal{K} just in the case that $\mathcal{K} \models q(\mathbf{a})$, where $q(\mathbf{a})$ is the Boolean query obtained by replacing each x_i by a_i . We will use the notation $\mathcal{K} \models q(\mathbf{a})$ to denote that \mathbf{a} is a certain answer to q w.r.t. \mathcal{K} . Later in the chapter, we will introduce alternative semantics for defining query answers, and we will use ‘classical semantics’ to refer to the certain answer semantics we have just defined.

Example 3. Consider the following queries

$$\begin{aligned} q_1(x) &= \text{Faculty}(x) & q_2(x) &= \exists y \text{Teaches}(x, y) \\ q_3(x) &= \exists y \text{Faculty}(x) \wedge \text{Teaches}(x, y) & q_4(x, y) &= \text{Faculty}(x) \wedge \text{Teaches}(x, y) \end{aligned}$$

If we compute the certain answers to these queries over our example KB $\mathcal{K}_{\text{univ}} = \langle \mathcal{T}_{\text{univ}}, \mathcal{A}_{\text{univ}} \rangle$, then due to the inconsistency of $\mathcal{K}_{\text{univ}}$, every individual appearing in $\mathcal{A}_{\text{univ}}$ will be returned as a certain answer to q_1 , q_2 , and q_3 , and every pair of individuals will be a certain answer for q_4 .

To better illustrate the notion of certain answers, let us consider the following $\mathcal{T}_{\text{univ}}$ -consistent subset of $\mathcal{A}_{\text{univ}}$:

$$\mathcal{A}_{\text{univ}}^{\text{cons}} = \{ \text{Prof}(\text{sam}), \text{Lect}(\text{kim}), \text{Fellow}(\text{julie}), \text{Fellow}(\text{alex}), \text{Teaches}(\text{alex}, \text{csc486}) \}$$

Evaluating the four queries over the KB consisting of $\mathcal{T}_{\text{univ}}$ and the preceding ABox yields the following results:

- $q_1(x)$ has four certain answers: sam, kim, julie, and alex
- $q_2(x)$ has three certain answers: sam, kim, and alex
- $q_3(x)$ has three certain answers: sam, kim, and alex
- $q_4(x, y)$ has a single certain answer: (alex, csc486)

Indeed, for q_1 , each of the individuals sam, kim, julie, and alex belongs to either Prof, Lect, or Fellow, which are declared to be subclasses of Faculty. For q_2 , we obtain sam and kim using the axioms $\text{Prof} \sqsubseteq \exists \text{Teaches}$ and $\text{Lect} \sqsubseteq \exists \text{Teaches}$; note that julie is not an answer to q_2 as the TBox does not guarantee that every Fellow teaches something. The certain answers to q_3 can be obtained by intersecting the certain answers of q_1 and q_2 . Finally, for q_4 , there are no answers involving sam and kim, since although these individuals are known to teach some course, there is no information in the KB that allows us to identify the course(s) taught.

2.3 Query Rewriting

First-order (FO) query rewriting is an algorithmic technique that allows us to reduce ontology-mediated query answering to the evaluation of first-order (\sim SQL) queries, which can be handled by relational database systems. The idea is as follows: the user query is first transformed (independently of the ABox) into an FO-query that incorporates the relevant information from the TBox, and in second step, the resulting query is evaluated over the ABox, viewed as a database.

Formally, given a CQ $q(\mathbf{x})$ and a TBox \mathcal{T} , we call an FO-query $q'(\mathbf{x})$ an *(FO)-rewriting* of $q(\mathbf{x})$ w.r.t. \mathcal{T} if the following equivalence holds for every ABox \mathcal{A} and every tuple of individuals \mathbf{a} (of the same arity as \mathbf{x}):

$$\mathcal{T}, \mathcal{A} \models q(\mathbf{a}) \quad \Leftrightarrow \quad \mathcal{I}_{\mathcal{A}} \models q'(\mathbf{a})$$

where $\mathcal{I}_{\mathcal{A}}$ is the finite interpretation isomorphic to \mathcal{A} , i.e., the domain consists of the individuals occurring in \mathcal{A} and every concept or role name P is interpreted by $\{\mathbf{a} \mid P(\mathbf{a}) \in \mathcal{A}\}$. Note that the symbol \models is used differently on the two sides of the equivalence. On the left, we are checking whether $q(\mathbf{a})$ is entailed from the KB $\langle \mathcal{T}, \mathcal{A} \rangle$, which requires us to consider all models of the KB, whereas on the right, we only need to test whether the FO-sentence $q'(\mathbf{a})$ holds in a single interpretation, $\mathcal{I}_{\mathcal{A}}$.

We recall that DL-Lite $_{\mathcal{R}}$ (like most DL-Lite dialects) possesses the *FO-rewritability* property, meaning that for every CQ q and every DL-Lite $_{\mathcal{R}}$ TBox \mathcal{T} , we can effectively construct an FO-rewriting q' of q w.r.t. \mathcal{T} . Many of the rewriting algorithms developed for DL-Lite produce rewritings that belong to the more restricted class of *union of conjunctive queries* (UCQs), which are disjunctions of CQs $q_1(\mathbf{x}) \vee \dots \vee q_n(\mathbf{x})$ having the same answer variables \mathbf{x} . In this case, we speak of UCQ-rewritings.

Example 4. The following queries

$$\begin{aligned} q'_1(x) &= \text{Faculty}(x) \vee \text{Prof}(x) \vee \text{Lect}(x) \vee \text{Fellow}(x) \\ q'_2(x) &= \exists y. \text{Teaches}(x, y) \vee \text{Prof}(x) \vee \text{Lect}(x) \\ q'_3(x) &= (\exists y. \text{Faculty}(x) \wedge \text{Teaches}(x, y)) \vee \text{Prof}(x) \vee \text{Lect}(x) \vee \\ &\quad (\exists y. \text{Fellow}(x) \wedge \text{Teaches}(x, y)) \\ q'_4(x, y) &= (\text{Faculty}(x) \wedge \text{Teaches}(x, y)) \vee (\text{Prof}(x) \wedge \text{Teaches}(x, y)) \\ &\quad (\text{Lect}(x) \wedge \text{Teaches}(x, y)) \vee (\text{Fellow}(x) \wedge \text{Teaches}(x, y)) \end{aligned}$$

are UCQ-rewritings, respectively, of the queries $q_1(x)$, $q_2(x)$, $q_3(x)$, and $q_4(x, y)$ w.r.t. the TBox $\mathcal{T}_{\text{univ}}$. Observe that the disjuncts of q'_i correspond to all of the ways that a (pair of) individual(s) can be derived as a certain answer. The query $q'_2(x)$, for example, states that an individual is a certain answer to $q_2(x) = \exists y \text{Teaches}(x, y)$ if the individual appears in the first argument of a **Teaches** assertion, or if it appears in a **Prof** assertion, or if it appears in a **Lect** assertion.

If we evaluate the rewriting q'_i over $\mathcal{I}_{\mathcal{A}_{\text{univ}}^{\text{cons}}}$, then we will obtain the certain answers of q_i over $\langle \mathcal{T}_{\text{univ}}, \mathcal{A}_{\text{univ}}^{\text{cons}} \rangle$. For example, evaluating $q'_2(x)$ over $\mathcal{I}_{\mathcal{A}_{\text{univ}}^{\text{cons}}}$ returns

the following three answers: **sam** (due to the disjunct $\text{Prof}(x)$), **kim** (due to the disjunct $\text{Lect}(x)$), and **alex** (due to the disjunct $\exists y.\text{Teaches}(x, y)$).

2.4 Complexity of Query Answering

As usual, when we speak of the complexity of ontology-mediated query answering, we mean the computational complexity of the associated *decision problem*, which is to determine whether $\langle \mathcal{T}, \mathcal{A} \rangle \models q(\mathbf{a})$ (here \mathbf{a} is a candidate answer, i.e. tuple of individuals of the same arity as q). There are two standard ways of measuring the complexity of query answering:

- *Combined complexity*: we measure the complexity in terms of the size of the entire input (\mathcal{T} , \mathcal{A} , and $q(\mathbf{a})$)
- *Data complexity*: we measure the complexity only in terms of the size of the ABox \mathcal{A} (so $|\mathcal{T}|$ and $|q(\mathbf{a})|$ are treated as fixed constants)

Data complexity is often considered the more relevant measure as the ABox (data) is typically significantly larger than the size of the TBox and query. However, combined complexity is often more fine-grained, allowing us to distinguish between different problems with the same data complexity. By considering both measures, we get a more complete picture of the complexity landscape.

In our complexity analysis, we will refer to the following complexity classes:

- AC^0 : problems that can be solved by a uniform family of circuits of constant depth and polynomial size, with unlimited fan-in AND gates and OR gates
- NL: problems solvable in non-deterministic logarithmic space
- coNL: problems whose complement is solvable in non-deterministic logarithmic space
- P: problems solvable in polynomial time
- NP: problems solvable in non-deterministic polynomial time
- coNP: problems whose complement is in NP
- Δ_2^p : problems solvable in polynomial time with access to an NP oracle
- $\Delta_2^p[O(\log n)]$: problems solvable in polynomial time with at most logarithmically many calls to an NP oracle
- Σ_2^p : problems solvable in non-deterministic polynomial time with access to an NP oracle
- Π_2^p : problems whose complement is in Σ_2^p
- EXP: problems solvable in deterministic single-exponential time

We recall that these complexity classes are related as follows:

$$\text{AC}^0 \subseteq \text{NL} \subseteq \text{P} \begin{array}{l} \subseteq \text{NP} \\ \subseteq \text{coNP} \end{array} \subseteq \Delta_2^p[O(\log n)] \subseteq \Delta_2^p \begin{array}{l} \subseteq \Sigma_2^p \\ \subseteq \Pi_2^p \end{array} \subseteq \text{EXP}$$

and that $\text{NL}=\text{coNL}$ [27, 44].

3 Inconsistency-Tolerant Semantics

Classical semantics does not allow us to obtain meaningful answers to queries posed over inconsistent DL knowledge bases. Indeed, if we have two inconsistent KBs \mathcal{K}_1 and \mathcal{K}_2 sharing the same set of individuals, then for every n -ary query, we will obtain the same set of certain answers, namely the set of all n -tuples of individuals appearing in these KBs. This is clearly undesirable, as the next example illustrates:

Example 5. Consider the ABoxes \mathcal{A}_1 and \mathcal{A}_2 defined as follows:

$$\begin{aligned}\mathcal{A}_1 &= \{\text{Prof}(\text{sam}), \text{Lect}(\text{sam}), \text{Fellow}(\text{alex})\} \\ \mathcal{A}_2 &= \{\text{Prof}(\text{sam}), \text{Fellow}(\text{alex}), \text{Lect}(\text{alex})\}\end{aligned}$$

Observe that the KBs $\langle \mathcal{T}_{\text{univ}}, \mathcal{A}_1 \rangle$ and $\langle \mathcal{T}_{\text{univ}}, \mathcal{A}_2 \rangle$ are both inconsistent and use the same set of individuals, so they will be treated identically under classical semantics. However, we naturally draw different conclusions from these two KBs. Indeed, from the first KB, it seems reasonable to infer that **Fellow(alex)** since the assertion **Fellow(alex)** is not involved in any contradictions. By contrast, there is conflicting information about the professional status of **sam** (is **sam** a professor or lecturer?), so we would probably choose not to return **sam** as an answer to a query asking for professors (similarly for lecturers). We might nevertheless conclude that **sam** is a faculty member (**Faculty(sam)**) and that **sam** teaches ($\exists y \text{Teaches}(\text{sam}, y)$), since all of the information about the individual **sam** supports these statements. Applying a similar reasoning on the second KB, one would naturally return **sam** as an answer to the query **Prof(x)** (since the ABox asserts **Prof(sam)** and contains no information contradicting this assertion). However, for the individual **alex**, we would probably conclude that **Faculty(alex)** but would not say more about the specific position **alex** occupies (nor whether **alex** teaches).

Several inconsistency-tolerant semantics have been proposed to overcome the inadequacy of classical semantics for querying inconsistent KBs. Each of these semantics provides a different way of defining the set of answers of a query over a (possibly inconsistent) KB. In what follows, we will use the notation $\mathcal{K} \models_S q(\mathbf{a})$ to denote that \mathbf{a} is an answer to q over the KB \mathcal{K} under semantics S . If no semantics S is indicated, then we mean the classical semantics.

In general, there is no single ‘best’ inconsistency-tolerant semantics, and indeed, it may be interesting to use multiple semantics together, e.g. to identify answers of different levels of confidence. In order to choose which (combinations of) semantics to use in a given application, it is important to understand the properties of the different semantics, as well as the relationships between them.

We propose two notions of consistency for an inconsistency-tolerant semantics. The first one requires that all answers returned by the semantics have an internally consistent justification. Formally, we say that a set $C \subseteq \mathcal{A}$ is a (*consistent*) \mathcal{T} -support of $q(\mathbf{a})$ if (i) C is \mathcal{T} -consistent, and (ii) $\langle \mathcal{T}, C \rangle \models q(\mathbf{a})$. The consistency condition can then be defined as follows:

Definition 1 (Consistent Support Property). A semantics S is said to satisfy the CONSISTENT SUPPORT property if for every KB $\langle \mathcal{T}, \mathcal{A} \rangle$, query q , and tuple \mathbf{a} , if $\langle \mathcal{T}, \mathcal{A} \rangle \models_S q(\mathbf{a})$, then there exists a \mathcal{T} -support $C \subseteq \mathcal{A}$ of $q(\mathbf{a})$.

The CONSISTENT SUPPORT property is important for explaining query results to users, as it ensures that we can always extract a consistent subset of the KB which yields a given query answer. Indeed, one could argue that CONSISTENT SUPPORT is a minimal requirement for an inconsistency-tolerant semantics to be meaningful.

Our second notion of consistency requires that the set of all query results obtainable from a KB are (simultaneously) consistent with the TBox:

Definition 2 (Consistent Results Property). A semantics S is said to satisfy 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 $\mathcal{I} \models q(\mathbf{a})$ for every $q(\mathbf{a})$ with $\langle \mathcal{T}, \mathcal{A} \rangle \models_S q(\mathbf{a})$.

The CONSISTENT RESULTS property means that users can safely combine the query results together without any risk of obtaining a contradiction. If a semantics does not satisfy this property, then users need to be made aware of this (e.g. by informing users that the returned results should be considered as ‘potential answers’).

We point out that neither of the consistency properties implies the other. Indeed, we will see examples of semantics that satisfy CONSISTENT SUPPORT but not CONSISTENT RESULTS, and other semantics that verify CONSISTENT RESULTS but not CONSISTENT SUPPORT.

In addition to comparing semantics based upon the properties they satisfy, we can also compare them w.r.t. the set of answers they define.

Definition 3. Given two semantics S and S' , we say that:

- S' is an under-approximation (or: sound approximation) of S just in the case that

$$\langle \mathcal{T}, \mathcal{A} \rangle \models_{S'} q(\mathbf{a}) \quad \Rightarrow \quad \langle \mathcal{T}, \mathcal{A} \rangle \models_S q(\mathbf{a})$$

for every KB $\langle \mathcal{T}, \mathcal{A} \rangle$, query q , and tuple \mathbf{a} .

- S' is an over-approximation (or: complete approximation) of S just in the case that

$$\langle \mathcal{T}, \mathcal{A} \rangle \models_S q(\mathbf{a}) \quad \Rightarrow \quad \langle \mathcal{T}, \mathcal{A} \rangle \models_{S'} q(\mathbf{a})$$

for every KB $\langle \mathcal{T}, \mathcal{A} \rangle$, query q , and tuple \mathbf{a} .

We observe that the two consistency properties are preserved by taking under-approximations:

Theorem 1. Suppose that S' is an under-approximation of S , and let $P \in \{\text{CONSISTENT SUPPORT, CONSISTENT RESULTS}\}$. If S satisfies P , then S' also satisfies P .

We shall see later in the chapter that under- and over-approximations are not only useful for gaining a better understanding of the relationships between different semantics, but also because they allow us to use the query results of a computationally well-behaved semantics in order to (partially) compute the answers to queries under a computationally more difficult semantics. Indeed, if S' is an under-approximation of S , then we immediately have that every answer obtained using semantics S' is an answer under semantics S . Conversely, if S' is an over-approximation of S , then knowing that a tuple is not an answer under S' tells us that the tuple cannot be an answer under S .

3.1 Repairs

In order to extract reasonable answers from an inconsistent KB, it is useful to consider those portions of the data that are consistent with the TBox. This idea is captured by the notion of repair, defined as follows:

Definition 4. A 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 $\text{Rep}(\mathcal{A}, \mathcal{T})$ to denote the set of repairs of \mathcal{A} w.r.t. \mathcal{T} , which we abbreviate to $\text{Rep}(\mathcal{K})$ when $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$.

The repairs of an ABox correspond to the different ways of achieving consistency with the TBox while retaining as much of the original data as possible. By definition, every consistent KB has a single repair, consisting of the original ABox. When a KB is inconsistent, it is guaranteed to have at least one repair (take the empty set and keep adding assertions as long as consistency is preserved), but will more typically have more than one repair, reflecting the incomplete knowledge of which assertions are to blame for the inconsistency. We illustrate the notion of repair in the following example:

Example 6. As seen in an earlier example, the KB $\mathcal{K}_{\text{univ}} = \langle \mathcal{T}_{\text{univ}}, \mathcal{A}_{\text{univ}} \rangle$ is inconsistent. There are twelve repairs of $\mathcal{A}_{\text{univ}}$ w.r.t. $\mathcal{T}_{\text{univ}}$:

$$\begin{aligned}
\mathcal{R}_1 &= \{\text{Prof}(\text{sam}), \text{Prof}(\text{kim}), \text{Fellow}(\text{julie})\} \cup \mathcal{A}_{\text{Int}} \\
\mathcal{R}_2 &= \{\text{Lect}(\text{sam}), \text{Lect}(\text{kim}), \text{Fellow}(\text{julie})\} \cup \mathcal{A}_{\text{Int}} \\
\mathcal{R}_3 &= \{\text{Fellow}(\text{sam}), \text{Prof}(\text{kim}), \text{Fellow}(\text{julie})\} \cup \mathcal{A}_{\text{Int}} \\
\mathcal{R}_4 &= \{\text{Prof}(\text{sam}), \text{Lect}(\text{kim}), \text{Fellow}(\text{julie})\} \cup \mathcal{A}_{\text{Int}} \\
\mathcal{R}_5 &= \{\text{Lect}(\text{sam}), \text{Prof}(\text{kim}), \text{Fellow}(\text{julie})\} \cup \mathcal{A}_{\text{Int}} \\
\mathcal{R}_6 &= \{\text{Fellow}(\text{sam}), \text{Lect}(\text{kim}), \text{Fellow}(\text{julie})\} \cup \mathcal{A}_{\text{Int}} \\
\mathcal{R}_7 &= \{\text{Prof}(\text{sam}), \text{Prof}(\text{kim}), \text{Teaches}(\text{csc343}, \text{julie})\} \cup \mathcal{A}_{\text{Int}} \\
\mathcal{R}_8 &= \{\text{Lect}(\text{sam}), \text{Lect}(\text{kim}), \text{Teaches}(\text{csc343}, \text{julie})\} \cup \mathcal{A}_{\text{Int}} \\
\mathcal{R}_9 &= \{\text{Fellow}(\text{sam}), \text{Prof}(\text{kim}), \text{Teaches}(\text{csc343}, \text{julie})\} \cup \mathcal{A}_{\text{Int}} \\
\mathcal{R}_{10} &= \{\text{Prof}(\text{sam}), \text{Lect}(\text{kim}), \text{Teaches}(\text{csc343}, \text{julie})\} \cup \mathcal{A}_{\text{Int}} \\
\mathcal{R}_{11} &= \{\text{Lect}(\text{sam}), \text{Prof}(\text{kim}), \text{Teaches}(\text{csc343}, \text{julie})\} \cup \mathcal{A}_{\text{Int}} \\
\mathcal{R}_{12} &= \{\text{Fellow}(\text{sam}), \text{Lect}(\text{kim}), \text{Teaches}(\text{csc343}, \text{julie})\} \cup \mathcal{A}_{\text{Int}}
\end{aligned}$$

where the ABox \mathcal{A}_{Int} that is common to all the repairs is as follows:

$$\mathcal{A}_{\text{Int}} = \{\text{Fellow}(\text{alex}), \text{Teaches}(\text{alex}, \text{csc486})\}$$

Indeed, it is easily verified that every \mathcal{R}_i is $\mathcal{T}_{\text{univ}}$ -consistent and is maximal in the sense that adding any additional assertion from $\mathcal{A}_{\text{univ}}$ leads to a contradiction. Moreover, every $\mathcal{T}_{\text{univ}}$ -consistent subset of $\mathcal{A}_{\text{univ}}$ is contained in one of the \mathcal{R}_i .

In the following sections, we will introduce several different inconsistency-tolerant semantics based upon the notion of repair.

3.2 AR Semantics

The most well-known, and arguably the most natural, inconsistency-tolerant semantics is the AR semantics, which was first introduced in [32], inspired by earlier work on consistent query answering in relational databases [1, 17, 7].

Note that in the following definition, and throughout this section, we assume that $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is a KB, q is a conjunctive query, and \mathbf{a} is a tuple of constants from \mathcal{A} of the same arity as q .

Definition 5 (AR semantics). *A tuple \mathbf{a} is a certain answer of q over $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ under the AR (ABox Repair) semantics, written $\mathcal{K} \models_{\text{AR}} q(\mathbf{a})$, just in the case that $\langle \mathcal{T}, \mathcal{B} \rangle \models q(\mathbf{a})$ for every repair $\mathcal{B} \in \text{Rep}(\mathcal{K})$.*

The intuition behind the AR semantics is as follows. Under the assumption that the ABox is mostly correct, it seems reasonable to suppose that one of the repairs reflects the correct part of the ABox. In the absence of further information, we cannot identify which repair is the “correct” one, and so we only consider a tuple to be a query answer if it can be obtained from every repair.

We remark that AR semantics can be viewed a natural generalization of classical semantics. Indeed, classical semantics requires that a query answer hold in *all models* of a KB, since it is unknown which interpretation correctly describes the actual situation. With AR semantics, we additionally have uncertainty on the repair, so we consider *all models of all repairs* of the KB.

We illustrate the AR semantics on our running example:

Example 7. We determine the answers for our example queries q_1, q_2, q_3 over the KB $\mathcal{K}_{\text{univ}}$ using the AR semantics. For query $q_1 = \text{Faculty}(x)$, we observe that for each of the repairs $\mathcal{R}_i \in \text{Rep}(\mathcal{K}_{\text{univ}})$ and each of the individuals $\text{ind} \in \{\text{sam}, \text{kim}, \text{alex}\}$, we have $\langle \mathcal{T}_{\text{univ}}, \mathcal{R}_i \rangle \models \text{Faculty}(\text{ind})$:

- each \mathcal{R}_i contains one of $\text{Prof}(\text{sam})$, $\text{Lect}(\text{sam})$, and $\text{Fellow}(\text{sam})$, and each of these assertions allows us to infer $\text{Faculty}(\text{sam})$;
- each \mathcal{R}_i contains either $\text{Prof}(\text{kim})$ or $\text{Lect}(\text{kim})$, and both assertions imply $\text{Faculty}(\text{kim})$ in the presence of $\mathcal{T}_{\text{univ}}$;
- each \mathcal{R}_i contains $\text{Fellow}(\text{alex})$, which allows us to derive $\text{Faculty}(\text{alex})$.

It follows that the individuals **sam**, **kim**, and **alex** are all answers to q_1 under AR semantics:

$$\mathcal{K}_{\text{univ}} \models_{\text{AR}} q_1(\text{sam}) \quad \mathcal{K}_{\text{univ}} \models_{\text{AR}} q_1(\text{kim}) \quad \mathcal{K}_{\text{univ}} \models_{\text{AR}} q_1(\text{alex})$$

Note that these are the only answers to q_1 under AR semantics:

$$\mathcal{K}_{\text{univ}} \not\models_{\text{AR}} q_1(\text{julie}) \quad \mathcal{K}_{\text{univ}} \not\models_{\text{AR}} q_1(\text{csc486}) \quad \mathcal{K}_{\text{univ}} \not\models_{\text{AR}} q_1(\text{csc343})$$

To see why **julie** does not count as an answer, observe that there exist repairs (like \mathcal{R}_7) from which $\text{Faculty}(\text{julie})$ cannot be derived. These repairs describe possible states of affairs in which **julie** may not be a faculty member.

Next consider the query $q_2 = \exists y \text{Teaches}(x, y)$. It is not hard to see that for every repair \mathcal{R}_i , we have $\langle \mathcal{T}_{\text{univ}}, \mathcal{R}_i \rangle \models q_2(\text{kim})$ and $\langle \mathcal{T}_{\text{univ}}, \mathcal{R}_i \rangle \models q_2(\text{alex})$. Indeed:

- each \mathcal{R}_i contains either $\text{Prof}(\text{kim})$ or $\text{Lect}(\text{kim})$, both of which allow us to infer $\exists y \text{Teaches}(\text{kim}, y)$;
- each \mathcal{R}_i contains $\text{Teaches}(\text{alex}, \text{csc486})$, which directly yields $\exists y \text{Teaches}(\text{alex}, y)$.

We thus obtain

$$\langle \mathcal{T}_{\text{univ}}, \mathcal{A}_{\text{univ}} \rangle \models_{\text{AR}} q_2(\text{kim}) \quad \langle \mathcal{T}_{\text{univ}}, \mathcal{A}_{\text{univ}} \rangle \models_{\text{AR}} q_2(\text{alex})$$

However, it can be verified that **sam** is *not* a certain answer to q_2 over the KB $\langle \mathcal{T}_{\text{univ}}, \mathcal{R}_3 \rangle$, and hence not an answer to q_2 under AR semantics. We can similarly show that **julie**, **csc486**, and **csc343** are not answers to q_2 under AR semantics.

We can show that the AR semantics verifies both consistency properties.

Theorem 2. *AR semantics satisfies the properties CONSISTENT SUPPORT and CONSISTENT RESULTS.*

Proof. For CONSISTENT SUPPORT, it suffices to observe that every KB possesses at least one repair, and so if a query result can be obtained from all repairs, then it can be obtained from at least one repair. For CONSISTENT RESULTS, pick some repair \mathcal{R} of \mathcal{K} , and let \mathcal{I} be any model of the consistent KB $\langle \mathcal{T}, \mathcal{R} \rangle$. By definition, if $\mathcal{K} \models_{\text{AR}} q(\mathbf{a})$, then $\langle \mathcal{T}, \mathcal{R} \rangle \models q(\mathbf{a})$, and hence $\mathcal{I} \models q(\mathbf{a})$. \square

While the AR semantics is very natural, there are at least two reasons to consider other semantics. First, it may be interesting to use an alternative semantics instead of (or conjointly with) the AR semantics in order to restrict the set of answers further, or inversely, to be more liberal and obtain more possible answers. Second, we shall see in the next section that query answering under AR semantics is very often intractable, thus motivating the interest of considering alternative semantics with better computational properties.

3.3 IAR Semantics

The AR semantics corresponds to querying each of the repairs separately, then intersecting the sets of answers. If we instead start by intersecting the repairs, then querying the resulting ABox, we obtain the IAR semantics [32]:

Definition 6 (IAR semantics). *A tuple \mathbf{a} is a certain answer of q over \mathcal{K} under the IAR (Intersection of ABox Repairs) semantics, written $\mathcal{K} \models_{\text{IAR}} q(\mathbf{a})$, just in the case that $\langle \mathcal{T}, \mathcal{D} \rangle \models q(\mathbf{a})$ where $\mathcal{D} = \bigcap_{\mathcal{B} \in \text{Rep}(\mathcal{K})} \mathcal{B}$.*

Observe that if an assertion does not appear in every repair, it is because the assertion belongs to a minimal inconsistent subset of the ABox. Thus, the IAR semantics computes the query answers that can be obtained from the ‘surest’ assertions in the ABox, i.e., those that are not involved in any contradiction. This is in contrast to the AR semantics, which allows inferences based upon potentially incorrect facts so long as all the alternatives support the conclusion.

The following theorem, which directly follows from the definitions, formalizes the relationship between AR and IAR semantics:

Theorem 3. *IAR semantics is an under-approximation of AR semantics.*

The difference between the IAR and AR semantics is illustrated by the next example:

Example 8. To determine the answers to q_1 , q_2 , and q_3 on $\mathcal{K}_{\text{univ}}$ under IAR semantics, it suffices to compute the certain answers to these queries over the intersection

$$\mathcal{A}_{\text{int}} = \{\text{Fellow}(\text{alex}), \text{Teaches}(\text{alex}, \text{csc486})\}$$

of the repairs of $\mathcal{K}_{\text{univ}}$. For all three queries, we obtain only alex as an answer:

$$\mathcal{K}_{\text{univ}} \models_{\text{IAR}} q_1(\text{alex}) \quad \mathcal{K}_{\text{univ}} \models_{\text{IAR}} q_2(\text{alex}) \quad \mathcal{K}_{\text{univ}} \models_{\text{IAR}} q_3(\text{alex})$$

Observe that sam and kim, which were answers to q_1 under AR semantics, are not considered answers under the stricter IAR semantics, since obtaining these answers requires reasoning by cases (e.g. sam is either a professor, lecturer, or research fellow; kim is either a professor or lecturer). By contrast, to derive $q_1(\text{alex})$, we only need to consider the assertion Fellow(alex), which belongs to every repair and thus can be viewed as ‘uncontroversial’. Similarly, for query q_2 , kim is an answer under AR semantics, but we cannot find a subset of facts common to all repairs from which we can derive these query answers. For this reason, kim is not returned as an answer under the stricter IAR semantics.

By combining Theorems 1, 2, and 3, we can show that IAR semantics satisfies both consistency properties:

Theorem 4. *IAR semantics satisfies the properties CONSISTENT SUPPORT and CONSISTENT RESULTS.*

3.4 Brave Semantics

Instead of considering those tuples that are answers w.r.t. every repair, one may consider those which are answers in at least one repair. This idea is captured by the brave semantics [13]:

Definition 7 (Brave semantics). *A tuple \mathbf{a} is a certain answer of q over $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ under the brave semantics, written $\mathcal{K} \models_{\text{brave}} q(\mathbf{a})$, just in the case that $\langle \mathcal{T}, \mathcal{B} \rangle \models q(\mathbf{a})$ for some repair $\mathcal{B} \in \text{Rep}(\mathcal{K})$.*

The following theorem, which relates the brave and AR semantics, is immediate from the definition:

Theorem 5. *Brave semantics is an over-approximation of AR semantics.*

The following example shows that the brave and AR semantics can differ:

Example 9. We continue our running example, this time using the brave semantics. For query q_1 , moving from AR to brave semantics yields one additional answer, namely julie:

$$\begin{array}{ll} \mathcal{K}_{\text{univ}} \models_{\text{brave}} q_1(\text{sam}) & \mathcal{K}_{\text{univ}} \models_{\text{brave}} q_1(\text{kim}) \\ \mathcal{K}_{\text{univ}} \models_{\text{brave}} q_1(\text{alex}) & \mathcal{K}_{\text{univ}} \models_{\text{brave}} q_1(\text{julie}) \end{array}$$

Indeed, while julie is not a certain answer to q_1 in every repair (as required by AR semantics), it is a certain answer to q_1 w.r.t. repairs $\mathcal{R}_1 - \mathcal{R}_6$, and thus holds under brave semantics. Each of these repairs describes a possible world in which $q_1(\text{julie})$ holds. Observe that the individuals csc486 and csc343 do not count as answers to q_1 under brave semantics, as they are not certain answers to q_1 in any of the repairs.

For query q_2 , we obtain the following four answers under brave semantics:

$$\begin{array}{ll} \mathcal{K}_{\text{univ}} \models_{\text{brave}} q_2(\text{sam}) & \mathcal{K}_{\text{univ}} \models_{\text{brave}} q_2(\text{kim}) \\ \mathcal{K}_{\text{univ}} \models_{\text{brave}} q_2(\text{alex}) & \mathcal{K}_{\text{univ}} \models_{\text{brave}} q_2(\text{csc343}) \end{array}$$

The individual csc343, which was not an answer under AR semantics, is returned as a brave answer because of the assertion $\text{Teaches}(\text{csc343}, \text{julie})$ which is present in repairs $\mathcal{R}_7 - \mathcal{R}_{12}$. The individuals julie and csc486 are not brave answers, as none of the repairs contains evidence that these individuals teach something.

We next show that the brave semantics satisfies one of the consistency properties, but not the other.

Theorem 6. *Brave semantics satisfies CONSISTENT SUPPORT but does not satisfy CONSISTENT RESULTS.*

Proof. The property CONSISTENT SUPPORT holds by definition: every query answer holds in some repairs, and repairs are consistent subsets of the original ABox. To see why CONSISTENT RESULTS does not hold, we observe that for our example KB $\mathcal{K}_{\text{univ}}$, we have both $\mathcal{K}_{\text{univ}} \models_{\text{brave}} \text{Prof}(\text{sam})$ and $\mathcal{K}_{\text{univ}} \models_{\text{brave}} \text{Lect}(\text{sam})$. Due to the presence of the negative inclusion $\text{Prof} \sqsubseteq \neg \text{Lect}$ in $\mathcal{T}_{\text{univ}}$, no model of $\mathcal{T}_{\text{univ}}$ can satisfy both $\text{Prof}(\text{sam})$ and $\text{Lect}(\text{sam})$. \square

In fact, it is not hard to see that brave semantics is the weakest semantics satisfying CONSISTENT SUPPORT:

Theorem 7. *For every semantics S satisfying CONSISTENT SUPPORT, brave semantics is an over-approximation of S .*

In other words, the answers returned by the brave semantics are guaranteed to contain all of the answers that can be obtained with any semantics that satisfies the CONSISTENT SUPPORT property. Thus, the brave answers can very naturally be considered as the set of possible or maybe answers.

3.5 k -Support and k -Defeater Semantics

We have seen in the preceding subsections that the IAR semantics and brave semantics provide natural under- and over-approximations, respectively, of the AR semantics. The families of k -support and k -defeater semantics [13] were introduced to generalize these semantics and obtain more fine-grained approximations.

We begin by considering the family of k -support semantics, which approximate the AR semantics from below:

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

- each S_i is a \mathcal{T} -support for $q(\mathbf{a})$ in \mathcal{A}
- for every $R \in \text{Rep}(\mathcal{K})$, there is some S_i with $S_i \subseteq R$

The intuition is as follows: if $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\text{AR}} q(\mathbf{a})$, then there must exist a set $\{S_1, \dots, S_n\}$ of \mathcal{T} -supports for $q(\mathbf{a})$ such that every repair contains at least one of the sets S_i . The k -support semantics can thus be seen as a restriction of the AR semantics in which a maximum of k different supports can be used.

The following example illustrates how the k -support semantics changes as we vary the value of k :

Example 10. When $k = 1$, we obtain the same query answers as for IAR semantics (cf. Theorem 8):

$$\mathcal{K}_{\text{univ}} \models_{1\text{-supp}} q_1(\text{alex}) \quad \mathcal{K}_{\text{univ}} \models_{1\text{-supp}} q_2(\text{alex})$$

To see why, observe that $\{\text{Fellow}(\text{alex}), \text{Teaches}(\text{alex}, \text{csc486})\}$ is a support for both query answers, and it is contained in every repair.

Evaluating q_1 under 2-support semantics yields one additional answer:

$$\mathcal{K}_{\text{univ}} \models_{2\text{-supp}} q_1(\text{kim}) \quad \mathcal{K}_{\text{univ}} \models_{2\text{-supp}} q_1(\text{alex})$$

To show that kim is an answer, we can take the supports $S_1 = \{\text{Prof}(\text{kim})\}$ and $S_2 = \{\text{Lect}(\text{kim})\}$ for $q_1(\text{kim})$ and observe that every repair $\mathcal{R}_i \in \text{Rep}(\mathcal{K}_{\text{univ}})$ contains either S_1 or S_2 .

When $k = 3$, we obtain the same answers for q_1 as under AR semantics:

$$\mathcal{K}_{\text{univ}} \models_{3\text{-supp}} q_1(\text{sam}) \quad \mathcal{K}_{\text{univ}} \models_{3\text{-supp}} q_1(\text{kim}) \quad \mathcal{K}_{\text{univ}} \models_{3\text{-supp}} q_1(\text{alex})$$

It can be verified that this same set of answers is obtained for every $k \geq 3$.

We next consider the query q_2 . When $k = 2$, we additionally obtain kim as an answer:

$$\mathcal{K}_{\text{univ}} \models_{2\text{-supp}} q_2(\text{kim}) \quad \mathcal{K}_{\text{univ}} \models_{2\text{-supp}} q_2(\text{alex})$$

Indeed, we can take the same pair of supports S_1 and S_2 as used for q_1 . We obtain exactly the same result for every higher value of k .

The key properties of the k -support semantics are summarized in the following theorem. The first point states that the k -support semantics coincides with the IAR semantics when $k = 1$. The second item shows that for every value of k , the k -support semantics is an under-approximation of the AR semantics. Moreover, by the third statement, there exists a value of k for which the AR semantics and k -support semantics coincide. By the fourth property, there is a monotone convergence of the k -support semantics to the AR semantics: when we move to a higher value of k , we keep all of the answers obtained for lower k values and possibly add some further answers, until we reach the AR semantics. Note that the value of k needed to converge to the AR semantics will depend on the particular KB and query under consideration. The final item states that, like the IAR semantics, the k -support semantics satisfy both consistency conditions.

Theorem 8. *For every KB \mathcal{K} :*

1. $\mathcal{K} \models_{\text{IAR}} q(\mathbf{a})$ if and only if $\mathcal{K} \models_{1\text{-supp}} q(\mathbf{a})$;
2. for every $k \geq 1$, $\mathcal{K} \models_{k\text{-supp}} q(\mathbf{a})$ implies $\mathcal{K} \models_{\text{AR}} q(\mathbf{a})$;
3. for every \mathcal{K} , there exists some $k \geq 1$ such that $\mathcal{K} \models_{\text{AR}} q(\mathbf{a})$ if and only if $\mathcal{K} \models_{k\text{-supp}} q(\mathbf{a})$;
4. for every $k \geq 1$, if $\mathcal{K} \models_{k\text{-supp}} q(\mathbf{a})$, then $\mathcal{K} \models_{k+1\text{-supp}} q(\mathbf{a})$;
5. for every $k \geq 1$, the k -support semantics satisfies properties CONSISTENT SUPPORT and CONSISTENT RESULTS.

Proof. The first two statements follows directly from the definitions of the semantics. For the third point, it suffices to take $k = 2^{|\mathcal{A}|}$. Indeed, there cannot be more than $2^{|\mathcal{A}|}$ different repairs of \mathcal{A} w.r.t. \mathcal{T} , and thus, at most $2^{|\mathcal{A}|}$ different supports are needed to cover all repairs. For the fourth property, we note that the definition allows us to use a support more than once. Thus, if $\mathcal{K} \models_{k\text{-supp}} q(\mathbf{a})$ with supports S_1, \dots, S_k , then we can use the sequence of supports S_1, \dots, S_k, S_{k+1} with $S_{k+1} = S_k$ to show that $\mathcal{K} \models_{k+1\text{-supp}} q(\mathbf{a})$. The final item follows from Theorems 1 and 2, together with the second item of the theorem. \square

We next introduce the family of k -defeater semantics, which approximate the AR semantics from above:

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

We remark that if an ABox does not contain any \mathcal{T} -support for $q(\mathbf{a})$, then \mathbf{a} is not a certain answer under 0-defeater semantics since one can simply take $S = \emptyset$ as the defeating set.

Example 11. For all three queries, the set of answers under 0-defeater semantics is the same as for brave semantics. One can show that for every $k \geq 1$, the answers to these queries under k -defeater semantics is the same as for AR semantics.

To illustrate, let us consider the query q_2 . When $k = 0$, we have the following four answers: sam, kim, alex, and csc343. When $k = 1$, we ‘lose’ the answers sam and csc343:

$$\mathcal{K}_{\text{univ}} \not\models_{1\text{-def}} q_2(\text{sam}) \quad \mathcal{K}_{\text{univ}} \not\models_{1\text{-def}} q_2(\text{csc343})$$

Indeed, $q_2(\text{sam})$ has two minimal supports $\{\text{Prof}(\text{sam})\}$ and $\{\text{Lect}(\text{sam})\}$, both of which are contradicted by the 1-element set $\{\text{Fellow}(\text{sam})\}$. As for csc343, it is easy to see that $\{\text{Teaches}(\text{csc343}, \text{julie})\}$ is the only minimal support for $q_2(\text{csc343})$, and it can be contradicted by the single assertion $\text{Fellow}(\text{julie})$. The two other individuals (kim and alex) are still answers under 1-defeater semantics:

$$\mathcal{K}_{\text{univ}} \models_{1\text{-def}} q_2(\text{kim}) \quad \mathcal{K}_{\text{univ}} \models_{1\text{-def}} q_2(\text{alex})$$

This is because there is no single assertion that simultaneously contradicts the two supports ($\{\text{Prof}(\text{kim})\}$ and $\{\text{Lect}(\text{kim})\}$) of $q_2(\text{kim})$, and nor any assertion in conflict with the unique support of $q_2(\text{alex})$ (which is $\{\text{Fellow}(\text{alex}), \text{Teaches}(\text{alex}, \text{csc486})\}$).

The properties of the k -defeater semantics are resumed in the following theorem. The first three statements show that the k -defeater semantics equals the brave semantics when $k = 0$ and converges to the AR semantics in the limit. The fourth statement shows that the convergence to the AR semantics is anti-monotone in k : as we increase the value of k , the set of answers can only decrease, until we reach the set of AR answers. Like the brave semantics, the k -defeater semantics satisfy the first consistency property, but not the second.

Theorem 9.

1. $\mathcal{K} \models_{\text{brave}} q(\mathbf{a})$ if and only if $\mathcal{K} \models_{0\text{-def}} q(\mathbf{a})$;
2. for every $k \geq 0$, $\mathcal{K} \models_{k\text{-def}} q(\mathbf{a})$ implies $\mathcal{K} \models_{\text{AR}} q(\mathbf{a})$;
3. for every \mathcal{K} , there exists k such that $\mathcal{K} \models_{\text{AR}} q(\mathbf{a})$ if and only if $\mathcal{K} \models_{k\text{-def}} q(\mathbf{a})$;
4. for every $k \geq 0$, if $\mathcal{K} \models_{k+1\text{-def}} q(\mathbf{a})$, then $\mathcal{K} \models_{k\text{-def}} q(\mathbf{a})$;
5. for every $k \geq 0$, the k -defeater semantics satisfies CONSISTENT SUPPORT but not CONSISTENT RESULTS.

Proof. The first two statements follow easily from the definitions. For the third statement, it suffices to set $k = |\mathcal{A}|$ since we only consider sets S that are subsets of \mathcal{A} . We prove the fourth statement by contraposition: if $\mathcal{K} \not\models_{k\text{-def}} q(\mathbf{a})$, then there exists a set $S \subseteq \mathcal{A}$ with $|S| \leq k$ that contradicts every support for $q(\mathbf{a})$, and this same set S can be used to witness that $\mathcal{K} \not\models_{k+1\text{-def}} q(\mathbf{a})$. Finally, regarding the consistency properties, to show that the k -defeater semantics satisfies CONSISTENT SUPPORT it suffices to recall that the brave semantics does not

satisfy this property (Theorem 6) and to apply Theorem 1. To see why CONSISTENT RESULTS is not satisfied by the k -defeater semantics, take the TBox $\mathcal{T} = \{A_1 \sqsubseteq \neg A_2\} \cup \{\exists R_j \sqsubseteq A_j, \exists R_j^- \sqsubseteq \neg B_j \mid j \in \{1, 2\}\}$ and the ABox $\mathcal{A} = \{R_j(a, b_i), B_j(b_i) \mid j \in \{1, 2\}, 1 \leq i \leq k+1\}$. It can be verified that for both $j \in \{1, 2\}$, we have $\langle \mathcal{T}, \mathcal{A} \rangle \models_{k\text{-def}} A_j(a)$, as any defeating set for $A_j(a)$ must contain $B_j(b_i)$ for every $1 \leq i \leq k+1$. However, there can be no model \mathcal{I} of \mathcal{T} that satisfies both $A_1(a)$ and $A_2(a)$, because \mathcal{T} contains the axiom $A_1 \sqsubseteq \neg A_2$. \square

3.6 ICR Semantics

Another way of obtaining a finer under-approximation of AR semantics than the IAR semantics is to close the repairs before intersecting them. This idea is formalized in the ICR semantics, proposed in [8]:

Definition 10 (ICR semantics). *A tuple \mathbf{a} is a certain answer of q over $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ under the ICR (Intersection of Closed Repairs) semantics, written $\mathcal{K} \models_{\text{ICR}} q(\mathbf{a})$, just in the case that $\langle \mathcal{T}, \mathcal{D} \rangle \models q(\mathbf{a})$ where $\mathcal{D} = \bigcap_{\mathcal{B} \in \text{Rep}(\mathcal{K})} \text{cl}_{\mathcal{T}}(\mathcal{B})$.*

Example 12. To compute the answers to q_1 , q_2 , and q_3 under ICR semantics, we first compute the closure of the repairs of $\mathcal{K}_{\text{univ}}$:

$$\begin{aligned}
\text{cl}_{\mathcal{T}_{\text{univ}}}(\mathcal{R}_1) &= \{\text{Prof}(\text{sam}), \text{Prof}(\text{kim}), \text{Fellow}(\text{julie})\} \cup \mathcal{A}'_{\text{Int}} \cup \{\text{Faculty}(\text{julie})\} \\
\text{cl}_{\mathcal{T}_{\text{univ}}}(\mathcal{R}_2) &= \{\text{Lect}(\text{sam}), \text{Lect}(\text{kim}), \text{Fellow}(\text{julie})\} \cup \mathcal{A}'_{\text{Int}} \cup \{\text{Faculty}(\text{julie})\} \\
\text{cl}_{\mathcal{T}_{\text{univ}}}(\mathcal{R}_3) &= \{\text{Fellow}(\text{sam}), \text{Prof}(\text{kim}), \text{Fellow}(\text{julie})\} \cup \mathcal{A}'_{\text{Int}} \cup \{\text{Faculty}(\text{julie})\} \\
\text{cl}_{\mathcal{T}_{\text{univ}}}(\mathcal{R}_4) &= \{\text{Prof}(\text{sam}), \text{Lect}(\text{kim}), \text{Fellow}(\text{julie})\} \cup \mathcal{A}'_{\text{Int}} \cup \{\text{Faculty}(\text{julie})\} \\
\text{cl}_{\mathcal{T}_{\text{univ}}}(\mathcal{R}_5) &= \{\text{Lect}(\text{sam}), \text{Prof}(\text{kim}), \text{Fellow}(\text{julie})\} \cup \mathcal{A}'_{\text{Int}} \cup \{\text{Faculty}(\text{julie})\} \\
\text{cl}_{\mathcal{T}_{\text{univ}}}(\mathcal{R}_6) &= \{\text{Fellow}(\text{sam}), \text{Lect}(\text{kim}), \text{Fellow}(\text{julie})\} \cup \mathcal{A}'_{\text{Int}} \cup \{\text{Faculty}(\text{julie})\} \\
\text{cl}_{\mathcal{T}_{\text{univ}}}(\mathcal{R}_7) &= \{\text{Prof}(\text{sam}), \text{Prof}(\text{kim}), \text{Teaches}(\text{csc343}, \text{julie})\} \cup \mathcal{A}'_{\text{Int}} \cup \{\text{Course}(\text{julie})\} \\
\text{cl}_{\mathcal{T}_{\text{univ}}}(\mathcal{R}_8) &= \{\text{Lect}(\text{sam}), \text{Lect}(\text{kim}), \text{Teaches}(\text{csc343}, \text{julie})\} \cup \mathcal{A}'_{\text{Int}} \cup \{\text{Course}(\text{julie})\} \\
\text{cl}_{\mathcal{T}_{\text{univ}}}(\mathcal{R}_9) &= \{\text{Fellow}(\text{sam}), \text{Prof}(\text{kim}), \text{Teaches}(\text{csc343}, \text{julie})\} \cup \mathcal{A}'_{\text{Int}} \cup \{\text{Course}(\text{julie})\} \\
\text{cl}_{\mathcal{T}_{\text{univ}}}(\mathcal{R}_{10}) &= \{\text{Prof}(\text{sam}), \text{Lect}(\text{kim}), \text{Teaches}(\text{csc343}, \text{julie})\} \cup \mathcal{A}'_{\text{Int}} \cup \{\text{Course}(\text{julie})\} \\
\text{cl}_{\mathcal{T}_{\text{univ}}}(\mathcal{R}_{11}) &= \{\text{Lect}(\text{sam}), \text{Prof}(\text{kim}), \text{Teaches}(\text{csc343}, \text{julie})\} \cup \mathcal{A}'_{\text{Int}} \cup \{\text{Course}(\text{julie})\} \\
\text{cl}_{\mathcal{T}_{\text{univ}}}(\mathcal{R}_{12}) &= \{\text{Fellow}(\text{sam}), \text{Lect}(\text{kim}), \text{Teaches}(\text{csc343}, \text{julie})\} \cup \mathcal{A}'_{\text{Int}} \cup \{\text{Course}(\text{julie})\}
\end{aligned}$$

where $\mathcal{A}'_{\text{Int}}$ is the following ABox:

$$\mathcal{A}'_{\text{Int}} = \mathcal{A}_{\text{Int}} \cup \{\text{Faculty}(\text{sam}), \text{Faculty}(\text{kim}), \text{Faculty}(\text{alex}), \text{Course}(\text{csc486})\}$$

and \mathcal{A}_{Int} is defined as before as the intersection of repairs. It can be verified that $\mathcal{A}'_{\text{Int}}$ is the ABox resulting from intersecting the twelve closed repairs.

The query q_1 has three certain answers over the KB $\langle \mathcal{T}_{\text{univ}}, \mathcal{A}'_{\text{Int}} \rangle$: **sam**, **kim**, **alex**. We thus obtain exactly the same set of answers as was obtained under AR semantics:

$$\mathcal{K}_{\text{univ}} \models_{\text{ICR}} q_1(\text{sam}) \quad \mathcal{K}_{\text{univ}} \models_{\text{ICR}} q_1(\text{kim}) \quad \mathcal{K}_{\text{univ}} \models_{\text{ICR}} q_1(\text{alex})$$

Observe that ICR semantics returns `sam` and `kim` as answers, whereas neither individual is an answer under IAR semantics.

For q_2 , there is a single certain answer over $\langle \mathcal{T}_{\text{univ}}, \mathcal{A}'_{\text{int}} \rangle$, namely the individual `alex`. It follows that we have:

$$\mathcal{K}_{\text{univ}} \models_{\text{ICR}} q_2(\text{alex})$$

and that all other individuals are not answers under ICR semantics. Indeed, while every repair allows us to infer $\exists y \text{Teaches}(\text{kim}, y)$, the closures of the repairs (hence their intersection) do not contain any assertions of the form $\text{Teaches}(\text{kim}, _)$.

As mentioned earlier, the ICR semantics is positioned between the AR and IAR semantics. Formally, we have the following relationships:

Theorem 10.

1. ICR semantics is an under-approximation of AR semantics;
2. ICR semantics is an over-approximation of IAR semantics;
3. ICR semantics is equal to AR semantics for the class of ground CQs (i.e. CQs without existential quantifiers).

Using Theorem 1, we can show that the ICR semantics satisfies both consistency properties.

Theorem 11. *ICR semantics satisfies the properties CONSISTENT SUPPORT and CONSISTENT RESULTS.*

3.7 CAR and ICAR Semantics

Like the ICR semantics, the CAR and ICAR semantics [32] also involve a closure operation, but this time the closure is performed directly on the input ABox (rather than on the repairs).

In order to be able to distinguish between different inconsistent KBs, the authors of [32] introduce a special closure operator ($\text{cl}_{\mathcal{T}}^*$) for inconsistent KBs, defined as follows:

$$\text{cl}_{\mathcal{T}}^*(\mathcal{A}) = \{\beta \mid \exists S \subseteq \mathcal{A} \text{ such that } S \text{ is } \mathcal{T}\text{-consistent and } \langle \mathcal{T}, S \rangle \models \beta\}$$

We observe that $\text{cl}_{\mathcal{T}}^*(\mathcal{A})$ corresponds to the set of ABox assertions that are entailed under brave semantics.

An alternative notion of repair is introduced in order to incorporate assertions from $\text{cl}_{\mathcal{T}}^*(\mathcal{A})$.

Definition 11 (Closed ABox repair). *A subset $\mathcal{R} \subseteq \text{cl}_{\mathcal{T}}^*(\mathcal{A})$ 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 $\mathcal{R}' \subseteq \text{cl}_{\mathcal{T}}^*(\mathcal{A})$ 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 $\text{ClosedRep}(\mathcal{K})$.*

It should be noted that every closed ABox repair is a repair of the KB $\langle \mathcal{T}, \text{cl}_{\mathcal{T}}^*(\mathcal{A}) \rangle$. However, the converse does not hold in general (i.e., some repairs of $\langle \mathcal{T}, \text{cl}_{\mathcal{T}}^*(\mathcal{A}) \rangle$ are not closed ABox repairs), as the following example demonstrates:

Example 13. We consider another example KB about the university domain:

$$\begin{aligned} \mathcal{T} &= \{\text{Prof} \sqsubseteq \text{Employee}, \text{UnderGrad} \sqsubseteq \text{Student}, \\ &\quad \text{Student} \sqsubseteq \neg \text{Prof}, \text{UnderGrad} \sqsubseteq \neg \text{Employee}\} \\ \mathcal{A} &= \{\text{Prof}(\text{sam}), \text{UnderGrad}(\text{sam})\} \end{aligned}$$

The TBox states that professors are employees, that undergraduate students are students, and that students and professors and undergraduate students and employees are disjoint classes. Applying the closure operator $\text{cl}_{\mathcal{T}}^*$ to \mathcal{A} gives:

$$\text{cl}_{\mathcal{T}}^*(\mathcal{A}) = \mathcal{A} \cup \{\text{Employee}(\text{sam}), \text{Student}(\text{sam})\}$$

and it can be easily verified that $\langle \mathcal{T}, \text{cl}_{\mathcal{T}}^*(\mathcal{A}) \rangle$ has the following three repairs:

$$\begin{aligned} \mathcal{R}_1 &= \{\text{Prof}(\text{sam}), \text{Employee}(\text{sam})\} \\ \mathcal{R}_2 &= \{\text{UnderGrad}(\text{sam}), \text{Student}(\text{sam})\} \\ \mathcal{R}_3 &= \{\text{Employee}(\text{sam}), \text{Student}(\text{sam})\} \end{aligned}$$

The repair \mathcal{R}_3 is not a closed ABox repair because it has an empty intersection with \mathcal{A} , unlike \mathcal{R}_1 and \mathcal{R}_2 .

However, closed ABox repairs can be equivalently defined as the subsets of $\text{cl}_{\mathcal{T}}^*(\mathcal{A})$ that can be obtained as follows: (i) take some $\mathcal{R} \in \text{Rep}(\langle \mathcal{T}, \mathcal{A} \rangle)$, and (ii) add to \mathcal{R} an inclusion-maximal subset C of $\text{cl}_{\mathcal{T}}^*(\mathcal{A}) \setminus \mathcal{A}$ such that $\mathcal{R} \cup C$ is \mathcal{T} -consistent. Thus, closed ABox repairs can be seen as maximally ‘completing’ the standard ABox repairs with additional facts from $\text{cl}_{\mathcal{T}}^*(\mathcal{A}) \setminus \mathcal{A}$.

We can now take any of the semantics from earlier and substitute closed ABox repairs for the standard notion of repair. In particular, if we use closed ABox repairs in conjunction with the AR and IAR semantics, then we obtain the CAR and ICAR semantics introduced in [32]:

Definition 12 (CAR semantics). A tuple \mathbf{a} is a certain answer of q over $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ under the CAR (Closed ABox Repair) semantics, written $\mathcal{K} \models_{\text{CAR}} q(\mathbf{a})$, just in the case that $\langle \mathcal{T}, \mathcal{R} \rangle \models q(\mathbf{a})$ for every $\mathcal{R} \in \text{ClosedRep}(\mathcal{K})$.

Definition 13 (ICAR semantics). A tuple \mathbf{a} is a certain answer of q over $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ under the ICAR (Intersection of Closed ABox Repairs) semantics, written $\mathcal{K} \models_{\text{ICAR}} q(\mathbf{a})$, just in the case that $\langle \mathcal{T}, \mathcal{D} \rangle \models q(\mathbf{a})$ where \mathcal{D} is the intersection of the closed ABox repairs of \mathcal{A} w.r.t. \mathcal{T} .

We return to our running example.

Example 14. Applying the special closure operator to our example KB yields:

$$\text{cl}_{\mathcal{T}_{\text{univ}}}^*(\mathcal{A}_{\text{univ}}) = \mathcal{A}_{\text{univ}} \cup \{\text{Faculty}(\text{sam}), \text{Faculty}(\text{kim}), \text{Faculty}(\text{alex}), \\ \text{Course}(\text{csc486}), \text{Faculty}(\text{julie}), \text{Course}(\text{julie})\}$$

We next compute the closed ABox repairs of our example KB $\langle \mathcal{T}_{\text{univ}}, \mathcal{A}_{\text{univ}} \rangle$. As mentioned earlier, these can be obtained by taking each of the repairs $\mathcal{R}_1 - \mathcal{R}_{12}$ (see Example 6) and adding as many facts from $\text{cl}_{\mathcal{T}_{\text{univ}}}^*(\mathcal{A}_{\text{univ}}) \setminus \mathcal{A}$ as possible while preserving consistency. For our example KB, it turns out that this gives us precisely the same result as closing each of the original repairs of the KB:

$$\text{ClosedRep}(\langle \mathcal{T}_{\text{univ}}, \mathcal{A}_{\text{univ}} \rangle) = \{\text{cl}_{\mathcal{T}_{\text{univ}}}(\mathcal{R}_i) \mid \mathcal{R}_i \in \text{Rep}(\langle \mathcal{T}_{\text{univ}}, \mathcal{A}_{\text{univ}} \rangle)\}$$

It follows that on this KB, the CAR semantics gives the same answers as the AR semantics, and the ICAR semantics coincides with the ICR semantics.

In general, however, the CAR and ICAR semantics can yield different results. To demonstrate this, let us slightly modify our example KB by adding $\exists \text{Teaches} \sqsubseteq \text{Faculty}$ to the TBox. Call the modified TBox $\mathcal{T}'_{\text{univ}}$ and modified KB $\mathcal{K}'_{\text{univ}}$. The special closure operator gives almost the same result as before:

$$\text{cl}_{\mathcal{T}'_{\text{univ}}}^*(\mathcal{A}_{\text{univ}}) = \mathcal{A}_{\text{univ}} \cup \{\text{Faculty}(\text{sam}), \text{Faculty}(\text{kim}), \text{Faculty}(\text{alex}), \text{Course}(\text{csc486}), \\ \text{Faculty}(\text{julie}), \text{Course}(\text{julie}), \text{Faculty}(\text{csc343})\}$$

except that we now also have the fact $\text{Faculty}(\text{csc343})$, which is derived from $\text{Teaches}(\text{csc343}, \text{julie})$ using the new inclusion. Since $\text{Faculty}(\text{csc343})$ is not involved in any contradictions, it can be added to each repair while preserving consistency, and so it belongs to every closed ABox repair. It follows that csc343 is an answer to q_1 under ICAR and CAR semantics. Observe that csc343 is not an answer under AR semantics since the assertion $\text{Teaches}(\text{csc343}, \text{julie})$ which is required to infer $\text{Faculty}(\text{csc343})$ is not present in all (standard) repairs.

The following theorem, which easily follows from the definitions, resumes the key relationships holding with respect to the CAR and ICAR semantics:

Theorem 12.

1. *The CAR semantics is an over-approximation of the AR semantics.*
2. *The ICAR semantics is an over-approximation of the ICR semantics.*
3. *The ICAR semantics is an under-approximation of the CAR semantics.*
4. *The CAR semantics and ICAR semantics coincide for ground CQs.*

The next theorem shows that the CAR and ICAR semantics satisfy the second consistency property, but not the first.

Theorem 13. *CAR and ICAR semantics satisfy CONSISTENT RESULTS but do not satisfy CONSISTENT SUPPORT.*

Proof. A counter-example for CONSISTENT SUPPORT is given in the next example. To show that CONSISTENT RESULTS holds for CAR, we can use essentially the same proof as for Theorem 2: pick a closed ABox repair \mathcal{R} of \mathcal{K} , and let \mathcal{I} be any model of the consistent KB $\langle \mathcal{T}, \mathcal{R} \rangle$. Then by the definition of the CAR semantics, if $\mathcal{K} \models_{\text{CAR}} q(\mathbf{a})$, then $\langle \mathcal{T}, \mathcal{R} \rangle \models q(\mathbf{a})$, and hence $\mathcal{I} \models q(\mathbf{a})$. For the ICAR semantics, we can use Theorem 1 and the fact that ICAR is a sound approximation of CAR. \square

Example 15. To see why the CAR and ICAR semantics do not satisfy CONSISTENT SUPPORT, we consider another example KB set in the university domain:

$$\begin{aligned} \mathcal{T} &= \{\text{Prof} \sqsubseteq \text{Employee}, \text{Student} \sqsubseteq \text{GetDiscount}, \text{Student} \sqsubseteq \neg \text{Prof}\} \\ \mathcal{A} &= \{\text{Prof}(\text{sam}), \text{Student}(\text{sam})\} \end{aligned}$$

The TBox states that professors are employees, that students get discounted access to some university services, and that students and professors are disjoint classes. The ABox states that `sam` is both a student and a professor. This KB is inconsistent and has two repairs:

$$\mathcal{B}_1 = \{\text{Prof}(\text{sam})\} \quad \mathcal{B}_2 = \{\text{Student}(\text{sam})\}$$

We compute the result of applying the special closure operator on this KB:

$$\text{cl}_{\mathcal{T}}^*(\mathcal{A}) = \mathcal{A} \cup \{\text{Employee}(\text{sam}), \text{GetDiscount}(\text{sam})\}$$

As both new assertions in the closure can be added to the repairs without introducing any contradiction, we obtain two closed ABox repairs, namely:

$$\begin{aligned} \mathcal{R}_1 &= \{\text{Prof}(\text{sam}), \text{Employee}(\text{sam}), \text{GetDiscount}(\text{sam})\} \\ \mathcal{R}_2 &= \{\text{Student}(\text{sam}), \text{Employee}(\text{sam}), \text{GetDiscount}(\text{sam})\} \end{aligned}$$

Now consider the query $q(x) = \text{Employee}(x) \wedge \text{GetDiscount}(x)$, which asks for employees that get the discount. If we use either of the CAR and ICAR semantics, then we return `sam` as an answer:

$$\langle \mathcal{T}, \mathcal{A} \rangle \models_{\text{CAR}} q(\text{sam}) \quad \langle \mathcal{T}, \mathcal{A} \rangle \models_{\text{ICAR}} q(\text{sam})$$

since `Employee(sam)` and `GetDiscount(sam)` appear in both closed ABox repairs. Observe however that there is no consistent \mathcal{T} -support for $q(\text{sam})$ in \mathcal{A} . Indeed, we needed `Prof(sam)` to infer that `sam` is an employee and `Student(sam)` to infer that `sam` gets the discount, but according to the information in the TBox, these two assertions cannot both hold.

3.8 *k*-Lazy Semantics

Another parameterized family of inconsistency-tolerant semantics, the *k*-lazy semantics, was proposed in [36]. The semantics are based upon an alternative notion of a repair, called a *k*-lazy repair, which is obtained from the ABox

by removing for each *cluster* of contradictory assertions (defined next), either a minimal subset of at most k assertions that restores the consistency of the cluster, or the whole cluster if there is no such subset. Because of the possible removal of some entire clusters, lazy repairs need not be standard repairs.

Definition 14 (Clusters, k -lazy repairs). *Given a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, we let $\equiv_{\perp}^{\mathcal{K}}$ be the smallest equivalence relation over the assertions in \mathcal{A} satisfying the following condition: if assertions α, β appear together in a minimal \mathcal{T} -inconsistent subset, then $\alpha \equiv_{\perp}^{\mathcal{K}} \beta$. The clusters of \mathcal{K} are the equivalence classes of $\equiv_{\perp}^{\mathcal{K}}$.*

A k -lazy repair of \mathcal{K} is obtained by removing from \mathcal{A} for each cluster \mathcal{C}_i of \mathcal{K} : either (i) an inclusion-minimal subset $\mathcal{C}'_i \subseteq \mathcal{C}_i$ such that $|\mathcal{C}'_i| \leq k$ and $\mathcal{C}_i \setminus \mathcal{C}'_i$ is \mathcal{T} -consistent, or (ii) the whole cluster \mathcal{C}_i if there does not exist a set $\mathcal{C}'_i \subseteq \mathcal{C}_i$ satisfying these conditions.

The k -lazy semantics is then defined like the AR semantics, except that we use k -lazy repairs instead of the usual notion of repair.

Definition 15 (k -lazy semantics). *A tuple \mathbf{a} is an answer for a query q over a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ under k -lazy semantics, written $\langle \mathcal{T}, \mathcal{A} \rangle \models_{k\text{-lazy}} q(\mathbf{a})$, if and only if $\langle \mathcal{T}, \mathcal{R}_k \rangle \models q(\mathbf{a})$ for every k -lazy repair \mathcal{R}_k of \mathcal{K} .*

We illustrate the k -lazy semantics with an example:

Example 16. Returning to our running example, we first compute the clusters of $\mathcal{K}_{\text{univ}}$ (see Example 2 for the minimal $\mathcal{T}_{\text{univ}}$ -inconsistent subsets of $\mathcal{A}_{\text{univ}}$):

- {Prof(sam), Lect(sam), Fellow(sam)}
- {Prof(kim), Lect(kim)}
- {Fellow(julie), Teaches(csc343, julie)}

There is only one 0-lazy repair, obtained by removing all of the clusters. Observe that removing all clusters is the same as taking the intersection of the standard repairs, so we obtain the same results as with the IAR semantics. In the case of query q_1 , we get a single answer: $\langle \mathcal{T}_{\text{univ}}, \mathcal{A}_{\text{univ}} \rangle \models_{0\text{-lazy}} q_1(\text{alex})$.

Restoring the consistency of the cluster {Prof(kim), Lect(kim)} is done by removing one of its assertions. Thus, every k -lazy repair of $\langle \mathcal{T}_{\text{univ}}, \mathcal{A}_{\text{univ}} \rangle$ for $k \geq 1$ contains either Prof(kim) or Lect(kim), and thus, $\langle \mathcal{T}_{\text{univ}}, \mathcal{A}_{\text{univ}} \rangle \models_{k\text{-lazy}} q_1(\text{kim})$.

To restore the consistency of the cluster {Prof(sam), Lect(sam), Fellow(sam)}, it is sufficient and necessary to remove two assertions. The 2-lazy repairs all contain some cause for $q_1(\text{sam})$. It follows that $\langle \mathcal{T}_{\text{univ}}, \mathcal{A}_{\text{univ}} \rangle \models_{k\text{-lazy}} q_1(\text{sam})$ for $k \geq 2$, but $\langle \mathcal{T}_{\text{univ}}, \mathcal{A}_{\text{univ}} \rangle \not\models_{1\text{-lazy}} q_1(\text{sam})$ since the 1-lazy repairs are obtained by removing the whole cluster {Prof(sam), Lect(sam), Fellow(sam)}.

The following theorem relates the k -lazy semantics to other inconsistency-tolerant semantics. In particular, it shows that 0-lazy semantics coincides with the IAR semantics and that k -lazy semantics is an under-approximation of brave semantics, for every value of k :

Theorem 14.

1. $\mathcal{K} \models_{\text{IAR}} q(\mathbf{a})$ if and only if $\mathcal{K} \models_{0\text{-lazy}} q(\mathbf{a})$;
2. for every $k \geq 0$, if $\mathcal{K} \models_{k\text{-lazy}} q(\mathbf{a})$, then $\mathcal{K} \models_{\text{brave}} q(\mathbf{a})$;
3. for every KB \mathcal{K} , there exists some $k \geq 0$ such that for every $k' \geq k$:
 $\mathcal{K} \models_{\text{AR}} q(\mathbf{a})$ if and only if $\mathcal{K} \models_{k'\text{-lazy}} q(\mathbf{a})$.

The last item of the theorem shows that the k -lazy semantics converge to the AR semantics. However, these semantics are not sound approximations of the AR semantics, and unlike the k -support semantics, the convergence is not monotone in k . The following example illustrates these points:

Example 17. To show that $\mathcal{K} \models_{k\text{-lazy}} q$ does not imply that $\mathcal{K} \models_{k+1\text{-lazy}} q$, we consider the following KB:

$$\begin{aligned} \mathcal{T} &= \{\exists \text{Teaches} \sqsubseteq \text{Faculty}, \text{Student} \sqsubseteq \neg \text{Faculty}\} \\ \mathcal{A} &= \{\text{Teaches}(\text{sam}, \text{csc343}), \text{Teaches}(\text{sam}, \text{csc236}), \text{Student}(\text{sam})\} \end{aligned}$$

All assertions belong to the same cluster. There is only one 1-lazy repair

$$\{\text{Teaches}(\text{sam}, \text{csc343}), \text{Teaches}(\text{sam}, \text{csc236})\}$$

so $\mathcal{K} \models_{1\text{-lazy}} q_1(\text{sam})$. However, for $k \geq 2$, there are two k -lazy repairs:

$$\{\text{Teaches}(\text{sam}, \text{csc343}), \text{Teaches}(\text{sam}, \text{csc236})\} \quad \{\text{Student}(\text{sam})\}$$

We thus have $\mathcal{K} \not\models_{k\text{-lazy}} q_1(\text{sam})$ for every $k \geq 2$. Note that under AR semantics, **sam** is not an answer to q_1 , and so the answers obtained with 1-lazy semantics are not included in the answers obtained with AR semantics.

The k -lazy semantics do however verify both consistency properties.

Theorem 15. *The k -lazy semantics satisfy CONSISTENT SUPPORT and CONSISTENT RESULTS.*

Proof. For CONSISTENT SUPPORT, it suffices to notice that if $q(\mathbf{a})$ holds under k -lazy semantics, then there is a k -lazy repair \mathcal{R} for which $\langle \mathcal{T}, \mathcal{R} \rangle \models q(\mathbf{a})$. Since k -lazy repairs are \mathcal{T} -consistent by definition, it follows that every query answer has a consistent support.

To show CONSISTENT RESULTS, we can proceed in the same manner as for the AR and CAR semantics: pick any k -lazy repair \mathcal{R} of \mathcal{K} , and let \mathcal{I} be any model of the consistent KB $\langle \mathcal{T}, \mathcal{R} \rangle$. It follows from the definition of k -lazy semantics that if $\mathcal{K} \models_{k\text{-lazy}} q(\mathbf{a})$, then $\langle \mathcal{T}, \mathcal{R} \rangle \models q(\mathbf{a})$, and hence $\mathcal{I} \models q(\mathbf{a})$. \square

3.9 Preferred Repair Semantics

The notion of repair in Definition 4 integrates a very simple preference relation, namely set inclusion. When additional information on the reliability of ABox assertions is available, it is natural to use this information to identify preferred repairs, and to use these repairs as the basis for inconsistency-tolerant query answering. This idea has been explored in [9], where a preorder is used to identify the most preferred repairs:

Definition 16. Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a KB, and let \preceq be a preorder over subsets of \mathcal{A} . A subset $\mathcal{A}' \subseteq \mathcal{A}$ is a \preceq -repair of \mathcal{K} if it satisfies the following two conditions:

- \mathcal{A}' is \mathcal{T} -consistent
- there is no \mathcal{T} -consistent $\mathcal{A}'' \subseteq \mathcal{A}$ such that $\mathcal{A}'' \preceq \mathcal{A}'$ and $\mathcal{A}' \not\preceq \mathcal{A}''$

The set of \preceq -repairs of \mathcal{K} is denoted $Rep_{\preceq}(\mathcal{K})$.

With this new notion of \preceq -repair, we can revisit the different inconsistency-tolerant semantics. Following [9, 14], we will consider variants of the AR, IAR, and brave semantics based upon preferred repairs.

Definition 17. A tuple \mathbf{a} is an answer for a query q over $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$

- under the \preceq -AR semantics if $\langle \mathcal{T}, \mathcal{B} \rangle \models q(\mathbf{a})$ for every $\mathcal{B} \in Rep_{\preceq}(\mathcal{K})$;
- under the \preceq -IAR semantics if $\langle \mathcal{T}, \mathcal{B}_{\cap} \rangle \models q(\mathbf{a})$ where $\mathcal{B}_{\cap} = \bigcap_{\mathcal{B} \in Rep_{\preceq}(\mathcal{K})} \mathcal{B}$;
- under the \preceq -brave semantics if $\langle \mathcal{T}, \mathcal{B} \rangle \models q(\mathbf{a})$ for some $\mathcal{B} \in Rep_{\preceq}(\mathcal{K})$.

The preceding definitions can be instantiated with different preference relations \preceq . We consider the four types of preference relations studied in [9], which correspond to well-known methods of defining preferences over subsets, cf. [22].

Cardinality (\leq) A first possibility is to compare subsets using set cardinality:

$$\mathcal{A}_1 \leq \mathcal{A}_2 \quad \text{iff} \quad |\mathcal{A}_1| \leq |\mathcal{A}_2|$$

The resulting notion of \leq -repair is appropriate when all assertions are believed to have the same (small) likelihood of being erroneous, in which case repairs with the largest number of assertions are most likely to be correct.

Example 18. Reconsider the TBox $\mathcal{T}'_{\text{univ}}$ (which adds $\exists \text{Teaches} \sqsubseteq \text{Faculty}$ to $\mathcal{T}_{\text{univ}}$), and consider the ABox:

$$\begin{aligned} \mathcal{A} = \{ & \text{Fellow}(\text{julie}), \text{Prof}(\text{julie}), \text{Teaches}(\text{julie}, \text{csc486}), \text{Teaches}(\text{julie}, \text{csc236}), \\ & \text{Teaches}(\text{csc343}, \text{julie}), \text{Course}(\text{julie}) \} \end{aligned}$$

The repairs of $\langle \mathcal{T}'_{\text{univ}}, \mathcal{A} \rangle$ are:

$$\begin{aligned} \mathcal{R}_1 &= \{ \text{Fellow}(\text{julie}), \text{Teaches}(\text{julie}, \text{csc486}), \text{Teaches}(\text{julie}, \text{csc236}) \} \\ \mathcal{R}_2 &= \{ \text{Prof}(\text{julie}), \text{Teaches}(\text{julie}, \text{csc486}), \text{Teaches}(\text{julie}, \text{csc236}) \} \\ \mathcal{R}_3 &= \{ \text{Teaches}(\text{csc343}, \text{julie}), \text{Course}(\text{julie}) \} \end{aligned}$$

Only \mathcal{R}_1 and \mathcal{R}_2 are \leq -repairs. It follows that $\langle \mathcal{T}'_{\text{univ}}, \mathcal{A} \rangle \models_{\leq\text{-IAR}} q_2(\text{julie})$, while $\langle \mathcal{T}'_{\text{univ}}, \mathcal{A} \rangle \not\models_{\leq\text{-AR}} q_2(\text{julie})$.

Priority levels (\subseteq_P, \leq_P) The next preference relations assume that the ABox assertions can be partitioned into priority levels $\mathcal{P}_1, \dots, \mathcal{P}_n$ based on their perceived reliability, with assertions in \mathcal{P}_1 considered most reliable, and those in

\mathcal{P}_n least reliable. Such a prioritization can be used to separate a part of the dataset that has already been validated from more recent additions. Alternatively, one might assign assertions to priority levels based upon the concept or role names they use (when some predicates are known to be more reliable), or the data sources from which they originate (in information integration applications). Given such a prioritization $P = \langle \mathcal{P}_1, \dots, \mathcal{P}_n \rangle$ of \mathcal{A} , we can refine the \subseteq and \leq preorders by considering each priority level in turn:

- *Prioritized set inclusion*: $\mathcal{A}_1 \subseteq_P \mathcal{A}_2$ iff $\mathcal{A}_1 \cap \mathcal{P}_i = \mathcal{A}_2 \cap \mathcal{P}_i$ for every $1 \leq i \leq n$, or there is some $1 \leq i \leq n$ such that $\mathcal{A}_1 \cap \mathcal{P}_i \subsetneq \mathcal{A}_2 \cap \mathcal{P}_i$ and for all $1 \leq j < i$, $\mathcal{A}_1 \cap \mathcal{P}_j = \mathcal{A}_2 \cap \mathcal{P}_j$.
- *Prioritized cardinality*: $\mathcal{A}_1 \leq_P \mathcal{A}_2$ iff $|\mathcal{A}_1 \cap \mathcal{P}_i| = |\mathcal{A}_2 \cap \mathcal{P}_i|$ for every $1 \leq i \leq n$, or there is some $1 \leq i \leq n$ such that $|\mathcal{A}_1 \cap \mathcal{P}_i| < |\mathcal{A}_2 \cap \mathcal{P}_i|$ and for all $1 \leq j < i$, $|\mathcal{A}_1 \cap \mathcal{P}_j| = |\mathcal{A}_2 \cap \mathcal{P}_j|$.

Notice that a single assertion on level \mathcal{P}_i is preferred to any number of assertions from \mathcal{P}_{i+1} , so these preorders are best suited for cases in which there is a significant difference in the perceived reliability of adjacent priority levels.

Example 19. We use the same KB has in Example 18 with the following prioritization:

$$\begin{aligned} \mathcal{P}_1 &= \{\text{Prof}(\text{julie}), \text{Teaches}(\text{julie}, \text{csc486}), \text{Course}(\text{julie})\} \\ \mathcal{P}_2 &= \{\text{Fellow}(\text{julie})\} \\ \mathcal{P}_3 &= \{\text{Teaches}(\text{csc343}, \text{julie}), \text{Teaches}(\text{julie}, \text{csc236})\} \end{aligned}$$

The \subseteq_P -repairs are \mathcal{R}_2 and \mathcal{R}_3 (\mathcal{R}_1 is not a \subseteq_P -repair because $\mathcal{R}_1 \cap \mathcal{P}_1 \subsetneq \mathcal{R}_2 \cap \mathcal{P}_1$). Note that $\langle \mathcal{T}'_{\text{univ}}, \mathcal{A} \rangle \not\models_{\subseteq_P\text{-brave}} \text{Fellow}(\text{julie})$.

There is only one \leq_P -repair, namely \mathcal{R}_2 . It follows that *julie* is an answer to q_1 , q_2 and q_3 under \leq_P -IAR semantics.

Weights (\leq_w) The reliability of different assertions can also be modelled quantitatively by using a function $w : \mathcal{A} \rightarrow \mathbb{N}$ to assign weights to the ABox assertions. Such a weight function w induces a preorder \leq_w over subsets of \mathcal{A} in the expected way:

$$\mathcal{A}_1 \leq_w \mathcal{A}_2 \quad \text{iff} \quad \sum_{\alpha \in \mathcal{A}_1} w(\alpha) \leq \sum_{\alpha \in \mathcal{A}_2} w(\alpha)$$

If the ABox is populated using information extraction techniques, the weights may be derived from the confidence levels output by the extraction tool. Weight-based preorders can also be used in place of the \leq_P preorder to allow for compensation between the priority levels.

Example 20. If we assign a weight of 1 to the assertions of \mathcal{P}_3 , 2 to the assertions of \mathcal{P}_2 and 3 to the assertions of \mathcal{P}_1 in the ABox of Example 19, the weights of the repairs are 6 for \mathcal{R}_1 , 7 for \mathcal{R}_2 , and 4 for \mathcal{R}_3 . Only \mathcal{R}_2 is a \leq_w -repair.

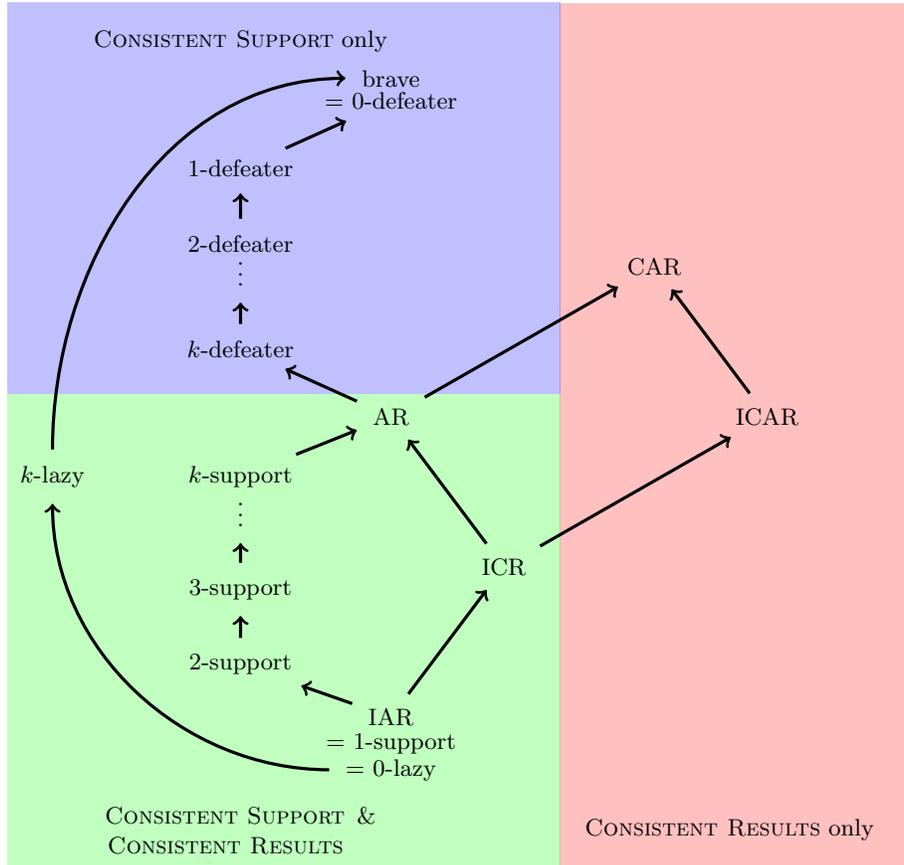


Fig. 1: Relationships between inconsistency-tolerant semantics.

3.10 Summary and Discussion

Figure 1 summarizes the relationships holding between the different semantics considered in this section. An arrow from S to S' indicates that S is a sound approximation of S' (or equivalently, that S' is an over-approximation of S). All relationships that hold can be derived from those indicated in the figure, so if there is no directed path from S to S' , this means that S is *not* an under-approximation of S' . We also indicate by means of the three shaded regions which consistency properties hold for the different semantics.

A recent work [6] has proposed a general framework for specifying different inconsistency-tolerant semantics in terms of pairs composed of a *modifier*, which creates a set of ABoxes from the original ABox, and an *inference strategy*, which states how query results are computed from these ABoxes. The authors consider (combinations of) three basic modifiers: *positive closure* (C), which adds to each ABox all facts that can be derived by applying the positive inclusions of the

TBox, *splitting into repairs* (R) of each ABox, and *selecting the cardinality-maximal ABoxes* (M). Four inference strategies are considered: *universal* (\forall), which requires a query answer to hold w.r.t. every ABox in the set, *existential* (\exists), which requires an answer to hold w.r.t. some ABox, *safe* (\cap), which considers the query answers that can be derived in the intersection of the ABoxes, *majority-based* (*maj*), which considers answers holding w.r.t. the majority of the ABoxes. We can thus express the AR semantics as the pair (R, \forall) : we first split into the different repairs, and then apply the universal inference strategy. The IAR, brave, and ICR semantics can be expressed respectively by the pairs (R, \cap) , (R, \exists) , and (CR, \cap) . For ICAR, CAR and k -lazy semantics, it is necessary to introduce further modifiers (e.g. adding the special closure operator).

4 Complexity of Inconsistency-Tolerant Query Answering

In this section, we will examine the computational properties of the inconsistency-tolerant semantics presented in the preceding subsection.

Our focus will be on description logics of the DL-Lite family [15], as they are a popular choice for OMQA and their computational properties have been more exhaustively studied than for other ontology languages. However, we will conclude the section with a brief overview of what is known about inconsistency-tolerant query answering with DLs outside the DL-Lite family.

As for the query language, we consider both CQs, which are the standard query language for OMQA, and the more restricted IQs. We note that all upper bounds obtained for IQs hold also for ground CQs.

4.1 Complexity Landscape for DL-Lite

In this subsection, we explore the complexity of inconsistency-tolerant query answering when the TBox is expressed in DL-Lite. The results we present (displayed in Figure 2) hold for common DL-Lite dialects like $DL\text{-Lite}_{core}$, $DL\text{-Lite}_{\mathcal{R}}$, and $DL\text{-Lite}_{\mathcal{A}}$, and more generally, for every DL-Lite dialect that possesses the following properties:

- every minimal support for $q(\mathbf{a})$ contains at most $|q|$ assertions;
- every minimal \mathcal{T} -inconsistent subset has cardinality at most two;
- CQ answering, IQ answering, and KB consistency can be performed by first-order query rewriting [15], and are thus in AC^0 in data complexity;
- CQ answering is NP-complete for combined complexity [15];
- IQ answering is NL-complete in combined complexity [3].

In what follows, we use the generic term ‘DL-Lite’ to mean any dialect with the preceding properties. We point out that the Horn versions of DL-Lite [3], which allow for conjunction (\sqcap), do not satisfy these conditions; interested readers can consult [13] for complexity results concerning those dialects.

We begin with the following lemma that gives the complexity of identifying assertions that occur in every repair and determining whether a subset of the ABox is a repair or equal to the intersection of all repairs.

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
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
ICR	coNP	coNP	$\Delta_2^P[O(\log n)]$	coNP
CAR	coNP	in AC^0	Π_2^P	NL
ICAR	in AC^0	in AC^0	NP	NL
k-lazy ($k \geq 1$)	coNP	in P	Π_2^P	in P

Fig. 2: Complexity of inconsistency-tolerant query answering in DL-Lite. All results are completeness results unless otherwise indicated.

Lemma 1. *In DL-Lite, it can be decided in NL in combined complexity whether an assertion $\alpha \in \mathcal{A}$ belongs to every repair of \mathcal{A} w.r.t. \mathcal{T} , and in P whether a subset $\mathcal{A}' \subseteq \mathcal{A}$ is (i) a repair of \mathcal{A} w.r.t. \mathcal{T} , or (ii) is equal to the intersection of all repairs of \mathcal{A} w.r.t. \mathcal{T} .*

Proof. For the first statement, we give an NL algorithm for deciding whether α does *not* belong to every repair of \mathcal{A} w.r.t. \mathcal{T} , and then use the fact that $NL=coNL$ to obtain an NL algorithm for the complement. To show that α is not present in every repair, it suffices to guess an assertion $\beta \in \mathcal{A}$ and show that $\{\alpha, \beta\}$ is \mathcal{T} -inconsistent. To see why, note that an assertion is omitted from a repair only if its introduction leads to a contradiction, and for DL-Lite KBs, it is well known that every minimal \mathcal{T} -inconsistent subset of an ABox contains at most two assertions. As consistency checking can be tested in NL, we obtain the required NL procedure.

To decide whether $\mathcal{A}' \subseteq \mathcal{A}$ is a repair of \mathcal{A} w.r.t. \mathcal{T} , it suffices to verify that \mathcal{A}' is \mathcal{T} -consistent and that for every $\alpha \in \mathcal{A} \setminus \mathcal{A}'$, the ABox $\mathcal{A}' \cup \{\alpha\}$ is \mathcal{T} -inconsistent. As consistency checking can be performed in NL (hence P) in combined complexity, we obtain a P procedure for repair checking. To decide whether \mathcal{A}' is equal to the intersection of all repairs, we use the NL procedure described in the previous paragraph to verify that every assertion in \mathcal{A}' belongs to every repair and that every assertion in $\mathcal{A} \setminus \mathcal{A}'$ does not belong to every repair. \square

We now pass in review the different semantics from the preceding section, starting with the AR semantics, which we show to be intractable even for instance queries:

Theorem 16 ([32]). *Under AR semantics, CQ answering over DL-Lite KBs is coNP-complete in data complexity. The same holds for IQs.*

Proof. For the upper bound, we consider the following non-deterministic procedure for showing that $\mathcal{K} \not\models_{\text{AR}} q(\mathbf{a})$: guess a subset $\mathcal{R} \subseteq \mathcal{A}$ and check that it is a repair of \mathcal{K} and such that $\langle \mathcal{T}, \mathcal{A} \rangle \not\models q(\mathbf{a})$. As repair checking and CQ answering (under classical semantics) are both in P for data complexity for DL-Lite KBs, the described procedure runs in non-deterministic polynomial time in $|\mathcal{A}|$.

The lower bound can be shown by a simple reduction from propositional unsatisfiability (UNSAT) [8]. Let $\varphi = c_1 \wedge \dots \wedge c_m$ be a propositional formula over variables v_1, \dots, v_k , where each c_i is a propositional clause. We use the following DL-Lite knowledge base, where the ABox encodes φ and the TBox does not depend on φ :

$$\begin{aligned} \mathcal{T} &= \{ \exists P^- \sqsubseteq \neg \exists N^-, \exists P \sqsubseteq \neg \exists U^-, \exists N \sqsubseteq \neg \exists U^-, \exists U \sqsubseteq A \} \\ \mathcal{A} &= \{ U(a, c_i) \mid 1 \leq i \leq m \} \cup \{ P(c_i, v_j) \mid v_j \in c_i \} \cup \{ N(c_i, v_j) \mid \neg v_j \in c_i \} \end{aligned}$$

It is not hard to verify that φ is unsatisfiable if and only if $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\text{AR}} A(a)$. Indeed, because of the axiom $\exists P^- \sqsubseteq \neg \exists N^-$, each repair corresponds to a valuation of the variables in which v_j assigned true if it has an incoming P -edge in the repair. If a clause c_i is not satisfied by the valuation encoded by the repair, then the individual c_i will have no outgoing P - or N -edges, and hence the repair will contain the assertion $U(a, c_i)$. Thus, if φ is unsatisfiable, every repair will contain an assertion $U(a, c_i)$, and so $A(a)$ will be entailed under AR semantics. Conversely, if $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\text{AR}} A(a)$, then the assertions $U(a, c_i)$ that are present in the repairs witness that the corresponding valuations each contain at least one unsatisfied clause. \square

We observe that in the preceding coNP-hardness proof, we needed to use DL-Lite axioms involving existential concepts. However, if we consider CQs, then we need only a single disjointness statement to obtain intractability.

Theorem 17 ([8]). *For any ontology language that allows for class disjointness ($A \sqsubseteq \neg B$), CQ answering under AR semantics is coNP-hard in data complexity.*

Proof. This time we use a restricted version of UNSAT, called 2+2UNSAT, proven coNP-hard in [20], in which each clause has 2 positive and 2 negative literals, and any of the four positions in a clause can be filled instead by one of the truth constants **true** and **false**. Consider an instance $\varphi = c_1 \wedge \dots \wedge c_m$ of 2+2-UNSAT over v_1, \dots, v_k , **true**, and **false**. We let $\mathcal{T} = \{T \sqsubseteq \neg F\}$, and encode φ in the ABox as follows:

$$\begin{aligned} &\{ P_1(c_i, u), P_2(c_i, x), N_1(c_i, y), N_2(c_i, z) \mid c_i = u \vee x \vee \neg y \vee \neg z, 1 \leq i \leq m \} \\ &\cup \{ T(v_j), F(v_j) \mid 1 \leq j \leq k \} \cup \{ T(\mathbf{true}), F(\mathbf{false}) \} \end{aligned}$$

It can be shown that φ is unsatisfiable just in the case that the following Boolean query is entailed from $\langle \mathcal{T}, \mathcal{A} \rangle$ under AR semantics:

$$\begin{aligned} \exists x, y_1, \dots, y_4 \quad &P_1(x, y_1) \wedge F(y_1) \wedge P_2(x, y_2) \wedge F(y_2) \\ &\wedge N_1(x, y_3) \wedge T(y_3) \wedge N_2(x, y_4) \wedge T(y_4) \end{aligned}$$

Intuitively, the axiom $T \sqsubseteq \neg F$ selects a truth value for each variable, so the repairs of \mathcal{A} correspond exactly to the set of valuations. Importantly, there is only one way to avoid satisfying a 2+2-clause: the first two variables must be assigned false and the last two variables must be assigned true. Thus, q holds in a repair iff there is an unsatisfied clause in the corresponding valuation. \square

For CQ answering, switching from data to combined complexity takes us one level higher in the polynomial hierarchy, while for IQs, we have the same complexity for both measures:

Theorem 18 ([13]). *Under AR semantics, CQ answering over DL-Lite KBs is Π_2^p -complete in combined complexity; for IQ answering, the combined complexity drops to coNP-complete.*

Proof. For the membership results, we can use the same non-deterministic procedure as in Theorem 16 for testing whether a tuple \mathbf{a} is *not* an answer: guess a subset \mathcal{R} of \mathcal{A} and check that it is a repair and satisfies $\langle \mathcal{T}, \mathcal{R} \rangle \not\models q(\mathbf{a})$. Since the latter checks can be performed in coNP (resp. in P if q is an IQ) in combined complexity, we obtain a Σ_2^p (resp. NP) procedure for deciding $\mathcal{K} \not\models_{\text{AR}} q(\mathbf{a})$ when q is a CQ (resp. IQ). It follows that the original problem is in Π_2^p for CQs, and in coNP for IQs.

The lower bound for instance queries follows from Theorem 16. For CQs, Π_2^p -hardness is shown in [13] by the following reduction from validity of $\text{QBF}_{2,\vee}$ formulas. Let $\varphi = \forall x_1, \dots, x_n \exists y_1, \dots, y_m \bigwedge_{j=1}^k c_j$ where $\bigwedge_{j=1}^k c_j$ is a 3CNF formula over the variables $x_1, \dots, x_n, y_1, \dots, y_m$, where every c_j is a clause of the form $\ell_j^1 \vee \ell_j^2 \vee \ell_j^3$. The variable of literal ℓ_j^p is denoted by $v(\ell_j^p)$. For example, for the clause $\neg x_1 \vee y_2 \vee \neg y_1$, we have $v(\neg x_1) = x_1$, $v(y_2) = y_2$, and $v(\neg y_1) = y_1$.

To decide whether φ is valid, we construct in polynomial time the following KB and Boolean query:

$$\begin{aligned} \mathcal{T} &= \{ \exists GX_i \sqsubseteq GX_i^- \mid 1 \leq i \leq n \} \\ \mathcal{A} &= \bigcup_{j=1}^k \{ L_j^1(c_j^V, V(v(\ell_j^1))), L_j^2(c_j^V, V(v(\ell_j^2))), L_j^3(c_j^V, V(v(\ell_j^3))) \mid \\ &\quad V \text{ is a valuation of } v(\ell_j^1), v(\ell_j^2), v(\ell_j^3) \text{ satisfying } C_j \} \\ &\quad \cup \{ GX_i(0, 1), GX_i(1, 0) \mid 1 \leq i \leq n \} \\ q &= \bigwedge_{j=1}^k \bigwedge_{h=1}^3 L_j^h(w_j, v(\ell_j^h)) \wedge \bigwedge_{i=1}^n GX_i(x_i, z_i) \end{aligned}$$

where all variables in q are existentially quantified. It is shown in [13] that φ is valid iff $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\text{AR}} q$. Intuitively, the assertion $GX_i(0, 1)$ (resp. $GX_i(1, 0)$) means that $x_i = 0$ (resp. $x_i = 1$), and the repairs of $\langle \mathcal{T}, \mathcal{A} \rangle$ are in one-to-one correspondence with the valuations of x_1, \dots, x_n since every repair must contain exactly one of assertions $GX_i(0, 1)$ and $GX_i(1, 0)$. The query q looks for a valuation of $x_1, \dots, x_n, y_1, \dots, y_m$ that satisfies $\bigwedge_{j=1}^k c_j$. More precisely, the variables w_1, \dots, w_k correspond to partial valuations that satisfy c_1, \dots, c_k respectively and

will be mapped to some individuals $c_1^{V_1}, \dots, c_k^{V_k}$, and each $v(\ell_j^h)$ is a variable x_i or y_i that will be mapped to either 0 or 1. If q is entailed under AR semantics, then for every valuation μ of x_1, \dots, x_n , we can define a satisfying valuation μ' for $x_1, \dots, x_n, y_1, \dots, y_m$ by examining the way that q is mapped into the repair corresponding to μ . Conversely, if the QBF is valid, then every valuation μ for x_1, \dots, x_n can be extended to a valuation μ' of $x_1, \dots, x_n, y_1, \dots, y_m$ that satisfies φ , so in every repair of \mathcal{K} , there will be a way of mapping the query. \square

The intractability of querying under AR semantics motivated the introduction of other semantics with more favourable computational properties. In particular, it led to the proposal of the IAR semantics, for which the complexity of query answering matches that of classical semantics:

Theorem 19 ([33, 34]). *Under IAR semantics, CQ answering over DL-Lite KBs is in AC^0 in data complexity and NP-complete in combined complexity, and IQ answering is NL-complete in combined complexity.*

Proof. The AC^0 membership result is proven by means of a query rewriting procedure. As shown in [33, 34], it is possible, for every DL-Lite TBox \mathcal{T} and CQ q , to construct an FO-query q' such that for every ABox \mathcal{A} and candidate answer \mathbf{a} , we have the following:

$$\langle \mathcal{T}, \mathcal{A} \rangle \models_{\text{IAR}} q(\mathbf{a}) \quad \iff \quad \mathcal{I}_{\mathcal{A}} \models q'(\mathbf{a})$$

As the rewriting is independent of the ABox, the preceding equivalence shows that the data complexity of CQ answering under IAR semantics is the same as for FO-query evaluation, namely, in AC^0 . It is beyond the scope of this chapter to detail the rewriting algorithm, but we give the intuition in Example 21.

The NP upper bound in combined complexity can be obtained as follows: guess a subset of $\mathcal{A}' \subseteq \mathcal{A}$ and a polynomial-size proof that $\langle \mathcal{T}, \mathcal{A}' \rangle \models q(\mathbf{a})$ and check that every assertion α in \mathcal{A}' belongs to the intersection of repairs. By Lemma 1, we know that the latter check can be performed in polynomial time. If q is a instance query, then $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\text{IAR}} q(\mathbf{a})$ iff there is an assertion $\alpha \in \mathcal{A}$ such that (i) $\langle \mathcal{T}, \{\alpha\} \rangle \models q(\mathbf{a})$, and (ii) α appears in every repair in $\text{Rep}\langle \mathcal{T}, \mathcal{A} \rangle$. It thus suffices to iterate (in non-logarithmic space) over the (binary encodings of) ABox assertions and check (in NL) whether the latter conditions are verified. \square

The following example illustrates the use of query rewriting to compute the answers under IAR semantics:

Example 21. The general idea is to add to the classical rewriting expressions that ensure that the assertions used to derive the query are not contradicted by other assertions in the ABox. For instance, we have seen in Example 4 that

$$q'_2(x) = \text{Prof}(x) \vee \text{Lect}(x) \vee \exists y. \text{Teaches}(x, y)$$

is a rewriting of $q_2(x) = \exists y \text{Teaches}(x, y)$ w.r.t. $\mathcal{T}_{\text{univ}}$. To transform q'_2 into a rewriting of q_2 for the IAR semantics, we add to every atom in the rewriting

a formula that ensures that there is no assertion in the ABox that will conflict with the (instantiation of) the atom. This yields the following query:

$$\begin{aligned} q_2''(x) = & \text{Prof}(x) \wedge (\neg \text{Lect}(x) \wedge \neg \text{Fellow}(x) \wedge \neg \text{Course}(x) \wedge \neg \exists z. \text{Teaches}(z, x)) \\ & \vee \text{Lect}(x) \wedge (\neg \text{Prof}(x) \wedge \neg \text{Fellow}(x) \wedge \neg \text{Course}(x) \wedge \neg \exists z. \text{Teaches}(z, x)) \\ & \vee \exists y. (\text{Teaches}(x, y) \wedge (\neg \text{Prof}(y) \wedge \neg \text{Lect}(y) \wedge \neg \text{Fellow}(y))) \end{aligned}$$

Observe, for example, that in the first line, we have added negations of the four types of atoms that can contradict the first disjunct $\text{Prof}(x)$. This disjunct in the first line states that an individual a is an answer to $q_2(x)$ under the IAR semantics if the assertion $\text{Prof}(a)$ is present in the ABox *and* the ABox does not contain $\text{Lect}(a)$, $\text{Fellow}(a)$, $\text{Course}(a)$, or any assertion of the form $\text{Teaches}(-, a)$.

Like the IAR semantics, we can show that querying DL-Lite KBs under brave semantics can be carried out by means of query rewriting and has the same data and combined complexity as classical semantics.

Theorem 20 ([13]). *Under brave semantics, CQ answering over DL-Lite KBs is in AC^0 in data complexity and NP-complete in combined complexity, and IQ answering is NL-complete in combined complexity.*

We omit the proof (which is a special case of Theorems 21 and 22) and instead give an example which illustrates how query rewriting works for brave semantics:

Example 22. As with the IAR semantics, we start with a standard rewriting of the query for classical semantics, represented as a union of CQs. We then add to each disjunct the necessary constraints to force that the set of ABox assertions witnessing the satisfaction of the disjunct be consistent with the TBox. For example, consider again the modified TBox $\mathcal{T}'_{\text{univ}}$ (which adds the inclusion $\exists \text{Teaches} \sqsubseteq \text{Faculty}$ to $\mathcal{T}_{\text{univ}}$) and the query $q_2 = \exists y \text{Teaches}(x, y)$. It can be verified that the query $q_2'(x) = \text{Prof}(x) \vee \text{Lect}(x) \vee \exists y. \text{Teaches}(x, y)$ is a rewriting of q_2 w.r.t. $\mathcal{T}'_{\text{univ}}$. We observe that if we find an assertion $\text{Prof}(a)$ or $\text{Lect}(a)$ in the ABox, then a is indeed an answer to q_2 under brave semantics, as the single-element set $\{\text{Prof}(a)\}$ and $\{\text{Lect}(a)\}$ are both $\mathcal{T}'_{\text{univ}}$ -consistent. Suppose however that we match the third disjunct using the assertion $\text{Teaches}(a, a)$. This does *not* prove that a is a brave answer to q_2 , as $\{\text{Teaches}(a, a)\}$ is $\mathcal{T}'_{\text{univ}}$ -inconsistent (as this would imply that a belongs to both Faculty and Course). To prevent such false matches, we must add an extra condition to the final disjunct to prevent x and y from being mapped to the same individual. One can show that the resulting query

$$q_2'''(x) = \text{Prof}(x) \vee \text{Lect}(x) \vee \exists y. (\text{Teaches}(x, y) \wedge x \neq y)$$

is a rewriting of q_2 w.r.t. $\mathcal{T}'_{\text{univ}}$ under the brave semantics.

The k -support and k -defeater semantics were introduced to obtain finer approximations of the AR semantics, while keeping the same desirable computational properties as the IAR and brave semantics. Indeed, as the following

theorems show, for every value of k , query answering under the k -support and k -defeater semantics has the same complexity as for classical semantics and can be performed using query rewriting.

Theorem 21 ([13]). *In DL-Lite, CQ answering under the k -support semantics is in AC^0 w.r.t. data complexity, for every $k \geq 1$. The same holds for the k -defeater semantics, for every $k \geq 0$.*

Proof (idea). We consider first the k -support semantics. In [13], the authors show how to construct, for every $k \geq 1$, every DL-Lite³ \mathcal{T} , and for every CQ q , an FO-query q' with the following properties:

$$\langle \mathcal{T}, \mathcal{A} \rangle \models_{k\text{-supp}} q(\mathbf{a}) \quad \iff \quad \mathcal{I}_{\mathcal{A}} \models q'(\mathbf{a})$$

Membership in AC^0 follows immediately. The rewriting q' takes the form of a big disjunction $\psi_1 \vee \dots \vee \psi_n$, where the disjuncts ψ_i correspond to the different possible choices of k supports for q of cardinality at most $|q|$, and each ψ_i asserts that the chosen supports are present in \mathcal{A} and that there is no \mathcal{T} -consistent subset of \mathcal{A} of cardinality at most k which conflicts with each of the supports. Since the rewriting is ABox-independent, the disjuncts do not mention the individuals involved in the supports, but just the types of atoms involved, and the (in)equalities holding between arguments of different atoms.

An analogous result is shown in [13] for the k -defeater semantics. In this case, the rewriting takes the form $\neg(\kappa_1 \vee \dots \vee \kappa_n)$, where every κ_i asserts the existence of a \mathcal{T} -consistent set of facts of cardinality at most k which conflicts with every minimal \mathcal{T} -support for q (the negation ensures that no such set is present in the ABox). Here we again utilize the fact that for DL-Lite KBs, the size of minimal \mathcal{T} -supports for q is bounded by $|q|$, and hence there are only finitely many types of supports to consider. \square

Theorem 22 ([13]). *In DL-Lite, CQ (resp. IQ) answering under the k -support semantics is in NP-complete (resp. NL-complete) w.r.t. combined complexity, for every $k \geq 1$. The same holds for the k -defeater semantics, for every $k \geq 0$.*

Proof. The lower bounds are inherited from the analogous results for classical semantics, so we only present proofs for the upper bounds, closely following the treatment in [13].

We begin by defining an NP procedure for deciding whether $\langle \mathcal{T}, \mathcal{A} \rangle \models_{k\text{-supp}} q(\mathbf{a})$ when q is a CQ. In the first step, we guess subsets S_1, \dots, S_k of \mathcal{A} , as well as polysize certificates that $\langle \mathcal{T}, S_i \rangle \models q(\mathbf{a})$, for each S_i . We next verify (in P) that each S_i is \mathcal{T} -consistent and that each certificate is valid. Finally, we check that every repair contains some S_i by considering all subsets $U \subseteq \mathcal{A}$ with $|U| \leq k$ and testing whether U is (i) \mathcal{T} -consistent and (ii) $\langle \mathcal{T}, U \cup S_i \rangle \models \perp$ for every $1 \leq i \leq k$. If no such a set is found, the procedure outputs yes, and otherwise it

³ In fact, the rewritability results in [13] are proven for all ontology languages for which CQ answering and unsatisfiability testing can be performed via UCQ _{\neq} -rewriting. As DL-Lite satisfies these conditions, the results apply to DL-Lite KBs.

outputs no. Since we take k to be a fixed constant, the described procedure runs in non-deterministic polynomial time. Regarding correctness, recall that since \mathcal{T} is a DL-Lite TBox, every minimal \mathcal{T} -inconsistent subset of \mathcal{A} contains at most 2 assertions. Hence, if there is some repair \mathcal{R} that does not contain any S_i , then we can find a subset $U \subseteq \mathcal{R}$ of cardinality at most k which contradicts every S_i .

Next we define an NL procedure for IQ answering under k -support semantics. The procedure begins by guessing k assertions $\alpha_1, \dots, \alpha_k$ from \mathcal{A} (intuitively: these are the k supports). Observe that by using a binary encoding, we need only logarithmic space to store this guess. We verify next that $\{\alpha_i\}$ is \mathcal{T} -consistent and $\langle \mathcal{T}, \{\alpha_i\} \rangle \models q(\mathbf{a})$ for every $1 \leq i \leq k$, outputting no if one of these conditions fails for some α_i . Both checks can be done in NL, since satisfiability and instance checking in DL-Lite are NL-complete w.r.t. combined complexity. Finally we need to make sure that every repair $R \in \text{Rep}(\mathcal{K})$ contains one of the α_i . As every minimal \mathcal{T} -inconsistent subset contains at most two assertions, it follows that there exists some repair which *does not* contain any α_i iff there is a subset $U \subseteq \mathcal{A}$ of cardinality at most k which is \mathcal{T} -consistent and such that $\langle \mathcal{T}, U \cup \{\alpha_i\} \rangle \models \perp$ for every α_i . We can therefore decide in NL whether such a repair exists by guessing such a subset U and checking whether it satisfies the required conditions. Using the fact that $\text{NL} = \text{coNL}$, we obtain a NL procedure for testing whether there is no such subset. If the check succeeds, we return yes, else no. It is easy to see that $\langle \mathcal{T}, \mathcal{A} \rangle \models_{k\text{-supp}} q(\mathbf{a})$ iff some execution of the procedure returns yes.

We next consider CQ answering under k -defeater semantics. Let S_1, \dots, S_m be an enumeration of the \mathcal{T} -consistent subsets of \mathcal{A} having cardinality at most k ; observe that m is polynomial in $|\mathcal{A}|$ since k is fixed. We know that $\langle \mathcal{T}, \mathcal{A} \rangle \models_{k\text{-def}} q(\mathbf{a})$ iff for every S_i , there is some \mathcal{T} -support C of $q(\mathbf{a})$ such that $S_i \cup C$ is \mathcal{T} -consistent. The first step in the procedure is thus to guess a sequence C_1, \dots, C_m of subsets of \mathcal{A} , one for each S_i , together with polysize certificates that $\langle \mathcal{T}, C_i \rangle \models q(\mathbf{a})$ for each C_i . In a second step, the procedure checks in polytime that for every $1 \leq i \leq m$, the certificate is valid and $S_i \cup C_i$ is \mathcal{T} -consistent. If all checks succeed, the procedure outputs yes, and otherwise it outputs no. It is easily verified that $\langle \mathcal{T}, \mathcal{A} \rangle \models_{k\text{-def}} q(\mathbf{a})$ iff there is an execution of this procedure that returns yes.

To complete the proof, we describe an NL procedure for checking whether $\langle \mathcal{T}, \mathcal{A} \rangle \not\models_{k\text{-def}} q(\mathbf{a})$ when q is an IQ (we can then use $\text{NL} = \text{coNL}$ to obtain an NL algorithm for the complement). The first step is to guess a (binary encoding of a) subset $S \subseteq \mathcal{A}$ of cardinality at most k . We then test in NL whether S is \mathcal{T} -consistent, and return no if not. Next we need to verify that $S \cup \{\alpha\}$ is \mathcal{T} -inconsistent for every minimal \mathcal{T} -support $\{\alpha\}$ of q , returning yes if this is the case, and no otherwise. This can be done by defining a NL procedure for the complementary problem, which works by guessing $\alpha \in \mathcal{A}$ and verifying that $\{\alpha\}$ is a \mathcal{T} -support of q and that $S \cup \{\alpha\}$ is \mathcal{T} -consistent. \square

We next consider the ICR semantics. While query answering remains coNP-complete in data complexity, we observe that the combined complexity for CQ answering is lower than for AR semantics ($\Delta_2^P[O(\log n)]$ rather than Π_2^P). Intuitively, this difference can be explained by the fact that ICR semantics is amenable to preprocessing. Indeed, if we compute and store in a query-

independent offline phase the intersection of the closure of the repairs, then at query time we can use classical querying algorithms on the resulting ABox to obtain the answers under the ICR semantics.

Theorem 23. *Under ICR semantics, CQ answering over DL-Lite KBs is coNP-complete in data complexity, and $\Delta_2^P[O(\log n)]$ -complete in combined complexity; IQ answering is coNP-complete for both data and combined complexity.*

Proof. As the ICR and AR semantics coincide for IQs (Theorem 10), we can reuse the proof of Theorem 16 to show that IQ answering is coNP-complete for both data and combined complexity.

To decide if a CQ is entailed under ICR semantics, we first use a coNP-oracle to decide, for every assertion that can be constructed using the vocabulary and individuals present in the KB, whether the assertion is entailed under AR semantics. Then we use a NP-oracle to decide if the query holds w.r.t. the ABox consisting of all assertions that are entailed under AR semantics. This yields a Δ_2^P upper bound. We further observe that the oracle calls can be structured as a tree, and thus we can apply results in [26] to obtain membership in $\Delta_2^P[O(\log n)]$.

A matching $\Delta_2^P[O(\log n)]$ lower bound can be proven by a rather involved reduction from the Parity(3SAT) problem (cf. [45, 23]), which builds upon a similar reduction from [13]. A detailed proof is given in Chapter 2 of [14]. \square

It turns out that if we instead adopt the ICAR semantics, then we can once again employ query rewriting and match the complexity of classical semantics.

Theorem 24 ([32, 33]). *Under ICAR semantics, CQ answering over DL-Lite KBs is in AC^0 in data complexity, and NP-complete in combined complexity; IQ answering is NL-complete in combined complexity.*

Proof. The lower bounds are again inherited from existing results for classical semantics. To show membership in AC^0 for data complexity, the authors of [33] show how to define, for every TBox \mathcal{T} and every CQ q , an FO-query q' with the following properties:

$$\langle \mathcal{T}, \mathcal{A} \rangle \models_{ICAR} q(\mathbf{a}) \quad \iff \quad \mathcal{I}_{\mathcal{A}} \models q'(\mathbf{a})$$

The idea is similar to the rewriting for IAR semantics, except that in addition to adding constraints to each atom in the standard rewriting, we must also rewrite each atom in order to take into account the different assertions that could cause the atom to appear in $cl_{\mathcal{T}}^*(\mathcal{A})$. We refer the reader to [33] for details.

Analogously to Lemma 1, one can show that deciding whether an assertion appears in every closed ABox repair can be done in NL in combined complexity. It follows that IQ answering under ICAR semantics is in NL: simply check that the assertion in question belongs to every closed ABox repair. For CQs, we can first compute the set of assertions appearing in every closed ABox repair (this can be done in P), and then perform standard CQ answering on the resulting ABox (whose combined complexity is in NP). \square

For the CAR semantics, the results are mixed: for IQs, the complexity is the same as for classical semantics, while for CQs, we have the same (high) complexity as with AR semantics.

Theorem 25 ([32, 33]). *Under CAR semantics, CQ answering over DL-Lite KBs is coNP-complete in data complexity and Π_2^P -complete in combined complexity; IQ answering is in AC^0 in data complexity and NL-complete in combined complexity.*

Proof. As the CAR semantics coincides with the ICAR semantics on ground CQs (Theorem 12), the results for IQs follow immediately from the analogous results for the ICAR semantics (Theorem 24).

For the coNP membership result, we observe that to show $\langle \mathcal{T}, \mathcal{A} \rangle \not\models_{CAR} q(\mathbf{a})$, it suffices to guess a set \mathcal{R} of ABox assertions built using the vocabulary and individuals from \mathcal{K} , and then check (in P in data complexity) that \mathcal{R} is a closed ABox repair and $\langle \mathcal{T}, \mathcal{R} \rangle \not\models q(\mathbf{a})$. For the coNP lower bound, we can reuse the reduction from the proof of Theorem 17. Indeed, since the TBox from that reduction contains no positive inclusions, the CAR and AR semantics give the same answers.

The combined complexity of CQ answering is not considered in [32, 33]), but membership in Π_2^P can be shown using the same approach as for data complexity: guess a set of assertions, and verify that the guessed set is a closed ABox repair that does not yield the query answer. The latter verifications involve a polynomial number of NL consistency checks, as well as a coNP check that the considered tuple is not an answer under classical semantics. For the lower bound, we can reuse the reduction used to show Π_2^P -hardness of CQ answering under AR semantics (Theorem 18), as the TBox only contains negative inclusions. \square

A tractability result for IQ answering⁴ under k -lazy semantics is presented in [36] for ontologies expressed in linear Datalog^{+/-}, which generalizes DL-Lite.

Theorem 26. *For every $k \geq 0$, IQ answering over DL-Lite KBs under k -lazy semantics is in P for data and combined complexity.*

Proof. Following the approach from [36], we first compute the clusters $\mathcal{C}_1, \dots, \mathcal{C}_m$ of the KB \mathcal{K} , and then for each cluster \mathcal{C}_i , we compute all inclusion-minimal subsets $\mathcal{C}'_i \subseteq \mathcal{C}_i$ of size at most k for which $\mathcal{C}_i \setminus \mathcal{C}'_i$ is \mathcal{T} -consistent. Let $\mathcal{C}_i^1, \dots, \mathcal{C}_i^{\ell_i}$ be the subsets computed for cluster \mathcal{C}_i , and if no such subset exists, set $\mathcal{C}_i^1 = \mathcal{C}_i$. We then observe that because we are in DL-Lite (and so every IQ answer can be inferred from a single fact), there exists a k -lazy repair \mathcal{R} such that $\langle \mathcal{T}, \mathcal{R} \rangle \not\models q(\mathbf{a})$ if and only if:

- $\langle \mathcal{T}, \mathcal{A} \setminus \bigcup_i \mathcal{C}_i \rangle \not\models q(\mathbf{a})$, and
- for every cluster \mathcal{C}_i , there exists some \mathcal{C}_i^j such that $\langle \mathcal{T}, \mathcal{C}_i \setminus \mathcal{C}_i^j \rangle \not\models q(\mathbf{a})$.

⁴ The formulation of the results in [36] suggests that CQ answering is also in P for data complexity, but as Theorem 27 shows, this is not the case.

We can thus examine each cluster \mathcal{C}_i in turn and see whether such a \mathcal{C}_i^j exists for every cluster \mathcal{C}_i . To complete the argument, we note that the described procedure runs in polynomial time in combined complexity, when k is a fixed constant. \square

However, for conjunctive queries, there is no improvement in complexity compared to AR semantics:

Theorem 27. *For every $k \geq 1$, CQ answering over DL-Lite KBs under k -lazy semantics is coNP-complete in data complexity and Π_2^P -complete in combined complexity.*

Proof. The upper bounds employ the usual strategy: to show $\langle \mathcal{T}, \mathcal{A} \rangle \not\models_{k\text{-lazy}} q(\mathbf{a})$, we guess a subset $\mathcal{R} \subseteq \mathcal{A}$ and verify that \mathcal{R} is a k -lazy repair and $\langle \mathcal{T}, \mathcal{R} \rangle \not\models q(\mathbf{a})$. We note that for DL-Lite KBs, the latter checks can be performed in P in data complexity and in NP in combined complexity.

As noted in [13], the reduction from [8] used to show coNP-hardness of CQ answering under AR semantics (presented in the proof of Theorem 17) can be used without any modification to prove coNP-hardness of CQ answering under k -lazy semantics, for every $k \geq 1$. Indeed, the clusters of the KB used in that reduction take the form $\{T(v_j), F(v_j)\}$, and since a single assertion needs to be removed from each cluster to restore consistency, the k -lazy repairs are the same as the usual repairs when $k \geq 1$.

For the Π_2^P -hardness result, we can similarly reuse the proof of the corresponding result for AR semantics. Indeed, the KB used in the reduction from [13] (see proof of Theorem 18) has clusters of the form $\{GX_i(0, 1), GX_i(1, 0)\}$, and thus for this KB, the k -lazy and AR semantics coincide starting from $k = 1$. \square

We close the subsection with the following theorem which shows how the data complexity of the AR, brave, and IAR semantics changes depending on the notion of preferred repair. Interestingly, the complexity depends mainly on the choice of preference relation, rather than on the base semantics. In all cases, adding preferences leads to intractability, but in the case of prioritized set inclusion (\subseteq_P), the coNP data complexity is no worse than for (plain) AR semantics.

Theorem 28 ([9, 14]). *For $S \in \{AR, IAR, brave\}$, the data complexity of CQ answering over DL-Lite KBs under S semantics with preferred repairs is:*

- $\Delta_2^P[O(\log n)]$ -complete if we use cardinality (\leq)
- coNP-complete (resp. NP-complete) if we use prioritized set inclusion (\subseteq_P) and $S \in \{AR, IAR\}$ (resp. $S = brave$)
- Δ_2^P -complete if we use prioritized cardinality (\leq_P)
- Δ_2^P -complete if we use weights (\leq_w)

For the last two cases (\leq_P and \leq_w), the complexity drops to $\Delta_2^P[O(\log n)]$ -complete under the assumption that there is a bound on the number of priority classes (resp. maximal weight).

Semantics	Data complexity		Combined complexity	
	CQs	IQs	CQs	IQs
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

Fig. 3: Complexity of inconsistency-tolerant query answering in \mathcal{EL}_\perp . All results are completeness results.

4.2 Beyond DL-Lite

In this subsection, we provide a brief glimpse at the complexity landscape for DLs outside the DL-Lite family. We will consider two representative DLs: \mathcal{EL}_\perp , which extends the lightweight \mathcal{EL} description logic with the ability to express disjointness using \perp , and \mathcal{ALC} , the prototypical expressive description logic. We recall that concepts in \mathcal{EL}_\perp are built from concept names and role names using the constructors \top , \perp , \sqcap , and $\exists R.C$. In \mathcal{ALC} , we have all of the preceding constructors as well as \sqcup , \neg , and $\forall R.C$.

In Figure 3, we display the complexity of querying \mathcal{EL}_\perp KBs under the AR, IAR, and brave semantics (for the sake of comparison, we also recall the complexity results for classical semantics [4, 16, 41, 31, 30]). The key thing to observe is that for \mathcal{EL}_\perp , query answering under the IAR and brave semantics is intractable in data complexity, even for IQs. This contrasts sharply with the tractability of these semantics in the DL-Lite setting. If we consider combined complexity, then the IAR semantics is bit better behaved than AR semantics, which can be explained by the fact that we can compute in advance the intersection of repairs, and then exploit standard querying algorithms. Interestingly, query answering under the brave semantics is NP-complete, irrespectively of whether we consider CQs or IQs, or whether we adopt data complexity or combined complexity.

The results for the AR and IAR semantics were proven in [42]⁵. For the brave semantics, the upper bounds can be proven in the ‘usual’ way: guess a subset $\mathcal{A}' \subseteq \mathcal{A}$ together with a proof that $\langle \mathcal{T}, \mathcal{A}' \rangle \models q(\mathbf{a})$, and verify that \mathcal{A}' is \mathcal{T} -consistent. We provide a proof of the NP-hardness of IQ answering under brave semantics (which we can adapt to show coNP-hardness for the IAR semantics):

Theorem 29. *In \mathcal{EL}_\perp , IQ answering under the IAR (resp. brave) semantics is coNP-hard (resp. NP-hard) in data complexity.*

Proof. The NP lower bound for IQ answering under brave semantics is by reduction from the satisfiability problem for propositional formulas in negation normal form (NNF). Let φ be an NNF formula over the variables v_1, \dots, v_m . We

⁵ The results in [42] are formulated for UCQs rather than CQs, but the same results are obtained for CQs.

Semantics	Data complexity		Combined complexity	
	CQs	IQs	CQs	IQs
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{ALC} . All results are completeness results.

define an \mathcal{EL}_\perp KB $(\mathcal{T}, \mathcal{A})$ as follows:

$$\begin{aligned}
\mathcal{T} &= \{T \sqcap F \sqsubseteq \perp, \quad A_\neg \sqcap \exists r_1.F \sqsubseteq T, \quad A_\wedge \sqcap \exists r_1.T \sqcap \exists r_2.T \sqsubseteq T, \\
&\quad A_\vee \sqcap \exists r_1.T \sqsubseteq T, \quad A_\vee \sqcap \exists r_2.T \sqsubseteq T\} \\
\mathcal{A} &= \{A_\wedge(a_\psi), r_1(a_\psi, a_{\chi_1}), r_2(a_\psi, a_{\chi_2}) \mid \psi = \chi_1 \wedge \chi_2 \text{ is a subformula of } \varphi\} \cup \\
&\quad \{A_\vee(a_\psi), r_1(a_\psi, a_{\chi_1}), r_2(a_\psi, a_{\chi_2}) \mid \psi = \chi_1 \vee \chi_2 \text{ is a subformula of } \varphi\} \cup \\
&\quad \{A_\neg(a_\psi), r_1(a_\psi, a_\chi) \mid \psi = \neg\chi \text{ is a subformula of } \varphi\} \cup \\
&\quad \{T(a_{v_i}), F(a_{v_i}) \mid 1 \leq i \leq m\}
\end{aligned}$$

The structure of the formula is encoded in the ABox, with every subformula ξ of φ (including the variables) represented by a corresponding individual a_ξ . The TBox axiom $T \sqsubseteq \neg F$ forces a choice of truth value for every variable, and the remaining TBox axioms serve to compute the truth value of a formula based upon the truth values of its subformula(s). Using these ideas, one can show that $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\text{brave}} T(a_\varphi)$ if and only if φ is satisfiable.

The coNP lower bound for IAR semantics can be proven by a minor modification of the above reduction. Let $\mathcal{T}' = \mathcal{T} \cup \{T \sqsubseteq \neg B\}$ and $\mathcal{A}' = \mathcal{A} \cup \{B(a_\varphi)\}$. We claim that $\langle \mathcal{T}', \mathcal{A}' \rangle \models_{\text{IAR}} B(a_\varphi)$ if and only if φ is unsatisfiable. Indeed, if φ is satisfiable, then by the preceding paragraph, there must exist a repair of the KB from which $T(a_\varphi)$ is entailed, and hence we can find a repair in which $B(a_\varphi)$ does not hold. Conversely, if φ is unsatisfiable, then we can use the satisfying valuation to construct such a repair. \square

In Figure 4, we give the complexity of querying \mathcal{ALC} KBs under the same three semantics. If we consider data complexity, then we again find that there is no apparent benefit to using the IAR and brave semantics rather than the AR semantics. If we consider combined complexity, then we obtain exactly the same complexity (EXP) as for classical semantics, intuitively because it is possible to enumerate in exponential-time all subsets of the ABox.

For the results concerning AR and IAR semantics, consult [42]⁶. For the brave semantics, the upper bounds and combined complexity lower bound are

⁶ The results in [42] are formulated for UCQs rather than CQs, but the same results are obtained for CQs.

straightforward. We give a proof of the Σ_2^p -hardness in data complexity, which holds already for the extension of \mathcal{EL}_\perp with disjunction.

Theorem 30. *In \mathcal{ELU}_\perp (hence, in \mathcal{ALC}), IQ answering under the brave semantics is Σ_2^p -hard in data complexity.*

Proof. We give a reduction from the validity problem for $\exists\forall$ -QBF, a well-known Σ_2^p -complete problem. Consider a 2QBF instance

$$\Phi = \exists x_1, \dots, x_n \forall y_1, \dots, y_m \tau_1 \vee \dots \vee \tau_p$$

where each $\tau_i = \ell_i^1 \wedge \ell_i^2 \wedge \ell_i^3$ is a propositional 3-term whose variables are drawn from $X = \{x_1, \dots, x_n\}$ and $Y = \{y_1, \dots, y_m\}$. We construct the following KB:

$$\begin{aligned} \mathcal{T} &= \{T \sqsubseteq \neg F, \quad V \sqsubseteq T \sqcup F, \quad \exists R.(S_1 \sqcap S_2 \sqcap S_3) \sqsubseteq A\} \cup \\ &\quad \{\exists P_k.T \sqsubseteq S_k \mid 1 \leq k \leq 3\} \cup \{\exists N_k.F \sqsubseteq S_k \mid 1 \leq k \leq 3\} \\ \mathcal{A}_\Phi &= \{P_k(t_i, z) \mid \ell_i^k = z\} \cup \{N_k(t_i, z) \mid \ell_i^k = \neg z\} \cup \\ &\quad \{T(x_j), F(x_j) \mid 1 \leq j \leq n\} \cup \{V(y_i) \mid 1 \leq i \leq m\} \cup \\ &\quad \{R(a, t_i) \mid 1 \leq i \leq p\} \end{aligned}$$

It can be shown that Φ is valid iff $\langle \mathcal{T}, \mathcal{A}_\Phi \rangle \models_{\text{brave}} A(a)$. We give the main ideas of the argument. Similarly to earlier proofs, the terms and variables are used as ABox individuals, and the roles P_k and N_k are used respectively to link a term t_i to the variables that occur (positively, P , or negatively, N) in the k th literal of the term. For every variable x_j , the inclusion $T \sqsubseteq \neg F$ will force us to choose which of the assertions $T(x_j)$ and $F(x_j)$ to keep. It follows that the repairs will correspond to the different valuations of the x -variables. For each variable y_j , the assertion $V(x_j)$ and inclusions $V \sqsubseteq T \sqcup F$ and $T \sqsubseteq \neg F$ will ensure that in every model of a given repair, we have either $T(y_j)$ or $F(y_j)$. It then remains to check that there is some repair for which every model corresponds to a valuation that satisfies one of the terms. To this end, we use the inclusions $\exists P_k.T \sqsubseteq S_k$ and $\exists N_k.F \sqsubseteq S_k$ to mark with S_k those terms whose k th literal is satisfied by the valuation. The individual a is linked via R to all of the terms, and the inclusion $\exists R.(S_1 \sqcap S_2 \sqcap S_3) \sqsubseteq A$ adds A to a if there is a term marked with S_1 , S_2 , and S_3 , meaning that the three literals in the term are satisfied in the considered valuation. \square

Finally, we should point out that the complexity of inconsistency-tolerant query answering has also been investigated for ontologies formulated using existential rules (aka Datalog^{+/-}), see e.g., [37, 35].

5 Systems for Inconsistency-Tolerant Query Answering

To the best of our knowledge, there are currently three systems for querying DL KBs under inconsistency-tolerant semantics:

- the **QuID** system [43, 34] that performs CQ answering under IAR semantics in an extension of DL-Lite_A with denial and identification constraints,
- the **CQAPri** system [9] that implements CQ answering under the (\subseteq_P -)IAR, (\subseteq_P -)AR, and brave semantics for DL-Lite_R KBs
- the system of Du et al. [21] for ground CQ answering under \leq_w -AR semantics for the highly expressive DL *SHIQ*.

We mention in passing that there are also a few systems for querying inconsistent relational databases under AR semantics, including **ConQuer**[25, 24]) and **Hippo** [18, 19], which implement polynomial-time procedures for tractable subcases, and the **ConsEx** [38] and **EQUIP** [28] systems, which reduce AR query answering to answer set programming and binary integer programming, respectively.

In what follows, we provide some further details on **QuID** and **CQAPri**, as these systems target DLs of the DL-Lite family, our focus in this chapter.

5.1 The QuID System

The **QuID** system implements three different approaches for querying DL-Lite KBs using the IAR semantics: first-order query rewriting, ABox annotation, and ABox cleaning.

Query rewriting A first-order query rewriting procedure for the IAR semantics was proposed in [33] for the logic DL-Lite_A, which extends DL-Lite_R with functionality assertions and attributes. In a later work [34], the rewriting procedure was extended to DL-Lite_{A, id, den}, an expressive member of the DL-Lite family offering identification and denial constraints. The basic idea underlying these rewriting procedures was presented in Example 21.

ABox annotation A second approach to IAR query answering [43] annotates the ABox in an offline phase, using an extra argument to record whether or not an assertion is involved in a minimal inconsistent set. At query time, the query is rewritten using an existing rewriting algorithm for classical semantics, and the rewritten query is modified by adding an extra argument to every atom with the ‘not in any conflict’ value. The modified query can then be evaluated over the annotated ABox.

ABox cleaning The third method [43] computes in an offline phase the assertions appearing in some minimal inconsistent subset and removes these assertions from the ABox. The user query is then handled using an existing query rewriting algorithm for classical semantics.

The three methods have been compared in [43], and the experiments showed that the ABox cleaning and annotation approaches often significantly outperform the method based on query rewriting. This is because the IAR-rewritings, which require rather complex conditions to block ‘unsafe’ assertions, are more difficult to evaluate than rewritings for classical semantics. However, as noted in [43], the annotation and cleaning approaches both involve making modifications to the ABox, which may not be feasible in applications where data is read-only.

5.2 The CQAPri System

The CQAPri system, which accepts DL-Lite \mathcal{R} KBs, uses the IAR, brave, and AR semantics in combination to identify three types of query answer:

- **Possible:** $\mathcal{K} \models_{\text{brave}} q(\mathbf{a})$ and $\mathcal{K} \not\models_{\text{AR}} q(\mathbf{a})$
- **Likely:** $\mathcal{K} \models_{\text{AR}} q(\mathbf{a})$ and $\mathcal{K} \not\models_{\text{IAR}} q(\mathbf{a})$
- **(Almost) sure:** $\mathcal{K} \models_{\text{IAR}} q(\mathbf{a})$

We briefly describe the steps needed to obtain such a classification.

In an offline phase, CQAPri computes and stores the set $\text{conflicts}(\mathcal{K})$ of minimal \mathcal{T} -inconsistent subsets (called *conflicts* in [9]) of the KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$. As \mathcal{K} is a DL-Lite \mathcal{R} KB, every set in $\text{conflicts}(\mathcal{K})$ contains at most two assertions.

When a query q arrives, CQAPri uses an off-the-shelf rewriting system to compute a UCQ-rewriting $q_1 \vee \dots \vee q_n$ of q under classical semantics. The rewriting is evaluated over the ABox to obtain the *candidate answers* of q . For each candidate answer \mathbf{a} , we store the *images* of $q(\mathbf{a})$ in \mathcal{A} , i.e., the sets of assertions corresponding to the homomorphic image of some $q_i(\mathbf{a})$. Candidate answers define a superset of the answers holding under the brave semantics.

Among the candidate answers, CQAPri identifies those holding under IAR semantics by checking whether there is some image of the candidate answer whose assertions are not involved in any conflict. It also identifies the candidate answers which are not brave-answers by discarding images which are \mathcal{T} -inconsistent: an answer that has only such images does not hold under brave semantics.

For the tuples that have been shown to be answers under brave semantics but not under IAR semantics, it remains to decide whether the tuple is an AR-answer. For every such tuple \mathbf{a} , a propositional formula $\varphi_{q(\mathbf{a})}?$ is constructed (in polynomial time in $|\mathcal{A}|$) and passed on to a SAT solver for evaluation. As shown in Theorem 31, the formula $\varphi_{q(\mathbf{a})}?$ is satisfiable iff $\mathcal{K} \not\models_{\text{AR}} q(\mathbf{a})$.

The formula $\varphi_{q(\mathbf{a})}?$ is defined as the conjunction of the formulas $\varphi_{\neg q(\mathbf{a})}$ and φ_{cons} displayed in Figure 5. The encoding makes use of the sets $\text{causes}(q(\mathbf{a}), \mathcal{K})$ and $\text{confl}(\mathcal{C}, \mathcal{K})$ defined as follows:

$$\begin{aligned} \text{causes}(q(\mathbf{a}), \mathcal{K}) &= \{ \mathcal{C} \subseteq \mathcal{A} \mid \mathcal{C} \text{ is minimal } \mathcal{T}\text{-consistent support for } q(\mathbf{a}) \} \\ \text{confl}(\mathcal{C}, \mathcal{K}) &= \{ \beta \mid \exists \alpha \in \mathcal{C}, \{ \alpha, \beta \} \in \text{conflicts}(\mathcal{K}) \} \end{aligned}$$

and the notation $\text{vars}(\varphi_{\neg q})$, which denotes the set of variables appearing in $\varphi_{\neg q}$.

Both formulas use variables of the form x_α , where α is an assertion from \mathcal{A} , so every valuation of the variables corresponds to a subset of \mathcal{A} . The formula $\varphi_{\neg q(\mathbf{a})}$ states that this subset must include, for every support \mathcal{C} for $q(\mathbf{a})$, some assertion that contradicts \mathcal{C} . The second formula φ_{cons} formula ensures that the set of selected assertions is consistent with the TBox. Taken together, these formulas guarantee the existence of a consistent subset of the ABox that can be extended to a repair of the KB from which $q(\mathbf{a})$ cannot be derived.

Theorem 31 ([9]). *Let $\varphi_{\neg q(\mathbf{a})}$ and φ_{cons} be defined as in Figure 5. Then $\mathcal{K} \not\models_{\text{AR}} q(\mathbf{a})$ if and only if $\varphi_{\neg q(\mathbf{a})} \wedge \varphi_{\text{cons}}$ is satisfiable.*

$$\begin{aligned}
\varphi_{\neg q} &= \bigwedge_{\mathcal{C} \in \text{causes}(q, \mathcal{K})} \bigvee_{\beta \in \text{confl}(\mathcal{C}, \mathcal{K})} x_{\beta} \\
\varphi_{\text{cons}} &= \bigwedge_{x_{\alpha}, x_{\beta} \in \text{vars}(\varphi_{\neg q}), \beta \in \text{confl}(\{\alpha\}, \mathcal{K})} \neg x_{\alpha} \vee \neg x_{\beta}
\end{aligned}$$

Fig. 5: SAT encoding for AR query answering.

The SAT encoding in Figure 5 can be adapted to handle the \subseteq_P -IAR and \subseteq_P -AR semantics by adding a third set of clauses that ensure that the set of selected assertions is maximal w.r.t. \subseteq_P . See [9] for details.

The experiments conducted in [9] show that the IAR semantics generally constitutes a very good approximation of the AR semantics and that query answering scales well on realistic cases, when a few percents of the ABox assertions are involved in a conflict. Indeed, in such cases, a large portion of the brave-answers hold under the IAR semantics, so CQAPri does not need to call a SAT solver to decide whether they are entailed under AR semantics.

6 Summary and Outlook

In this chapter, we have surveyed a number of different inconsistency-tolerant semantics that have been proposed to obtain meaningful query results from inconsistent DL knowledge bases. In order to provide a basis for selecting the appropriate semantic(s) for a given application, we compared the different semantics in terms of the consistency properties they satisfy, the set of query answers they return (using the notion of under- and over-approximations), and the computational complexity of the associated query answering task, focusing mainly on DL-Lite KBs. We have also described how the IAR, brave, and AR semantics can be implemented in a query answering system. Preliminary experiments reported in [9] suggest that for DL-Lite knowledge bases, it is possible to efficiently compute the answers holding under the appealing but intractable AR semantics by combining tractable approximations in the form of the IAR and brave semantics with the power of modern SAT solvers.

With the emergence of the inconsistency-tolerant query answering systems for DL-Lite comes a set of new challenges related to improving both the efficiency and the usability of such systems. Regarding efficiency, both of the systems that have been implemented for querying inconsistent DL-Lite KBs employ UCQ-rewriting algorithms. It would be interesting to see whether other approaches to DL-Lite query answering, like the combined approach [29] or rewritings into positive existential or non-recursive Datalog queries, can be suitably adapted for use in inconsistency-tolerant querying systems. As for usability, one problem that has begun to be explored is how to explain query results to users, that is, how to justify why a given tuple appears as an answer under the considered

inconsistency-tolerant semantics and why some other tuples do not appear in the results. Some first results on explaining query (non)answers under the AR, IAR, and brave semantics have been presented in [10]. Interaction with the user is also important if we want to improve the quality of the ABox. A recent work [11] proposes to allow users to 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.

Another important challenge for future work is to devise practical methods for inconsistency-tolerant OMQA that work for DLs outside the DL-Lite family. Indeed, as shown in Section 4, the complexity landscape changes rather dramatically when we leave the DL-Lite world. In particular, we have seen that even for \mathcal{EL}_\perp , a relatively simple DL that has good computational properties, IQ answering under the IAR and brave semantics is intractable in data complexity. Thus, a first research direction is to invent new inconsistency-tolerant semantics that permit tractable query answering in DLs beyond DL-Lite. A second possibility is to design algorithms for existing inconsistency-tolerant semantics that work well in practice, even if they are not guaranteed to run in polynomial time.

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 (1999)
2. Arenas, M., Bertossi, L.E., Kifer, M.: Applications of annotated predicate calculus to querying inconsistent databases. In: Proceedings of the 1st International Conference on Computational Logic. pp. 926–941 (2000)
3. Artale, A., Calvanese, D., Kontchakov, R., Zakharyashev, M.: The DL-Lite family and relations. *Journal of Artificial Intelligence Research (JAIR)* 36, 1–69 (2009)
4. Baader, F., Brandt, S., Lutz, C.: Pushing the \mathcal{EL} envelope. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI). pp. 364–369 (2005)
5. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F. (eds.): *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press (2003)
6. Baget, J., Benferhat, S., Bouraoui, Z., Croitoru, M., Mugnier, M., Papini, O., Rocher, S., Tabia, K.: A general modifier-based framework for inconsistency-tolerant query answering. In: Proceedings of the 15th International Conference on the Principles of Knowledge Representation and Reasoning (KR). pp. 513–516 (2016)
7. Bertossi, L.E.: *Database Repairing and Consistent Query Answering*. Synthesis Lectures on Data Management, Morgan & Claypool Publishers (2011)
8. Bienvenu, M.: On the complexity of consistent query answering in the presence of simple ontologies. In: Proceedings of the 26th AAAI Conference on Artificial Intelligence (2012)
9. 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 (2014)

10. 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 (2016)
11. Bienvenu, M., Bourgaux, C., Goasdoué, F.: Query-driven repairing of inconsistent DL-Lite knowledge bases. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI) (2016)
12. Bienvenu, M., Ortiz, M.: Ontology-mediated query answering with data-tractable description logics. In: Lecture Notes of the 11th International Reasoning Web Summer School. LNCS, vol. 9203, pp. 218–307. Springer (2015)
13. 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)
14. Bourgaux, C.: Inconsistency handling in ontology-mediated query answering manuscript of PhD thesis, in preparation (2016)
15. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *Journal of Automated Reasoning (JAR)* 39(3), 385–429 (2007)
16. Calvanese, D., Giacomo, G.D., Lembo, D., Lenzerini, M., Rosati, R.: Data complexity of query answering in description logics. In: Proceedings of the 10th International Conference on the Principles of Knowledge Representation and Reasoning (KR). pp. 260–270 (2006)
17. Chomicki, J.: Consistent query answering: Five easy pieces. In: Proceedings of the 10th International Conference on Database Theory (ICDT). pp. 1–17 (2007)
18. Chomicki, J., Marcinkowski, J., Staworko, S.: Computing consistent query answers using conflict hypergraphs. In: Proceedings of the International Conference on Information and Knowledge Management (CIKM). pp. 417–426 (2004)
19. Chomicki, J., Marcinkowski, J., Staworko, S.: Hippo: A system for computing consistent answers to a class of SQL queries. In: Proceedings of the 9th International Conference on Extending Database Technology (EDBT). pp. 841–844 (2004)
20. Donini, F.M., Lenzerini, M., Nardi, D., Schaerf, A.: Deduction in concept languages: From subsumption to instance checking. *Journal of Logic and Computation (JLC)* 4(4), 423–452 (1994)
21. Du, J., Qi, G., Shen, Y.D.: Weight-based consistent query answering over inconsistent *SHIQ* knowledge bases. *Knowledge and Information Systems* 34(2), 335–371 (2013)
22. Eiter, T., Gottlob, G.: The complexity of logic-based abduction. *Journal of the ACM* 42(1), 3–42 (1995)
23. Eiter, T., Gottlob, G.: The complexity class Θ_2^P : Recent results and applications in AI and modal logic. In: Proceedings of the 11th International Symposium on Fundamentals of Computation Theory (FCT). pp. 1–18 (1997)
24. Fuxman, A., Fazli, E., Miller, R.J.: Conquer: Efficient management of inconsistent databases. In: Proceedings of the ACM International Conference on Management of Data (SIGMOD). pp. 155–166 (2005)
25. Fuxman, A., Miller, R.J.: First-order query rewriting for inconsistent databases. In: Proceedings of the 10th International Conference on Database Theory (ICDT). pp. 337–351 (2005)
26. Gottlob, G.: NP Trees and Carnap’s modal logic. *Journal of the ACM* 42(2), 421–457 (1995)
27. Immerman, N.: Nondeterministic space is closed under complementation. *SIAM Journal of Computing* 17(5), 935–938 (1988)

28. Kolaitis, P.G., Pema, E., Tan, W.C.: Efficient querying of inconsistent databases with binary integer programming. *Proceedings of the VLDB Endowment (PVLDB)* 6(6), 397–408 (2013)
29. Kontchakov, R., Lutz, C., Toman, D., Wolter, F., Zakharyashev, M.: The combined approach to query answering in DL-Lite. In: *Proceedings of the 12th International Conference on the Principles of Knowledge Representation and Reasoning (KR)* (2010)
30. Krisnadhi, A., Lutz, C.: Data complexity in the \mathcal{EL} family of DLs. In: *Proceedings of the 20th International Workshop on Description Logics (DL)* (2007)
31. Krötzsch, M., Rudolph, S.: Conjunctive queries for \mathcal{EL} with composition of roles. In: *Proceedings of the 20th International Workshop on Description Logics (DL)* (2007)
32. 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)
33. 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)
34. Lembo, D., Lenzerini, M., Rosati, R., Ruzzi, M., Savo, D.F.: Inconsistency-tolerant query answering in ontology-based data access. *Journal of Web Semantics (JWS)* 33, 3–29 (2015)
35. Lukasiewicz, T., Martinez, M.V., Pieris, A., Simari, G.I.: From classical to consistent query answering under existential rules. In: *Proceedings of the 29th AAAI Conference on Artificial Intelligence*. pp. 1546–1552 (2015)
36. Lukasiewicz, T., Martinez, M.V., Simari, G.I.: Inconsistency handling in Datalog+/- ontologies. In: *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI)* (2012)
37. Lukasiewicz, T., Martinez, M.V., Simari, G.I.: Complexity of inconsistency-tolerant query answering in Datalog+/- . In: *Proceedings of On the Move to Meaningful Internet Systems Conference (OTM)*. pp. 488–500 (2013)
38. Marileo, M.C., Bertossi, L.E.: The consistency extractor system: Answer set programs for consistent query answering in databases. *Data and Knowledge Engineering* 69(6), 545–572 (2010)
39. Motik, B., Cuenca Grau, B., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C.: OWL 2 Web Ontology Language Profiles. W3C Recommendation (2012), available at <http://www.w3.org/TR/owl2-profiles/>
40. W3C OWL Working Group: OWL 2 Web Ontology Language: Document Overview. W3C Recommendation (2009), available at <http://www.w3.org/TR/2012/REC-owl2-overview-20121211/>
41. Rosati, R.: On conjunctive query answering in \mathcal{EL} . In: *Proceedings of the Twentieth International Workshop on Description Logics (DL)* (2007)
42. Rosati, R.: On the complexity of dealing with inconsistency in description logic ontologies. In: *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)* (2011)
43. Rosati, R., Ruzzi, M., Graziosi, M., Masotti, G.: Evaluation of techniques for inconsistency handling in OWL 2 QL ontologies. In: *Proceedings of the 12th International Semantic Web Conference (ISWC)* (2012)
44. Szelepcsényi, R.: The method of forcing for nondeterministic automata. *Bulletin of the EATCS* 33, 96–99 (1987)
45. Wagner, K.W.: More complicated questions about maxima and minima, and some closures of NP. *Theoretical Computer Science* 51, 53–80 (1987)