Inconsistency-Tolerant Semantics Based on (Preferred) Repairs

Camille Bourgaux
DI ENS, ENS, CNRS, PSL University & Inria, Paris, France

Abstract

Real-world datasets are plagued by data quality issues which may render the data inconsistent w.r.t. a set of constraints, be they given by database integrity constraints or ontologies. A prominent way to handle such inconsistent data is to use inconsistency-tolerant semantics to obtain meaningful answers to queries. Most of these semantics are based on some notion of repairs, which represent ways of restoring the data consistency. The most basic kind of repairs is that of subset repairs, which are maximal consistent subsets of the dataset. However, in many scenarios, one can define preferred repairs based on some preference information. These lecture notes present inconsistency-tolerant semantics, focusing on the repair-based ones, then review different kinds of preferred repairs that have been considered in the literature. We present in particular the relationships between different kinds of preferred repairs and other notions related to inconsistency handling, the computational complexity of reasoning with (preferred) repairs, and some implementations.

2012 ACM Subject Classification Theory of computation → Automated reasoning; Computing methodologies → Description logics; Computing methodologies → Nonmonotonic, default reasoning and belief revision; Computing methodologies → Reasoning about belief and knowledge; Information systems → Data cleaning

Keywords and phrases Knowledge bases, databases, inconsistency handling, repairs, preferences

Digital Object Identifier 10.4230/OASIcs.RW.2024/2025.5

Category Invited Paper

Related Version *Full Version*: `https://inria.hal.science/hal-05291421`

Funding This lecture was supported by the ANR AI Chair INTENDED (ANR-19-CHIA-0014).

Acknowledgements I am grateful to the RW 2025 organizers for the invitation to give this lecture. Several sections of these lecture notes are based on some work on priority-based repairs that I co-authored with Meghyn Bienvenu.

1 Introduction

Many relational datasets are associated with either some integrity constraints, which aim at preventing errors by expressing properties expected from the stored data [1], or with an ontology, which expresses knowledge on the domain of interest and allows a user to formulate queries in familiar terms and obtain more complete answers through logical reasoning [124, 136, 40]. Integrity constraints and ontologies both consist of logical formulas that can be written as first-order logic sentences. In particular, ontologies are typically expressed in some description logics [14] or extensions of Datalog [20, 56]. The main difference between the database setting, which considers a dataset equipped with a set of integrity constraints, and the knowledge base setting, where a dataset is enriched with an ontology, is that the former uses the logical formulas to check whether the data complies with some rules it is expected to follow while the latter uses them to infer implicit information from the explicitly stored data. However, the question of how to handle the case where the dataset is inconsistent w.r.t. the background knowledge is crucial in both settings. Indeed, it is widely acknowledged that real-world datasets are plagued by data quality issues, and it is not always possible to restore consistency since it may be difficult to identify the erroneous parts of the data, and removing all potentially erroneous facts may lead to unacceptable loss of information.

A prominent way to tackle this issue is to use some inconsistency-tolerant semantics. Such a semantics was first proposed in the database area under the name of consistent query answering, with the goal of obtaining query answers that would be consistent w.r.t. the database constraints even though the database itself was not [8, 25]. In a nutshell, a query answer is consistent if it can be obtained from all repairs of the database, where a repair is a dataset consistent w.r.t. the constraints which minimally differs from the original dataset. As the inconsistency issue is even more problematic for knowledge bases, since everything is entailed by an inconsistent knowledge base under the classical semantics, consistent query answering was adapted to knowledge bases under the name of AR semantics (AR standing for ABox Repair, as a dataset is called an ABox in description logic parlance) [98, 99]. While this semantics remains the most studied and well-established, many other inconsistency-tolerant semantics have been proposed, e.g., to address the intractability of consistent query answering via approximations, or to get a finer classification of the reliability of query answers. A great deal of research effort has been put into exploring the problem of querying inconsistent databases and knowledge bases under these semantics, in particular on the theoretical side, with extensive studies of the computational complexity of this problem in various settings, but also with more practically-oriented research aiming at developing and implementing efficient algorithms for this problem (see surveys [25, 26, 29, 28]).

The notion of repair is at the basis of most of inconsistency-tolerant semantics. However, in many scenarios, one can define preferred repairs based on some preference information, and use these preferred repairs instead of standard ones when querying inconsistent databases or knowledge bases to obtain more accurate answers. For example, optimal repairs based on weights associated to the dataset facts, on a priority relation between facts, or on preference rules have been considered [102, 129, 73, 33, 30, 100, 122, 50, 107, 32].

The goal of these lecture notes is to provide an overview of inconsistency handling based on database or knowledge base (preferred) repairs. After some preliminaries, we introduce the basic notion of a repair in Section 3 and present some characterizations of repairs, based on the concept of conflict hypergraph and argumentation frameworks. In Section 4, we briefly present existing inconsistency-tolerant semantics, focusing on the repair-based ones. We then review the different kinds of preferred repairs that have been considered in the literature in Section 5, focusing on their properties and relationships, and presenting known links to other frameworks for inconsistency handling. We also expand the focus to consider related work on other ways of defining optimal consistent datasets that are not repairs per se. In Section 6, we present some results on the computational complexity of deciding whether a dataset is a repair and of reasoning under repair-based semantics, then on the impact of using preferred repairs instead of standard ones, focusing in particular on data complexity. In Section 7, we give a concise overview of systems that have been implemented for querying inconsistent databases and knowledge bases under (preferred) repair-based semantics and illustrate some prominent practical approaches. We conclude with a discussion of recent related work and research directions in Section 8.

Contents

2 Preliminaries

In this section, we introduce the database and knowledge base settings we will consider throughout this chapter and recall some relevant results on their computational complexity. We assume a basic knowledge of first-order logic (FO) syntax and semantics.

2.1 Databases, knowledge bases, queries

Let $\mathbf{P}$, $\mathbf{C}$, and $\mathbf{V}$ be three disjoint countable sets of *predicates*, *constants* and *variables* respectively. Each predicate is associated with an arity, i.e., a non-negative integer. A *term* is a constant or a variable and a *(relational) atom* has form $P(t_1, \dots, t_n)$ where P is a predicate of arity n and $t_1, \dots, t_n$ are terms. A *fact* is a ground atom, i.e., an atom that does not contain any variable. A *dataset* is a finite set of facts. We consider pairs $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ where $\mathcal{D}$ is a dataset and $\mathcal{T}$ is a logical *theory*, i.e., a finite set of logical formulas, which is interpreted either as a set of integrity constraints, or as an ontology. In the former case, we call $\mathcal{K}$ a *database*, and in the latter, a *knowledge base* (KB). As ontology languages, we consider description logics and Datalog$^\pm$ (a.k.a. existential rules with negative constraints). We use *interpretations* to define the semantics of databases and KBs. An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a domain $\Delta^{\mathcal{I}}$ and an interpretation function $\cdot^{\mathcal{I}}$, which maps each constant $c \in \mathbf{C}$ to some domain element $c^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ and each n-ary predicate $P \in \mathbf{P}$ to a set of n-tuples $P^{\mathcal{I}} \subseteq (\Delta^{\mathcal{I}})^n$. Given an interpretation $\mathcal{I}$ and an FO sentence ϕ, we write $\mathcal{I} \models \phi$ to denote that $\mathcal{I}$ satisfies ϕ under the standard FO semantics. Given a dataset $\mathcal{D}$, we define the *interpretation corresponding to* $\mathcal{D}$ as follows: $\mathcal{I}_{\mathcal{D}} = (\mathbf{C}, \cdot^{\mathcal{I}_{\mathcal{D}}})$ and $\cdot^{\mathcal{I}_{\mathcal{D}}}$ interprets each constant by itself and each predicate P by the set of tuples $\{(c_1, \dots, c_n) \mid P(c_1, \dots, c_n) \in \mathcal{D}\}$.

Database integrity constraints. An *integrity constraint* (IC) is simply an FO sentence. A dataset $\mathcal{D}$ *satisfies* an IC ϕ iff $\mathcal{I}_{\mathcal{D}} \models \phi$. If $\mathcal{D}$ does not satisfy ϕ, we also say that $\mathcal{D}$ *violates* ϕ. A database $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ is *consistent* if $\mathcal{D}$ satisfies all ICs in $\mathcal{T}$, inconsistent otherwise (written $\mathcal{K} \models \bot$). The dataset $\mathcal{D}$ is $\mathcal{T}$*-(in)consistent* if $\langle \mathcal{D}, \mathcal{T} \rangle$ is (in)consistent. Most studied IC languages can be written as disjunctive rules of the form:

$$\forall \vec{x}(\beta[\vec{x}] \wedge \epsilon[\vec{x}] \rightarrow \bigvee_{i=1}^{k} \exists \vec{y_i} \eta_i[\vec{x}, \vec{y_i}]) \tag{1}$$

where $\beta[\vec{x}]$ and $\eta_i[\vec{x}, \vec{y_i}]$ (for $1 \leq i \leq k$) are conjunctions of relational atoms whose terms are in $\vec{x}$ and $\vec{x} \cup \vec{y_i}$, respectively, and $\epsilon[\vec{x}]$ is a conjunction of inequality atoms whose terms are in $\vec{x}$. Moreover, every variable in $\vec{x}$ is required to occur in $\beta[\vec{x}]$ to ensure safety. We often omit the universal quantifier and write $\beta[\vec{x}] \wedge \epsilon[\vec{x}] \rightarrow \bigvee_{i=1}^{k} \exists \vec{y_i} \eta_i[\vec{x}, \vec{y_i}]$. The left-hand side $\beta[\vec{x}] \wedge \epsilon[\vec{x}]$ is called the *body* of the rule and the right-hand side $\bigvee_{i=1}^{k} \exists \vec{y_i} \eta_i[\vec{x}, \vec{y_i}]$ its *head*. Figure 1 (left) shows a hierarchy of IC languages [11] (in this picture, FO stands for domain independent FO sentences).

- A *universal constraint* (UC) is a rule of form (1) without existential quantifiers. Any UC can be written as a set of rules of form $\beta[\vec{x}] \wedge \nu[\vec{x}] \wedge \epsilon[\vec{x}] \rightarrow \bot$, where $\nu[\vec{x}]$ is a conjunction of *negated* relational atoms (e.g., $P(x, y) \rightarrow Q(x) \vee (R(x, y) \wedge S(y))$ can be written as two rules $P(x, y) \wedge \neg Q(x) \wedge \neg R(x, y) \rightarrow \bot$ and $P(x, y) \wedge \neg Q(x) \wedge \neg S(y) \rightarrow \bot$).
- A *denial constraint* (DC) is a rule of form (1) with an empty head: $\beta[\vec{x}] \wedge \epsilon[\vec{x}] \rightarrow \bot$.
- An *equality-generating dependency* (EGD) is a DC where $\epsilon[\vec{x}]$ is a single inequality and is often written as $\beta[\vec{x}] \rightarrow x_i = x_j$.
- Given a predicate P of arity n, a *functional dependency* (FD) over P is of the form $P(x_1, \dots, x_n) \wedge P(y_1, \dots, y_n) \wedge \bigwedge_{i \in I} x_i = y_i \rightarrow \bigwedge_{j \in J} x_j = y_j$ for two disjoint sets of indexes $I, J \subseteq \{1, \dots, n\}$. Any FD can be written as a set of EGDs (e.g., $P(x_1, x_2, x_3) \wedge P(y_1, y_2, y_3) \wedge x_1 = y_1 \rightarrow x_2 = y_2 \wedge x_3 = y_3$ can be written as two EGDs $P(x_1, x_2, x_3) \wedge P(x_1, y_2, y_3) \rightarrow x_i = y_i$ for $i \in \{2, 3\}$).
- A *key constraint* is an FD such that $I \cup J = \{1, \dots, n\}$.
- A *disjunctive tuple-generating dependency* ($\vee$-TGD) is a rule of form (1) with an empty $\epsilon[\vec{x}]$, i.e., no inequalities.

- A *tuple-generating dependency* (TGD) is a $\vee$-TGD without disjunction (i.e., where $k = 1$).
- A *local-as-view TGD* (lav TGD) is a TGD where the body $\beta[\vec{x}]$ is a single atom.
- An *inclusion dependency* (ID) is a lav TGD where the head $\eta[\vec{x}, \vec{y}]$ is also a single atom.
- A *full* ($\vee$-)TGD is a ($\vee$-)TGD that does not contain any existential quantifier.

DCs (hence all ICs encompassed by DCs) are *anti-monotone* constraints: if a dataset $\mathcal{D}$ satisfies a DC ϕ, then every $\mathcal{D}' \subseteq \mathcal{D}$ also satisfies ϕ.

▶ **Example 1.** Consider the following university database $\langle \mathcal{D}_{\mathsf{DB}}, \mathcal{T}_{\mathsf{DB}} \rangle$. The predicate $\mathsf{Student}$ is used to register students with their identifiers, first names and last names, $\mathsf{TakeCourse}$ associates students and courses they are enrolled in, and $\mathsf{GradCourse}$ and $\mathsf{UndergradCourse}$ list the graduate and undergraduate courses, respectively. There are five ICs: (IC_1) is a key constraint that states that the identifier determines the student name; (IC_2) is a TGD that states that every student must be enrolled in some course; (IC_3) is a full $\vee$-TGD that states that every course must be registered as a graduate or undergraduate course; (IC_4) is a DC that ensures that no student identifier occurs in the second position of $\mathsf{TakeCourse}$, which is intended for course identifiers; (IC_5) is a DC that states that graduate and undergraduate courses are disjoint. The database $\langle \mathcal{D}_{\mathsf{DB}}, \mathcal{T}_{\mathsf{DB}} \rangle$ is inconsistent: it violates (IC_1) because of the two first facts, (IC_3) because 456 is not registered as a graduate nor an undergraduate course while it occurs in second position in $\mathsf{TakeCourse}(123, 456)$, ($\mathsf{IC}_4$) because of $\mathsf{Student}(456, \mathit{Bea}, \mathit{Blue})$ and $\mathsf{TakeCourse}(123, 456)$, and ($\mathsf{IC}_5$) because of the last two facts. Note that if we remove $\mathsf{TakeCourse}(123, 456)$ from $\mathcal{D}_{\mathsf{DB}}$, ($\mathsf{IC}_3$) and ($\mathsf{IC}_4$) will be satisfied, but (IC_2) will become violated since $\mathsf{TakeCourse}(123, 456)$ is the only fact that associates some course to 123.

$$
\begin{aligned}
\mathcal{T}_{\mathsf{DB}} = \{ & \mathsf{Student}(x, y, z) \wedge \mathsf{Student}(x', y', z') \wedge x = x' \rightarrow y = y' \wedge z = z', && (\mathsf{IC}_1)\\
& \mathsf{Student}(x, y, z) \rightarrow \exists v \mathsf{TakeCourse}(x, v), && (\mathsf{IC}_2)\\
& \mathsf{TakeCourse}(x, y) \rightarrow \mathsf{GradCourse}(y) \vee \mathsf{UndergradCourse}(y), && (\mathsf{IC}_3)\\
& \mathsf{Student}(x, y, z) \wedge \mathsf{TakeCourse}(v, x) \rightarrow \bot, && (\mathsf{IC}_4)\\
& \mathsf{GradCourse}(x) \wedge \mathsf{UndergradCourse}(x) \rightarrow \bot \} && (\mathsf{IC}_5)\\
\mathcal{D}_{\mathsf{DB}} = \{ & \mathsf{Student}(123, \mathit{Ana}, \mathit{Azure}), \mathsf{Student}(123, \mathit{Ann}, \mathit{Azure}), \mathsf{Student}(456, \mathit{Bea}, \mathit{Blue}),\\
& \mathsf{TakeCourse}(123, 456), \mathsf{TakeCourse}(456, \mathit{CS1}),\\
& \mathsf{GradCourse}(\mathit{CS1}), \mathsf{UndergradCourse}(\mathit{CS1}) \}
\end{aligned}
$$

Datalog$^{\pm}$. A Datalog$^{\pm}$ ontology, or Datalog$^{\pm}$ *program*, is a finite set of rules of two kinds:

- TGDs (also called *existential rules*), i.e., rules of the form $\beta[\vec{x}] \rightarrow \exists \vec{y} \eta[\vec{x}, \vec{y}]$, and
- *negative constraints* (NCs), which are rules of the form $\beta[\vec{x}] \rightarrow \bot$ where $\beta[\vec{x}]$ is a conjunction of relational atoms with variables $\vec{x}$ (and may contain constants) [104].

However, the semantics of Datalog$^{\pm}$ is different from that of database integrity constraints. Indeed, while databases are interpreted under the *closed world assumption*, KBs are interpreted under the *open world assumption*, so that facts that do not belong to the dataset are not deemed to be false. Hence, instead of evaluating rules on the interpretation corresponding to the dataset as in the database setting, one reasons over the set of all models of the KB. A *model* of a Datalog$^{\pm}$ KB $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ is an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ such that

- $\Delta^{\mathcal{I}}$ contains all constants from $\mathcal{D}$ and $c^{\mathcal{I}} = c$ for each constant c that occurs in $\mathcal{D}$ (*standard name assumption*),
- $\mathcal{I} \models P(c_1, \ldots, c_n)$, i.e., $(c_1, \ldots, c_n) \in P^{\mathcal{I}}$, for every $P(c_1, \ldots, c_n) \in \mathcal{D}$, and
- $\mathcal{I} \models \phi$ for every $\phi \in \mathcal{T}$.

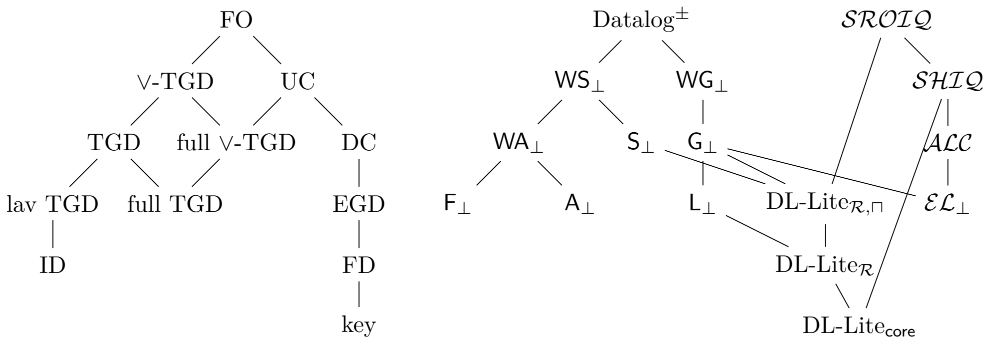


Figure 1 Hierarchies of database integrity contraint languages (left) [11, Fig. 1] and of some ontology languages (right). There is a downward path from $\mathcal{L}_1$ to $\mathcal{L}_2$ if any set of ICs (resp. any ontology) in $\mathcal{L}_2$ can be rewritten into an equivalent set of ICs (resp. ontology) in $\mathcal{L}_1$.

A KB is *consistent*, or satisfiable, if it has a model, inconsistent otherwise ($\mathcal{K} \models \bot$). Again, $\mathcal{D}$ is $\mathcal{T}$*-(in)consistent* if $\langle \mathcal{D}, \mathcal{T} \rangle$ is (in)consistent. A KB *entails* an FO sentence ϕ, written $\mathcal{K} \models \phi$, if $\mathcal{I} \models \phi$ for every model $\mathcal{I}$ of $\mathcal{K}$. To illustrate the difference between TGDs used as ICs or as ontology language, consider $\mathcal{D} = \{P(c)\}$ and $\mathcal{T} = \{P(x) \rightarrow Q(x)\}$. The database $\langle \mathcal{D}, \mathcal{T} \rangle$ is inconsistent while the KB $\langle \mathcal{D}, \mathcal{T} \rangle$ is not (and entails $Q(c)$). Since query answering in Datalog$^\pm$ is undecidable (this is already the case for existential rules, without NCs), a wealth of decidable fragments have been designed. Some of the most studied are as follows [104].

- Recall that a TGD is called *full* if it does not have any existential quantifier. We denote by F the ontology language of full TGDs, also known as *Datalog*.
- A set of TGDs is *acyclic* if its predicate graph, defined as the directed graph with vertices $\mathbf{P}$ and an edge from P to Q iff there is a rule whose body contains P and whose head contains Q, is acyclic. We denote by A the language of acyclic sets of TGDs.
- A TGD is *guarded* if its body has an atom (called a guard) that contains all variables occurring in the body. We denote by G the language of guarded TGDs.
- A TGD is *linear* if its body contains a single atom. We denote by L the language of linear TGDs.
- A set of TGDs is *sticky* if it respects a syntactic condition which ensures that terms that unify with variables that appear more than once in the body of some rule are always propagated by the rules. We denote by S the language of sets of sticky TGDs.
- A, G and S come with their respective "weak" versions which restrict the acyclicity, guardedness and stickiness syntactic conditions to take into account only potentially "harmful" variables: weakly acyclic (WA), weakly guarded (WG) and weakly sticky (WS).

For every TGD language X, we denote by $\mathsf{X}_\bot$ its extension with NCs. Figure 1 (right) shows a hierarchy of ontology languages, including these Datalog$^\pm$ fragments.

▶ **Example 2.** Consider the KB $\langle \mathcal{D}_{\mathsf{KB}}, \mathcal{T}_{\mathsf{KB}} \rangle$ obtained from the database of Example 1 by dropping (IC_1) and (IC_3) (since key constraints and disjunctive TGDs are not supported by Datalog$^\pm$), which belongs to all the Datalog$^\pm$ fragments of Figure 1 except $\mathsf{F}_\bot$.

$$\begin{aligned}
\mathcal{T}_{\mathsf{KB}} = \{&\mathsf{Student}(x,y,z) \rightarrow \exists v \mathsf{TakeCourse}(x,v), && (\mathsf{TGD})\\
&\mathsf{Student}(x,y,z) \wedge \mathsf{TakeCourse}(v,x) \rightarrow \bot, && (\mathsf{NC}_1)\\
&\mathsf{GradCourse}(x) \wedge \mathsf{UndergradCourse}(x) \rightarrow \bot\} && (\mathsf{NC}_2)\\
\mathcal{D}_{\mathsf{KB}} = \{&\mathsf{Student}(123, \mathit{Ana}, \mathit{Azure}), \mathsf{Student}(123, \mathit{Ann}, \mathit{Azure}), \mathsf{Student}(456, \mathit{Bea}, \mathit{Blue}),\\
&\mathsf{TakeCourse}(123, 456), \mathsf{TakeCourse}(456, \mathit{CS1}),\\
&\mathsf{GradCourse}(\mathit{CS1}), \mathsf{UndergradCourse}(\mathit{CS1})\}
\end{aligned}$$

$\langle \mathcal{D}_{\mathsf{KB}}, \mathcal{T}_{\mathsf{KB}} \rangle$ is inconsistent since for every interpretation $\mathcal{I}$, if $\mathcal{I} \models \mathsf{Student}(456, \mathit{Bea}, \mathit{Blue})$ and $\mathcal{I} \models \mathsf{TakeCourse}(123, 456)$, then $\mathcal{I}$ does not satisfy (NC_1), and similarly, every interpretation that satisfies the two last facts violates (NC_2). However, in contrast with the database setting, removing the facts $\mathsf{TakeCourse}(123, 456)$ and, e.g., $\mathsf{GradCourse}(\mathit{CS1})$ from $\mathcal{D}_{\mathsf{KB}}$ yields a consistent KB. For example, the interpretation $\mathcal{I} = (\mathbf{C}, \cdot^{\mathcal{I}})$ below is a model of $\langle \mathcal{D}_{\mathsf{KB}} \setminus \{\mathsf{TakeCourse}(123, 456), \mathsf{GradCourse}(\mathit{CS1})\}, \mathcal{T}_{\mathsf{KB}} \rangle$. Note that $\mathcal{I}$ uses a fresh element n to witness a course taken by 123 and satisfy the TGD.

$$\begin{aligned}
c^{\mathcal{I}} =& c \text{ for every } c \in \mathbf{C}\\
\mathsf{Student}^{\mathcal{I}} =& \{(123, \mathit{Ana}, \mathit{Azure}), (123, \mathit{Ann}, \mathit{Azure}), (456, \mathit{Bea}, \mathit{Blue})\}\\
\mathsf{TakeCourse}^{\mathcal{I}} =& \{(123, n), (456, \mathit{CS1})\}\\
\mathsf{UndergradCourse}^{\mathcal{I}} =& \{\mathit{CS1}\}\\
\mathsf{GradCourse}^{\mathcal{I}} =& \emptyset
\end{aligned}$$

Table 1 Syntax and semantics of DL concept and role constructors and TBox axioms: a denotes an individual name, C, C_1 and C_2 concepts (concept names or complex concepts), R and S role names or inverse roles, and n a natural number.

Name	**Syntax**	**Semantics**
Top concept	$\top$	$\Delta^{\mathcal{I}}$
Bottom concept	$\bot$	$\emptyset$
Nominal	$\{a\}$	$\{a^{\mathcal{I}}\}$
Negation	$\neg C$	$\Delta^{\mathcal{I}} \backslash C^{\mathcal{I}}$
Conjunction	$C_1 \sqcap C_2$	$C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$
Disjunction	$C_1 \sqcup C_2$	$C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$
Existential restriction	$\exists R$	$\{d_1 \mid \text{there exists } (d_1, d_2) \in R^{\mathcal{I}}\}$
Qualified existential restriction	$\exists R.C$	$\{d_1 \mid \text{there exists } (d_1, d_2) \in R^{\mathcal{I}} \text{ with } d_2 \in C^{\mathcal{I}}\}$
Qualified universal restriction	$\forall R.C$	$\{d_1 \mid d_2 \in C^{\mathcal{I}} \text{ for all } (d_1, d_2) \in R^{\mathcal{I}}\}$
Qualified cardinality restrictions	$\geq nR.C$	$\{d_1 \mid \lvert\{d_2 \mid (d_1, d_2) \in R^{\mathcal{I}} \text{ and } d_2 \in C^{\mathcal{I}}\}\rvert \geq n\}$
	$\leq nR.C$	$\{d_1 \mid \lvert\{d_2 \mid (d_1, d_2) \in R^{\mathcal{I}} \text{ and } d_2 \in C^{\mathcal{I}}\}\rvert \leq n\}$
Inverse role	R^-	$\{(d_2, d_1) \mid (d_1, d_2) \in R^{\mathcal{I}}\}$
Concept inclusion	$C_1 \sqsubseteq C_2$	$C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$
Role inclusion	$R \sqsubseteq S$	$R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$
Complex role inclusion	$R_1 \circ \dots \circ R_n \sqsubseteq S$	$R_1^{\mathcal{I}} \circ \dots \circ R_n^{\mathcal{I}} \subseteq S^{\mathcal{I}}$
Role disjointness	$R \sqsubseteq \neg S$	$R^{\mathcal{I}} \cap S^{\mathcal{I}} = \emptyset$
Transitivity axiom	$(\mathsf{trans}\ R)$	$R^{\mathcal{I}} \circ R^{\mathcal{I}} \subseteq R^{\mathcal{I}}$

Description logics. Description logics (DLs) form a family of ontology languages that are incomparable with Datalog$^\pm$: on the one hand the DL syntax restricts a lot what can be expressed in TGD-like rules, but on the other hand some DL languages allow for features not available in Datalog$^\pm$ TGDs, such as disjunction in the head, use of constants, or cardinality restrictions. DL predicates can only have arity 1 or 2, and are called *concept* and *role* names, respectively. Note that in DL parlance, constants are called *individual names*, facts are called *assertions*, datasets are called *ABoxes*, and ontologies *TBoxes*. Moreover, the ontology axioms are not written as FO sentences but in a concise syntax, which enforces restrictions on the language expressivity. Table 1 gives the syntax and semantics of the most common DL axioms and we list below a few examples among the many DL languages.

- $\mathcal{ALC}$ is the prototypical DL and allows for concept inclusions (and no other kind of axiom) using the constructors $\top, \bot, \neg C, C_1 \sqcap C_2, C_1 \sqcup C_2, \exists R.C, \forall R.C$.
- The expressive $\mathcal{SHIQ}$ extends $\mathcal{ALC}$ with role inclusions, transitivity axioms, inverse roles, and qualified cardinality restrictions, with the restriction that roles that occur in qualified cardinality restrictions are required to be *simple*, meaning that they should not have any transitive subrole (i.e., R is simple if $\mathcal{T} \models S \sqsubseteq R$ implies $(\mathsf{trans}\ S) \notin \mathcal{T}$).
- One can further extend $\mathcal{SHIQ}$ with nominals, complex role inclusions, role disjointness axioms, a universal role and a *Self* concept, with again some restrictions on the use of complex roles to ensure decidability. The resulting language, called $\mathcal{SROIQ}$, is closely related to OWL, the ontology language standard for the Semantic Web [121].
- The lightweight DL $\mathcal{EL}_\bot$ is the fragment of $\mathcal{ALC}$ that only allows for $\top, \bot, C_1 \sqcap C_2, \exists R.C$.
- Another family of lightweight DLs is the DL-Lite family. In particular:
  - A DL-Lite$_{\mathsf{core}}$ TBox consists of concept inclusions of the form $B \sqsubseteq C$ where $B := A \mid \exists R$ with A a concept name and R a role name or inverse role, and $C := B \mid \neg B$.
  - DL-Lite$_\mathcal{R}$ extends DL-Lite$_{\mathsf{core}}$ with role inclusions and role disjointness axioms.
  - DL-Lite$_{\mathcal{R},\sqcap}$[1] extends DL-Lite$_\mathcal{R}$ with concept inclusions of the form $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq A$.

An interpretation $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ is a *model* of a DL KB $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ if

- $\mathcal{I}$ satisfies all facts in $\mathcal{D}$ (i.e., $a^\mathcal{I} \in A^\mathcal{I}$ and $(a^\mathcal{I}, b^\mathcal{I}) \in R^\mathcal{I}$ for all concept and role assertions $A(a)$ and $R(a,b)$ in $\mathcal{D}$) and
- $\mathcal{I}$ satisfies all axioms in $\mathcal{T}$ as explained in Table 1.

The only difference with the semantics of Datalog$^\pm$ KBs is that we do not need to make the standard name assumption, nor the unique name assumption (even though these assumptions are sometimes made): in general, two different constants from $\mathcal{D}$ can be interpreted by the same element of $\Delta^\mathcal{I}$.

▶ **Example 3.** The database of Example 1 can be "translated" into the $\mathcal{SHIQ}$ KB $\langle \mathcal{D}_{\mathsf{DL}}, \mathcal{T}_{\mathsf{DL}} \rangle$. Since predicates of arity greater than 2 are not allowed, the ternary predicate Student is replaced by a concept Student that stores the student identifiers and two roles FirstName and LastName that associate identifiers with first and last names. (ax_1)-(ax_2) state that students have at most one first name and one last name, (ax_3) that every student takes some course, (ax_4) that everything in the range of TakeCourse is a graduate course or an undergraduate course, (ax_5) that the concept Student and the range of TakeCourse are disjoint, and (ax_6) that GradCourse and UndergradCourse are disjoint.

---

[1] DL-Lite$^\mathcal{H}_{\mathsf{horn}}$ [12], the variant of DL-Lite$_{\mathcal{R},\sqcap}$ [58] without role disjointness axioms, may be more common. However, both languages have the same computational behavior [12].

$$\begin{aligned}
\mathcal{T}_{\mathsf{DL}} = \{&\mathsf{Student} \sqsubseteq \leq 1\mathsf{FirstName}.\top, \quad \mathsf{Student} \sqsubseteq \leq 1\mathsf{LastName}.\top, && (\mathsf{ax}_1)\text{-}(\mathsf{ax}_2)\\
&\mathsf{Student} \sqsubseteq \exists \mathsf{TakeCourse}.\top, && (\mathsf{ax}_3)\\
&\exists \mathsf{TakeCourse}^-.\top \sqsubseteq \mathsf{GradCourse} \sqcup \mathsf{UndergradCourse}, && (\mathsf{ax}_4)\\
&\mathsf{Student} \sqcap \exists \mathsf{TakeCourse}^-.\top \sqsubseteq \bot, && (\mathsf{ax}_5)\\
&\mathsf{GradCourse} \sqcap \mathsf{UndergradCourse} \sqsubseteq \bot\} && (\mathsf{ax}_6)\\
\mathcal{D}_{\mathsf{DL}} = \{&\mathsf{Student}(123), \mathsf{Student}(456), \mathsf{LastName}(123, \mathit{Azure}), \mathsf{LastName}(456, \mathit{Blue}),\\
&\mathsf{FirstName}(123, \mathit{Ana}), \mathsf{FirstName}(123, \mathit{Ann}), \mathsf{FirstName}(456, \mathit{Bea}),\\
&\mathsf{TakeCourse}(123, 456), \mathsf{TakeCourse}(456, \mathit{CS1}),\\
&\mathsf{GradCourse}(\mathit{CS1}), \mathsf{UndergradCourse}(\mathit{CS1})\}
\end{aligned}$$

As in Example 2, the KB is inconsistent but removing the facts $\mathsf{TakeCourse}(123, 456)$ and $\mathsf{GradCourse}(\mathit{CS1})$ makes it consistent, when interpreted without the unique name assumption: a model of $\langle \mathcal{D}_{\mathsf{DL}} \setminus \{\mathsf{TakeCourse}(123, 456), \mathsf{GradCourse}(\mathit{CS1})\}, \mathcal{T}_{\mathsf{DL}}\rangle$ simply interprets *Ann* and *Ana* by the same domain element (however, if interpreted under the unique name assumption, the KB would be inconsistent).

Queries. An *FO query* is simply an FO formula, whose free variables are called the *answer variables*. A tuple $\vec{a} = (a_1, \dots, a_n)$ is a (certain) *answer* to a query $q(\vec{x})$ with answer variables $\vec{x} = (x_1, \dots, x_n)$ over a KB $\mathcal{K} = \langle \mathcal{D}, \mathcal{T}\rangle$ if $\mathcal{K} \models q(\vec{a})$, where $q(\vec{a})$ is the FO sentence obtained by replacing each x_i by a_i in $q(\vec{x})$. For a database $\mathcal{K} = \langle \mathcal{D}, \mathcal{T}\rangle$, we simply define $\mathcal{K} \models q(\vec{a})$ by $\mathcal{I}_{\mathcal{D}} \models q(\vec{a})$. A *conjunctive query (CQ)* is an FO query of the form $q(\vec{x}) = \exists \vec{y} \varphi(\vec{x}, \vec{y})$ where $\varphi(\vec{x}, \vec{y})$ is a conjunction of relational atoms. A *union of conjunctive queries (UCQ)* has form $q(\vec{x}) = \bigvee_{i=1}^{n} q_i(\vec{x})$ where each $q_i(\vec{x})$ is a CQ. We often see a UCQ as a finite set of CQs. A query that has no answer variable is called a *Boolean query* (BCQ, BUCQ).

▶ **Example 4.** Consider the dataset $\mathcal{D}'_{\mathsf{DL}} = \{\mathsf{Student}(123), \mathsf{TakeCourse}(456, \mathit{CS1})\}$ and the ontology $\mathcal{T}_{\mathsf{DL}}$ from Example 3. The CQ $q(x) = \exists y \mathsf{TakeCourse}(x, y)$ has two certain answers over $\langle \mathcal{D}'_{\mathsf{DL}}, \mathcal{T}_{\mathsf{DL}}\rangle$: (123) and (456).

FO- and UCQ-rewritability. We say that CQ answering under an ontology language $\mathcal{L}$ is *UCQ-rewritable* (resp. *FO-rewritable*) if for every BCQ q and ontology $\mathcal{T}$ in $\mathcal{L}$, there exists a BUCQ (resp. an FO query) q_r such that for every dataset $\mathcal{D}$, $\langle \mathcal{D}, \mathcal{T}\rangle \models q$ iff $\mathcal{I}_{\mathcal{D}} \models q_r$. In the BUCQ case, we denote the set of BCQs in q_r by $\mathsf{Rew}(q, \mathcal{T})$.

▶ **Fact 5.** *CQ answering under* L, A *and* S *is UCQ-rewritable (cf. [131, Def. 2.19 and Properties 4, 5, 6], and [104, Sec. 3.2.1]), as well as under DL-Lite*$_{\mathcal{R},\sqcap}$ *and its sub-languages [58].*

▶ **Example 6.** Given the DL-Lite$_{\mathsf{core}}$ ontology $\mathcal{T} = \{\mathsf{Student} \sqsubseteq \exists \mathsf{TakeCourse}\}$ and BCQ $q = \exists y \mathsf{TakeCourse}(123, y)$, $\mathsf{Rew}(q, \mathcal{T}) = \{q, q'\}$ with $q' = \mathsf{Student}(123)$.

▶ **Proposition 7** (Consistency checking via UCQ-rewritability)**.** *For every Datalog*$^{\pm}$ *fragment* X*, if CQ answering under* X *is UCQ-rewritable, then for every ontology* $\mathcal{T}$ *in* $\mathsf{X}_{\bot}$*, there exists a BUCQ* $q^{\mathcal{T}}_{\bot}$ *such that for every dataset* $\mathcal{D}$*,* $\langle \mathcal{D}, \mathcal{T}\rangle \models \bot$ *iff* $\mathcal{I}_{\mathcal{D}} \models q^{\mathcal{T}}_{\bot}$*.*

Proof. For every Datalog$^{\pm}$ KB $\langle \mathcal{D}, \mathcal{T}\rangle$, $\langle \mathcal{D}, \mathcal{T}\rangle \models \bot$ iff there exists $\beta[\vec{x}] \to \bot \in \mathcal{T}$ such that $\langle \mathcal{D}, \mathcal{T}^+\rangle \models \exists \vec{x} \beta[\vec{x}]$, where $\mathcal{T}^+$ is the set of TGDs in $\mathcal{T}$, so if $\mathcal{T}^+$ is in X and CQ answering under X is UCQ-rewritable, $\langle \mathcal{D}, \mathcal{T}\rangle \models \bot$ iff $\mathcal{I}_{\mathcal{D}} \models q'$ for some $q' \in \bigcup_{\beta[\vec{x}] \to \bot \in \mathcal{T}} \mathsf{Rew}(\exists \vec{x} \beta[\vec{x}], \mathcal{T}^+)$. ◀

2.2 Computational complexity

We consider two ways of measuring the computational complexity of decision problems: the *data complexity* is measured w.r.t. the size of the dataset $\mathcal{D}$ while the *combined complexity* takes into account the size of the whole input.

Complexity classes. We will refer to the following complexity classes:

$$\begin{tikzcd}[column sep=tiny, row sep=tiny]
 & \mathrm{NP} \arrow[dr, phantom, "\subsetneq"] & & & \Sigma_2^{\mathrm{P}} \arrow[dr, phantom, "\subsetneq"] & & & \mathrm{NExp} \arrow[dr, phantom, "\subsetneq"] & \\
\mathsf{AC}^0 \subseteq \mathrm{L} \subseteq \mathrm{NL} \subseteq \mathrm{P} \arrow[ur, phantom, "\subsetneq"] \arrow[dr, phantom, "\subsetneq"] & & \mathrm{DP} \subseteq \Theta_2^{\mathrm{P}} \subseteq \Delta_2^{\mathrm{P}} \arrow[ur, phantom, "\subsetneq"] \arrow[dr, phantom, "\subsetneq"] & & & \cdots \subseteq \mathrm{PSpace} \subseteq \mathrm{Exp} \arrow[ur, phantom, "\subsetneq"] \arrow[dr, phantom, "\subsetneq"] & & & \mathrm{DExp} \subseteq \cdots \subseteq 2\mathrm{Exp} \\
 & \mathrm{coNP} \arrow[ur, phantom, "\subsetneq"] & & & \Pi_2^{\mathrm{P}} \arrow[ur, phantom, "\subsetneq"] & & & \mathrm{coNExp} \arrow[ur, phantom, "\subsetneq"] &
\end{tikzcd}$$

- Below P:
  - AC^0: problems solvable by a family of circuits of constant depth and polynomial size w.r.t. the size of the input, with unbounded-fanin AND and OR gates.
  - L: problems solvable in logarithmic space w.r.t. the size of the input.
  - NL: problems solvable in non-deterministic logarithmic space. These are problems for which there exists a certificate of polynomial size which can be verified by a deterministic logarithmic-space bounded Turing machine that has an additional read-only read-once input tape. It is known (by Immerman–Szelepcsényi theorem) that NL = coNL.
- The polynomial hierarchy and other classes between P and PSpace:
  - $\mathrm{P} = \Sigma_0^{\mathrm{P}}$: problems which are solvable in polynomial time w.r.t. the size of the input.
  - $\mathrm{NP} = \Sigma_1^{\mathrm{P}}$: problems which are solvable in non-deterministic polynomial time. These are problems for which there exists a certificate of polynomial size which can be verified by a deterministic polynomial-time Turing machine.
  - $\mathrm{coNP} = \Pi_1^{\mathrm{P}}$: problems whose complement is in NP.
  - In general, for $i \geq 0$, $\Delta_{i+1}^{\mathrm{P}} = \mathrm{P}^{\Sigma_i^{\mathrm{P}}}$ is the class of problems solvable in polynomial time with access to a Σ_i^{P} oracle, $\Sigma_{i+1}^{\mathrm{P}} = \mathrm{NP}^{\Sigma_i^{\mathrm{P}}}$, and $\Pi_{i+1}^{\mathrm{P}} = \mathrm{coNP}^{\Sigma_i^{\mathrm{P}}}$.
  - $\Theta_{i+1}^{\mathrm{P}}$ is the class of problems solvable in polynomial time with at most logarithmically many calls to a Σ_i^{P} oracle. It is known that these are problems solvable by polynomially many *independent* (parallel) calls to a Σ_i^{P} oracle [47].
  - DP: problems that are the intersection of a problem in NP and a problem in coNP.
  - PSpace: problems which are solvable in polynomial space. It is known (by Savitch's theorem) that PSpace = NPSpace.
- The (weak) exponential hierarchy and other classes between Exp and 2Exp:
  - $\mathrm{Exp} = \Sigma_0^{\mathrm{Exp}}$: problems which are solvable in exponential time, i.e., in $O(2^{p(n)})$ where p is a polynomial function and n is the input size.
  - $\mathrm{NExp} = \Sigma_1^{\mathrm{Exp}}$: problems which are solvable in non-deterministic exponential time.
  - $\mathrm{coNExp} = \Pi_1^{\mathrm{Exp}}$: problems whose complement is in NExp.
  - In general, for $i \geq 0$, $\Delta_{i+1}^{\mathrm{Exp}} = \mathrm{Exp}^{\Sigma_i^{\mathrm{P}}}$ is the class of problems solvable in exponential time with access to a Σ_i^{P} oracle, $\Sigma_{i+1}^{\mathrm{Exp}} = \mathrm{NExp}^{\Sigma_i^{\mathrm{P}}}$, and $\Pi_{i+1}^{\mathrm{Exp}} = \mathrm{coNExp}^{\Sigma_i^{\mathrm{P}}}$.
  - DExp: problems that are the intersection of a problem in NExp and one in coNExp.
  - $\mathrm{P}^{\mathrm{NExp}}$: problems that are solvable in polynomial time with an NExp oracle. It is known (since the strong exponential hierarchy collapses [89]) that $\mathrm{NP}^{\mathrm{NExp}} = \mathrm{P}^{\mathrm{NExp}}$.
  - 2Exp: problems which are solvable in double-exponential time.

Table 2 Complexity of BCQ entailment. All non-"in" entries are completeness results. For Datalog$^{\pm}$ fragments, references are given for X rather than $\mathsf{X}_\perp$, cf. Remark 9.

		Data complexity		Combined complexity	
DLs	DL-Lite$_{\mathcal{R}}$	in AC^0	[12, Th. 8.9]	NP	[58, Th. 44] + easy extension
	DL-Lite$_{\mathcal{R},\sqcap}$	in AC^0	[12, Th. 8.9]	NP	[58, Th. 44] + easy extension
	$\mathcal{EL}_\perp$	P	[125, Th. 2]	NP	[125, Th. 2]
	$\mathcal{ALC}$	coNP	[120, Th. 4.6]	Exp	[110, Th. 4]
	$\mathcal{SHIQ}$	coNP	[84, Cor. 36]	2Exp	[84, Cor. 33]
Datalog$^{\pm}$	$\mathsf{L}_\perp$	in AC^0	[56, Cor. 10]	PSpace	[55, Th. 12.7]
	$\mathsf{A}_\perp$	in AC^0	[104, Pr. 3.3]	NExp	[104, Pr. 3.3]
	$\mathsf{G}_\perp$	P	[54, Th. 6.1]	2Exp	[54, Th. 6.1]
	$\mathsf{S}_\perp$	in AC^0	[57, Th. 3.5]	Exp	[57, Th. 3.3 & 3.4]
	$\mathsf{F}_\perp$	P	[70, Th. 4.4]	Exp	[70, Th. 4.5]
Database		in AC^0	[135, Tab. 1]	NP	[63, Th. 7]

Complexity of BCQ entailment and consistency checking. We illustrate the computational complexity of the decision problems we consider on a few selected languages from Figure 1. Table 2 shows the data and combined complexity of *BCQ entailment*, the decision problem which takes as input a database or KB $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ and a Boolean conjunctive query q and outputs "yes" if $\mathcal{K} \models q$, and "no" otherwise. Table 3 shows the data and combined complexity of *consistency checking*, the decision problem which takes as input a database or KB $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ and outputs "yes" if $\mathcal{K}$ is consistent, and "no" otherwise.

▶ Remark 8 (Database setting). In the database setting, ICs do not play any role in BCQ evaluation, so we do not distinguish between the different IC languages in Table 2. Regarding consistency checking, $\langle \mathcal{D}, \mathcal{T} \rangle \models \perp$ iff $\mathcal{I}_{\mathcal{D}} \models \bigvee_{\phi \in \mathcal{T}} \neg\phi$. This straightforwardly yields the AC^0 data complexity upper bounds and the coNP upper bounds for the IC languages such that the $\neg\phi$ are BCQs with safe negation and inequalities (i.e., for universal constraints).

▶ Remark 9 (Datalog$^{\pm}$: X versus $\mathsf{X}_\perp$). For Datalog$^{\pm}$ fragments, most complexity results have actually been shown for BCQ entailment in X rather than in $\mathsf{X}_\perp$, but they extend to both *inconsistency* checking and BCQ entailment in $\mathsf{X}_\perp$:

- Lower bounds: If $\mathcal{T}$ contains only TGDs, for any BCQ q, $\langle \mathcal{D}, \mathcal{T} \rangle \models q$ iff $\langle \mathcal{D}, \mathcal{T} \cup \{q \rightarrow \perp\} \rangle \models \perp$, hence inconsistency checking in $\mathsf{X}_\perp$ is at least as hard as BCQ entailment in X.
- Combined complexity and P data complexity upper bounds: Recall that (i) $\langle \mathcal{D}, \mathcal{T} \rangle \models \perp$ iff there exists $\beta[\vec{x}] \rightarrow \perp \in \mathcal{T}$ such that $\langle \mathcal{D}, \mathcal{T}^+ \rangle \models \exists \vec{x} \beta[\vec{x}]$, where $\mathcal{T}^+$ is the set of TGDs in $\mathcal{T}$ (meaning that $\mathcal{T}^+$ is in X if $\mathcal{T}$ is in $\mathsf{X}_\perp$), and (ii) for every BCQ q, $\langle \mathcal{D}, \mathcal{T} \rangle \models q$ iff either $\langle \mathcal{D}, \mathcal{T}^+ \rangle \models q$ or $\langle \mathcal{D}, \mathcal{T} \rangle \models \perp$ (since an inconsistent KB entails every query).
  - We thus only need a linear number of checks of BCQ entailment under X to decide (in)consistency checking and BCQ entailment under $\mathsf{X}_\perp$. This yields the P data complexity upper bounds and the Exp and 2Exp combined complexity upper bounds.
  - For the PSpace upper bound for $\mathsf{L}_\perp$, guess an NC $\beta[\vec{x}] \rightarrow \perp \in \mathcal{T}$ (or q in the BCQ entailment case) and check in PSpace that $\langle \mathcal{D}, \mathcal{T}^+ \rangle \models \exists \vec{x} \beta[\vec{x}]$ (or $\langle \mathcal{D}, \mathcal{T}^+ \rangle \models q$). The result follows from the fact that NPSpace = PSpace.
  - For the NExp/coNExp upper bounds for $\mathsf{A}_\perp$, proceed as above, but guess the BCQ $\exists \vec{x} \beta[\vec{x}]$ (or q) together with the certificate that shows that it is entailed by $\langle \mathcal{D}, \mathcal{T}^+ \rangle$, which can be verified in exponential time since BCQ entailment in A is in NExp.

Table 3 Complexity of consistency checking. All non-"in" entries are completeness results.

		Data complexity		Combined complexity	
DLs	DL-Lite$_\mathcal{R}$	in AC^0	[12, Th. 8.3]	NL	[12, Th. 8.2]
	DL-Lite$_{\mathcal{R},\sqcap}$	in AC^0		P	
	$\mathcal{EL}_\bot$	P	[13, Th. 4][59, Th. 7]	P	[13, Th. 4][59, Th. 7]
	$\mathcal{ALC}$	NP	[90, Th. 2]	Exp	[132, Cor. 3.1]
	$\mathcal{SHIQ}$	NP		Exp	[132, Cor. 6.30]
Datalog$^\pm$	$\mathsf{L}_\bot$	in AC^0	Table 2 and Remark 9	PSpace	Table 2 and Remark 9
	$\mathsf{A}_\bot$	in AC^0		coNExp	
	$\mathsf{G}_\bot$	P		2Exp	
	$\mathsf{S}_\bot$	in AC^0		Exp	
	$\mathsf{F}_\bot$	P		Exp	
ICs	FD	in AC^0	FO query evaluation [135, Tab. 1]	in L	folklore (e.g., [11])
	DC	in AC^0		coNP	BCQ eval. [63, Th. 7] extend. to BCQ with safe $\neg$ and $\neq$
	full TGD	in AC^0		coNP	
	UC	in AC^0		coNP	
	TGD	in AC^0		Π_2^p	[123, Th. 3.1]

- AC^0 upper bounds: If $\mathsf{X}_\bot \in \{\mathsf{L}_\bot, \mathsf{A}_\bot, \mathsf{S}_\bot\}$, since CQ answering under X is UCQ-rewritable (Fact 5), by Proposition 7, for every ontology $\mathcal{T}$ in $\mathsf{X}_\bot$, there exists a BUCQ $q_\bot^\mathcal{T}$ such that for every dataset $\mathcal{D}$, $\langle \mathcal{D}, \mathcal{T} \rangle \models \bot$ iff $\mathcal{I}_\mathcal{D} \models q_\bot^\mathcal{T}$. Moreover, for every BCQ q, $\langle \mathcal{D}, \mathcal{T} \rangle \models q$ iff $\mathcal{I}_\mathcal{D} \models q_\bot^\mathcal{T} \vee \bigvee_{q' \in \mathsf{Rew}(q, \mathcal{T}^+)} q'$ where $\mathcal{T}^+$ is the set of TGDs in $\mathcal{T}$. This yields the AC^0 data complexity upper bound for consistency checking and BCQ entailment in $\mathsf{X}_\bot$.

3 Dataset repairs of inconsistent databases or knowledge bases

In this section, we define the central notion of a (dataset) repair then present some characterizations of repairs based on the conflict hypergraph of a database or KB and on abstract argumentation frameworks.

3.1 Definition

We assume that all logical theories (sets of ICs or ontologies) we consider are satisfiable and reliable, so that if a database or KB $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ is inconsistent, the problem stems from the dataset $\mathcal{D}$. In this context, it makes sense to define *(dataset) repairs* of $\mathcal{K}$ as consistent datasets that minimally differ from $\mathcal{D}$. In the database setting, one can restore consistency by deleting and/or adding facts to $\mathcal{D}$ (if $\mathcal{D} = \{P(c)\}$ and $\mathcal{T} = \{P(x) \rightarrow Q(x)\}$, we can obtain a consistent database by removing $P(c)$ or by adding $Q(c)$ to $\mathcal{D}$), except for anti-monotone constraints for which only deleting facts can solve a constraint violation. In the KB setting, because of the open world assumption, repairs can only be obtained by removing facts.

▶ **Definition 10** (Dataset repairs). *A* symmetric difference repair*, or Δ-repair, of $\mathcal{D}$ w.r.t. $\mathcal{T}$ is a $\mathcal{T}$-consistent dataset $\mathcal{R}$ such that there is no $\mathcal{T}$-consistent $\mathcal{R}'$ such that $\mathcal{R}'\Delta\mathcal{D} \subsetneq \mathcal{R}\Delta\mathcal{D}$, where Δ is the symmetric difference operator: $S_1 \Delta S_2 = (S_1 \setminus S_2) \cup (S_2 \setminus S_1)$.*

A subset repair *(resp.* superset repair*), or $\subseteq$-repair (resp. $\supseteq$-repair), of $\mathcal{D}$ w.r.t. $\mathcal{T}$ is a $\mathcal{T}$-consistent dataset $\mathcal{R}$ such that $\mathcal{R} \subseteq \mathcal{D}$ (resp. $\mathcal{R} \supseteq \mathcal{D}$) and there is no $\mathcal{T}$-consistent $\mathcal{R}'$ such that $\mathcal{R} \subsetneq \mathcal{R}' \subseteq \mathcal{D}$ (resp. $\mathcal{R} \supsetneq \mathcal{R}' \supseteq \mathcal{D}$).*

For $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ and $x \in \{\Delta, \subseteq, \supseteq\}$, we use $S\text{-}Rep_x(\mathcal{K})$ or $S\text{-}Rep_x(\mathcal{D}, \mathcal{T})$ to denote the set of all x-repairs of $\mathcal{K}$, i.e., x-repairs of $\mathcal{D}$ w.r.t. $\mathcal{T}$. When $\mathcal{K}$ is a KB or a database with anti-monotone ICs, $S\text{-}Rep_\Delta(\mathcal{K}) = S\text{-}Rep_\subseteq(\mathcal{K})$ and $S\text{-}Rep_\supseteq(\mathcal{K}) = \emptyset$ if $\mathcal{K}$ is inconsistent, so we can simply write $S\text{-}Rep(\mathcal{K})$ for $S\text{-}Rep_\subseteq(\mathcal{K})$ and talk about "repairs" instead of $\subseteq$-repairs.

In the above definition, the "S" in $S\text{-}Rep_x(\mathcal{K})$ can be understood as "standard" or "set-inclusion-based", in contrast with the optimal repairs that we consider in Section 5 and which compare datasets using other criteria than set-inclusion. It follows from the definition that $S\text{-}Rep_\subseteq(\mathcal{K}) \subseteq S\text{-}Rep_\Delta(\mathcal{K})$ and $S\text{-}Rep_\supseteq(\mathcal{K}) \subseteq S\text{-}Rep_\Delta(\mathcal{K})$.

▶ **Example 11.** Recall $\langle \mathcal{D}_{\mathsf{DB}}, \mathcal{T}_{\mathsf{DB}} \rangle$ from Example 1. The Δ-repairs of $\mathcal{D}_{\mathsf{DB}}$ w.r.t. $\mathcal{T}_{\mathsf{DB}}$ are as follows, among which only $\mathcal{R}_1$ and $\mathcal{R}_1'$ are $\subseteq$-repairs:

$$\begin{aligned}
\mathcal{R}_1 &= \{\mathsf{Student}(456, \mathit{Bea}, \mathit{Blue}), \mathsf{TakeCourse}(456, \mathit{CS1}), \mathsf{UndergradCourse}(\mathit{CS1})\}\\
\mathcal{R}_1' &= \{\mathsf{Student}(456, \mathit{Bea}, \mathit{Blue}), \mathsf{TakeCourse}(456, \mathit{CS1}), \mathsf{GradCourse}(\mathit{CS1})\}\\
\mathcal{R}_2 &= \{\mathsf{Student}(123, \mathit{Ana}, \mathit{Azure}), \mathsf{TakeCourse}(123, 456), \mathsf{GradCourse}(456),\\
&\qquad \mathsf{TakeCourse}(456, \mathit{CS1}), \mathsf{UndergradCourse}(\mathit{CS1})\}\\
\mathcal{R}_2' &= \{\mathsf{Student}(123, \mathit{Ana}, \mathit{Azure}), \mathsf{TakeCourse}(123, 456), \mathsf{GradCourse}(456),\\
&\qquad \mathsf{TakeCourse}(456, \mathit{CS1}), \mathsf{GradCourse}(\mathit{CS1})\}\\
\mathcal{R}_3 &= \{\mathsf{Student}(123, \mathit{Ana}, \mathit{Azure}), \mathsf{TakeCourse}(123, 456), \mathsf{UndergradCourse}(456),\\
&\qquad \mathsf{TakeCourse}(456, \mathit{CS1}), \mathsf{UndergradCourse}(\mathit{CS1})\}\\
\mathcal{R}_3' &= \{\mathsf{Student}(123, \mathit{Ana}, \mathit{Azure}), \mathsf{TakeCourse}(123, 456), \mathsf{UndergradCourse}(456),\\
&\qquad \mathsf{TakeCourse}(456, \mathit{CS1}), \mathsf{GradCourse}(\mathit{CS1})\}\\
{\mathcal{R}_4}^c &= \{\mathsf{Student}(123, \mathit{Ana}, \mathit{Azure}), \mathsf{Student}(456, \mathit{Bea}, \mathit{Blue}), \mathsf{TakeCourse}(123, c),\\
&\qquad \mathsf{GradCourse}(c), \mathsf{TakeCourse}(456, \mathit{CS1}), \mathsf{UndergradCourse}(\mathit{CS1})\} \text{ for } c \in \mathbf{C}\\
{\mathcal{R}_4'}^c &= \{\mathsf{Student}(123, \mathit{Ana}, \mathit{Azure}), \mathsf{Student}(456, \mathit{Bea}, \mathit{Blue}), \mathsf{TakeCourse}(123, c),\\
&\qquad \mathsf{GradCourse}(c), \mathsf{TakeCourse}(456, \mathit{CS1}), \mathsf{GradCourse}(\mathit{CS1})\} \text{ for } c \in \mathbf{C}\\
{\mathcal{R}_5}^c &= \{\mathsf{Student}(123, \mathit{Ana}, \mathit{Azure}), \mathsf{Student}(456, \mathit{Bea}, \mathit{Blue}), \mathsf{TakeCourse}(123, c),\\
&\qquad \mathsf{UndergradCourse}(c), \mathsf{TakeCourse}(456, \mathit{CS1}), \mathsf{UndergradCourse}(\mathit{CS1})\} \text{ for } c \in \mathbf{C}\\
{\mathcal{R}_5'}^c &= \{\mathsf{Student}(123, \mathit{Ana}, \mathit{Azure}), \mathsf{Student}(456, \mathit{Bea}, \mathit{Blue}), \mathsf{TakeCourse}(123, c),\\
&\qquad \mathsf{UndergradCourse}(c), \mathsf{TakeCourse}(456, \mathit{CS1}), \mathsf{GradCourse}(\mathit{CS1})\} \text{ for } c \in \mathbf{C}
\end{aligned}$$

plus for each repair that contains $\mathsf{Student}(123, \mathit{Ana}, \mathit{Azure})$, the repair obtained by replacing this fact by $\mathsf{Student}(123, \mathit{Ann}, \mathit{Azure})$. Note that there are infinitely many repairs since c can be any constant.

Now recall $\langle \mathcal{D}_{\mathsf{DL}}, \mathcal{T}_{\mathsf{DL}} \rangle$ from Example 3. The repairs of $\mathcal{D}_{\mathsf{DL}}$ w.r.t. $\mathcal{T}_{\mathsf{DL}}$ are as follows:

$$\begin{aligned}
\mathcal{R}_1 &= \mathcal{B} \cup \{\mathsf{Student}(456), \mathsf{UndergradCourse}(\mathit{CS1})\}\\
\mathcal{R}_2 &= \mathcal{B} \cup \{\mathsf{Student}(456), \mathsf{GradCourse}(\mathit{CS1})\}\\
\mathcal{R}_3 &= \mathcal{B} \cup \{\mathsf{TakeCourse}(123, 456), \mathsf{UndergradCourse}(\mathit{CS1})\}\\
\mathcal{R}_4 &= \mathcal{B} \cup \{\mathsf{TakeCourse}(123, 456), \mathsf{GradCourse}(\mathit{CS1})\}
\end{aligned}$$

with

$$\begin{aligned}
\mathcal{B} = \{&\mathsf{Student}(123), \mathsf{LastName}(123, \mathit{Azure}), \mathsf{LastName}(456, \mathit{Blue}), \mathsf{FirstName}(123, \mathit{Ana}),\\
&\mathsf{FirstName}(123, \mathit{Ann}), \mathsf{FirstName}(456, \mathit{Bea}), \mathsf{TakeCourse}(456, \mathit{CS1})\}
\end{aligned}$$

Note that if we made the unique name assumption, we would obtain eight repairs $\{\mathcal{R}'_i, \mathcal{R}''_i \mid 1 \leq i \leq 4\}$ with $\mathcal{R}'_i = \mathcal{R}_i \setminus \{\mathsf{FirstName}(123, Ana)\}$ and $\mathcal{R}''_i = \mathcal{R}_i \setminus \{\mathsf{FirstName}(123, Ann)\}$.

▶ Remark 12 (KBs with closed predicates). KBs with *closed predicates*, in which some predicates are interpreted under the closed world assumption, have also been studied (e.g., [111, 119, 39]). In this context, it makes sense to consider Δ-repairs also for KBs. However, the investigation of inconsistency handling for such KBs has just started, and focused on $\subseteq$-repairs [116].

3.2 Repair characterization via the conflict hypergraph

If $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ is a KB or a database with anti-monotone ICs (hence for which Δ-repairs and $\subseteq$-repairs coincide), it is well known that the (subset) repairs of $\mathcal{D}$ w.r.t. $\mathcal{T}$ are the maximal independent sets of the *conflict hypergraph* whose vertices are the facts in $\mathcal{D}$ and whose hyperedges are the *conflicts* of $\mathcal{K}$:

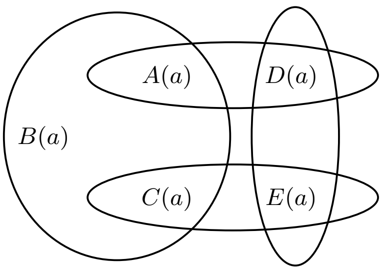
▶ **Definition 13** (Conflicts of a KB or database with anti-monotone constraints). *Let $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ be a KB or a database with anti-monotone ICs. A* conflict *of $\mathcal{K}$ is a $\subseteq$-minimal $\mathcal{T}$-inconsistent subset of $\mathcal{D}$. We denote by $Conf(\mathcal{K})$ or $Conf(\mathcal{D}, \mathcal{T})$ the set of all conflicts of $\mathcal{K}$.*

▶ **Proposition 14.** *Let $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ be a KB or a database with anti-monotone ICs and $\mathcal{R} \subseteq \mathcal{D}$. Then $\mathcal{R} \in S\text{-}Rep(\mathcal{K})$ iff $\mathcal{R}$ is a maximal independent set of $\mathcal{G} = (\mathcal{D}, Conf(\mathcal{K}))$.*

▶ **Example 15.** Let $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ with $\mathcal{D} = \{A(a), B(a), C(a), D(a), E(a)\}$ and

$$\mathcal{T} = \{A \sqcap B \sqcap C \sqsubseteq \bot, A \sqcap D \sqsubseteq \bot, C \sqcap E \sqsubseteq \bot, D \sqcap E \sqsubseteq \bot\}.$$

The conflict hypergraph $\mathcal{G} = (\mathcal{D}, Conf(\mathcal{K}))$ is pictured below, and the repairs of $\mathcal{K}$ are its three maximal independent sets.



Maximal independent sets of $\mathcal{G}$:
$\mathcal{R}_1 = \{A(a), B(a), E(a)\}$
$\mathcal{R}_2 = \{A(a), C(a)\}$
$\mathcal{R}_3 = \{B(a), C(a), D(a)\}$

For databases with constraints which are not anti-monotone, which may be violated because of the *absence* of some fact, the notion of conflict hypergraph is less direct. However, for universal constraints (which have no existential quantifier in the head), we can define a conflict hypergraph whose vertices are literals (i.e., facts or negated facts) that express whether a fact belongs to the dataset or not [32]. Conflicts can then be defined as minimal sets of literals that necessarily lead to a constraint violation.

▶ **Definition 16** (Conflicts of a database with universal constraints). *Let $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ be a database such that $\mathcal{T}$ is a set of UCs. Let $Facts^{\mathcal{T}}_{\mathcal{D}}$ be the union of $\mathcal{D}$ and facts of the form $P(c_1, \ldots, c_n)$ where P is a predicate that occurs in $\mathcal{T}$, and $c_1, \ldots, c_n$ are constants that occur in $\mathcal{D}$, and let $Lits^{\mathcal{T}}_{\mathcal{D}} = \mathcal{D} \cup \{\neg\alpha \mid \alpha \in Facts^{\mathcal{T}}_{\mathcal{D}} \setminus \mathcal{D}\}$ be the set of* literals *of $\mathcal{D}$. A* conflict *of $\mathcal{K}$ is a $\subseteq$-minimal set $\mathcal{C} \subseteq Lits^{\mathcal{T}}_{\mathcal{D}}$ such that for every interpretation $\mathcal{I}$, if $\mathcal{I} \models \mathcal{C}$, then $\mathcal{I} \not\models \mathcal{T}$.*

▶ **Proposition 17** ([32, Prop. 2]). *Let $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ be a database such that $\mathcal{T}$ is a set of UCs and $\mathcal{R} \subseteq \mathit{Facts}^{\mathcal{T}}_{\mathcal{D}}$. Let $\mathit{Int}_{\mathcal{D}}(\mathcal{R}) = (\mathcal{R} \cap \mathcal{D}) \cup \{\neg\alpha \mid \alpha \in \mathit{Facts}^{\mathcal{T}}_{\mathcal{D}} \setminus (\mathcal{R} \cup \mathcal{D})\}$ be the set of literals upon which $\mathcal{R}$ and $\mathcal{D}$ agree. Then $\mathcal{R} \in \textit{S-Rep}_{\Delta}(\mathcal{K})$ iff $\mathit{Int}_{\mathcal{D}}(\mathcal{R})$ is a maximal independent set of $\mathcal{G} = (\mathit{Lits}^{\mathcal{T}}_{\mathcal{D}}, \mathit{Conf}(\mathcal{K}))$.*

▶ **Example 18** ([32]). Let $\mathcal{D} = \{A(a), B(a)\}$ and $\mathcal{T}$ be the following set of UCs:

$$\mathcal{T} = \{A(x) \to C(x),\ B(x) \to D(x),\ C(x) \land D(x) \to \bot\}$$

The conflicts, conflict graph (binary conflicts) and Δ-repairs of $\mathcal{D}$ w.r.t. $\mathcal{T}$ are as follows:

$$\mathit{Conf}(\mathcal{D}, \mathcal{T}) = \{\{A(a), \neg C(a)\},\ \{B(a), \neg D(a)\},\ \{A(a), B(a)\}\}$$

$$\textit{S-Rep}_{\Delta}(\mathcal{D}, \mathcal{T}) = \{\emptyset,\ \{A(a), C(a)\},\ \{B(a), D(a)\}\}$$

$$\begin{array}{ccc} A(a) & \text{—} & \neg C(a) \\ | & & \\ B(a) & \text{—} & \neg D(a) \end{array}$$

The first (resp. second) conflict directly violates the first (resp. second) IC. To see why $\{A(a), B(a)\}$ is also a conflict, consider any interpretation $\mathcal{I}$ such that $\mathcal{I} \models \{A(a), B(a)\}$. Then if $\mathcal{I} \not\models C(a)$ or $\mathcal{I} \not\models D(a)$, $\mathcal{I}$ violates the first or second constraint, respectively, and if $\mathcal{I} \models C(a)$ and $\mathcal{I} \models D(a)$, it violates the third constraint.

We can obtain, e.g., the second repair, $\{A(a), C(a)\}$ by considering the maximal independent set of the conflict hypergraph $\{A(a), \neg D(a)\}$, which gives the literals upon which the repair and $\mathcal{D}$ agree (so implicitly also those on which they disagree: $B(a)$ and $\neg C(a)$, which have to be "flipped" to obtain the repair $\{A(a), C(a)\}$).

3.3 Connections with abstract argumentation

A long line of work connects inconsistent KBs and *argumentation frameworks*, a formalism that aims at representing and reasoning with contradictory information [7, 85, 67, 10]. We present here some very simple reductions from inconsistent KBs or databases to abstract argumentation frameworks, which will also serve as a basis when we discuss the relationships between more involved notions of preferred repair and abstract argumentation in Section 5.

Background on abstract argumentation. We start by introducing the basics of abstract argumentation frameworks [76] and their extension with collective attacks [81].

▶ **Definition 19** (Argumentation framework). *An* argumentation framework (AF) *is a pair $(\mathit{Args}, \rightsquigarrow)$ where Args is a finite set of arguments and $\rightsquigarrow \subseteq \mathit{Args} \times \mathit{Args}$ is the attack relation. When $(\alpha, \beta) \in \rightsquigarrow$, we say that α* attacks *β, alternatively denoted by $\alpha \rightsquigarrow \beta$.*

▶ **Definition 20** (Set-based argumentation framework). *A* set-based AF (SETAF) *is a pair $(\mathit{Args}, \rightsquigarrow)$ where Args is a finite set of arguments and $\rightsquigarrow \subseteq (2^{\mathit{Args}} \setminus \{\emptyset\}) \times \mathit{Args}$ is the attack relation. We say that S* attacks *α, and write $S \rightsquigarrow \alpha$, to mean $(S, \alpha) \in \rightsquigarrow$.*

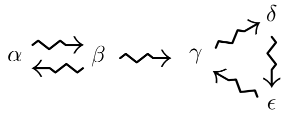
Given an AF $(\mathit{Args}, \rightsquigarrow)$ and some $A \subseteq \mathit{Args}$, let $A^+ = \{\beta \mid \alpha \rightsquigarrow \beta \text{ for some } \alpha \in A\}$ be the set of arguments attacked by arguments from A. A set $A \subseteq \mathit{Args}$ *defends* $\gamma \in \mathit{Args}$ iff $\{\beta \mid \beta \rightsquigarrow \gamma\} \subseteq A^+$. If $(\mathit{Args}, \rightsquigarrow)$ is a SETAF, these definitions are adapted as expected: $A^+ = \{\beta \mid S \rightsquigarrow \beta \text{ for some } S \subseteq A\}$ and A *defends* β iff $A^+ \cap S \neq \emptyset$ whenever $S \rightsquigarrow \beta$.

Given a (SET)AF $F = (\mathit{Args}, \rightsquigarrow)$, a set $A \subseteq \mathit{Args}$ is *conflict-free* if $A \cap A^+ = \emptyset$. The *characteristic function* $\Gamma_F : 2^{\mathit{Args}} \mapsto 2^{\mathit{Args}}$ of F is defined as follows: $\Gamma_F(A) = \{\alpha \mid \alpha \text{ is defended by } A\}$. A set $A \subseteq \mathit{Args}$ is *admissible* if it is conflict-free and defends itself: $A \subseteq \Gamma_F(A)$. (SET)AF semantics are based upon sets of arguments called *extensions*.

▶ **Definition 21** (Extensions). *Let $F = (Args, \rightsquigarrow)$ be a (SET)AF. Then $E \subseteq Args$ is a:*
- naive extension *iff E is a $\subseteq$-maximal conflict-free subset of Args;*
- preferred extension *iff E is a $\subseteq$-maximal admissible set;*
- stable extension *iff $E^+ = Args \setminus E$.*

Stable extensions are preferred extensions but the converse does not hold in general [76].

▶ **Example 22.** The AF depicted below has the following extensions.



Naive extensions: $\{\alpha, \gamma\}$, $\{\alpha, \delta\}$, $\{\alpha, \epsilon\}$, $\{\beta, \delta\}$, $\{\beta, \epsilon\}$
Preferred extensions: $\{\alpha\}$, $\{\beta, \delta\}$
Stable extension: $\{\beta, \delta\}$

From KB and database repairs to (SET)AF extensions. Let $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ be a KB or a database with anti-monotone ICs and assume that $\mathcal{D}$ *does not contain any self-conflicting fact* (i.e., there is no fact $\alpha \in \mathcal{D}$ such that $\langle \{\alpha\}, \mathcal{T} \rangle \models \bot$). The SETAF associated to $\mathcal{K}$ is $F_\mathcal{K} = (\mathcal{D}, \rightsquigarrow_\mathcal{K})$, where $\rightsquigarrow_\mathcal{K} = \{(\mathcal{C} \setminus \{\alpha\}, \alpha) \mid \mathcal{C} \in \mathit{Conf}(\mathcal{K}), \alpha \in \mathcal{C}\}$ [30]. This SETAF is *strongly symmetric*, which implies that preferred and stable extensions coincide [30].

▶ **Proposition 23** ([30]). *Let $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ be a KB or a database with anti-monotone ICs without self-conflicting facts, $F_\mathcal{K} = (\mathcal{D}, \rightsquigarrow_\mathcal{K})$ be its associated SETAF and $\mathcal{R} \subseteq \mathcal{D}$.*
- *$\mathcal{R} \in S\text{-}Rep(\mathcal{K})$ iff $\mathcal{R}$ is a naive extension of $F_\mathcal{K}$.*
- *$\mathcal{R} \in S\text{-}Rep(\mathcal{K})$ iff $\mathcal{R}$ is a preferred extension of $F_\mathcal{K}$.*
- *$\mathcal{R} \in S\text{-}Rep(\mathcal{K})$ iff $\mathcal{R}$ is a stable extension of $F_\mathcal{K}$.*

▶ **Example 24.** Consider $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ defined in Example 15. The SETAF associated to $\mathcal{K}$ is $F_\mathcal{K} = (\{A(a), B(a), C(a), D(a), E(a)\}, \rightsquigarrow_\mathcal{K})$ where $\rightsquigarrow_\mathcal{K}$ is as follows.

$$\begin{array}{rrr}
\{A(a), B(a)\} \rightsquigarrow_\mathcal{K} C(a) & \{A(a), C(a)\} \rightsquigarrow_\mathcal{K} B(a) & \{B(a), C(a)\} \rightsquigarrow_\mathcal{K} A(a) \\
\{A(a)\} \rightsquigarrow_\mathcal{K} D(a) & \{D(a)\} \rightsquigarrow_\mathcal{K} A(a) & \\
\{C(a)\} \rightsquigarrow_\mathcal{K} E(a) & \{E(a)\} \rightsquigarrow_\mathcal{K} C(a) & \\
\{D(a)\} \rightsquigarrow_\mathcal{K} E(a) & \{E(a)\} \rightsquigarrow_\mathcal{K} D(a) &
\end{array}$$

Since this reduction relies entirely on the relationship between repairs and conflicts, it is not difficult to extend it to handle the case of Δ-repairs of database with UCs, using literals instead of facts as the set of arguments. We did however not define conflicts for a database whose ICs have *existential quantifiers in the head* so we cannot easily extend the preceding reduction to such databases. However, there is also a connection between $\subseteq$-*repairs* of databases with inclusion dependencies (of the form $P(\vec{x}) \rightarrow \exists \vec{y} Q[\vec{x}, \vec{y}]$) and functional dependencies (of the form $P(\vec{x}) \wedge P(\vec{y}) \wedge \bigwedge_{i \in I} x_i = y_i \rightarrow \bigwedge_{j \in J} x_j = y_j$) and AF extensions [113]. Let $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ be a database such that $\mathcal{T}$ is a set of IDs and FDs. The AF associated to $\mathcal{K}$ is $F_\mathcal{K} = (Args_\mathcal{K}, \rightsquigarrow_\mathcal{K})$, where $Args_\mathcal{K} = \mathcal{D} \cup \{\alpha_\tau \mid \alpha \in \mathcal{D}, \tau \in \mathcal{T}, \alpha = P(\vec{c}), \tau = P(\vec{x}) \rightarrow \exists \vec{y} Q[\vec{x}, \vec{y}]\}$ and

$$\begin{aligned}
\rightsquigarrow_\mathcal{K} = \{ & (\alpha, \beta), (\beta, \alpha) \mid \exists \tau \in \mathcal{T}, \tau \text{ FD }, \langle \{\alpha, \beta\}, \{\tau\} \rangle \models \bot\} \cup \\
& \{(\alpha_\tau, \alpha), (\alpha_\tau, \alpha_\tau) \mid \alpha_\tau \in Args_\mathcal{K}\} \cup \\
& \{(\beta, \alpha_\tau) \mid \alpha_\tau \in Args_\mathcal{K}, \alpha = P(\vec{c}), \tau = P(\vec{x}) \rightarrow \exists \vec{y} Q[\vec{x}, \vec{y}], \beta \in \mathcal{D}, \beta = Q[\vec{c}, \vec{d}]\}.
\end{aligned}$$

The first line corresponds to the attack relation of the (SET)AF associated to $\langle \mathcal{D}, \mathcal{T}_{\mathsf{FD}} \rangle$, where $\mathcal{T}_{\mathsf{FD}}$ is the set of FDs in $\mathcal{T}$, as defined before, while the second and third lines deal with the IDs: each α_τ attacks $\alpha \in \mathcal{D}$ that requires the presence of some fact to satisfy τ and is attacked by such facts β that are in $\mathcal{D}$.

▶ **Proposition 25** ([113]). *Let $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ with $\mathcal{T}$ a set of IDs and FDs, $F_\mathcal{K} = (Args_\mathcal{K}, \rightsquigarrow_\mathcal{K})$ be its associated AF and $\mathcal{R} \subseteq \mathcal{D}$. Then $\mathcal{R} \in S\text{-}Rep_\subseteq(\mathcal{K})$ iff $\mathcal{R}$ is a preferred extension of $F_\mathcal{K}$.*

Naive, stable and preferred extensions may not coincide in this case, as illustrated below.

▶ **Example 26.** Let $\mathcal{D} = \{\alpha, \beta, \gamma, \delta\}$ with $\alpha = P(c)$, $\beta = P(d)$, $\gamma = Q(d, e)$, and $\delta = Q(d, f)$ and $\mathcal{T}$ consists of the ID $\tau = P(x) \rightarrow \exists y Q(x, y)$ and the FD $Q(x, y) \wedge Q(x, z) \rightarrow y = z$. The AF associated with $\mathcal{K}$ is such that $Args_\mathcal{K} = \{\alpha, \beta, \gamma, \delta, \alpha_\tau, \beta_\tau\}$, and $\rightsquigarrow_\mathcal{K}$ is as follows.

$$\gamma \rightsquigarrow_\mathcal{K} \delta \quad \delta \rightsquigarrow_\mathcal{K} \gamma \quad \alpha_\tau \rightsquigarrow_\mathcal{K} \alpha \quad \alpha_\tau \rightsquigarrow_\mathcal{K} \alpha_\tau \quad \beta_\tau \rightsquigarrow_\mathcal{K} \beta \quad \beta_\tau \rightsquigarrow_\mathcal{K} \beta_\tau \quad \gamma \rightsquigarrow_\mathcal{K} \beta_\tau \quad \delta \rightsquigarrow_\mathcal{K} \beta_\tau$$

The naive extensions of $F_\mathcal{K} = (Args_\mathcal{K}, \rightsquigarrow_\mathcal{K})$ are $\{\alpha, \beta, \gamma\}$ and $\{\alpha, \beta, \delta\}$. Its preferred extensions are $\{\beta, \gamma\}$ and $\{\beta, \delta\}$ (since α is not defended against α_τ), which are also the two $\subseteq$-repairs of $\mathcal{K}$. However, there is no stable extension (since for every $E \subseteq Args_\mathcal{K}$, either $\alpha_\tau \in E$ so that $\alpha_\tau \in E^+$ while $\alpha_\tau \notin Args_\mathcal{K} \setminus E$, or $\alpha_\tau \notin E$ so that $\alpha_\tau \notin E^+$ while $\alpha_\tau \in Args_\mathcal{K} \setminus E$).

From AF extensions to database repairs. Interestingly, it was recently shown that in the other direction, every AF can be encoded as an inconsistent database (using IDs and FDs) whose $\subseteq$-repairs correspond to the preferred extensions of the AF [112].

4 Inconsistency-tolerant semantics

In this section, we briefly review the inconsistency-tolerant semantics that have been proposed in the literature, with a focus on those based on dataset repairs. For more details and examples, we refer to previous chapter and survey on inconsistency handling in DL KBs [29, 28].

4.1 Repair-based semantics

In the following definitions we use the notation $S\text{-}Rep(\mathcal{K})$ even when $\mathcal{K}$ is a database with general ICs, meaning that we can choose between $S\text{-}Rep_\Delta(\mathcal{K})$, $S\text{-}Rep_\subseteq(\mathcal{K})$ or $S\text{-}Rep_\supseteq(\mathcal{K})$ when relevant to get three variants of each semantics. The relationships between the semantics and their properties are summarized in Figure 2, and Example 36 illustrates them.

Consistent query answering (CQA), a.k.a. AR semantics. The most studied and well-established inconsistency-tolerant semantics, called *consistent query answering (CQA)* in the database community [8, 25] and *AR (ABox Repair)* semantics in the DL community [98, 99], considers that a query answer holds if it holds in every repair. This mode of reasoning is also known as *cautious reasoning*, e.g., in the area of abstract argumentation.

▶ **Definition 27** (CQA semantics). *A tuple $\vec{a}$ is an answer to the query $q(\vec{x})$ over $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ under the* CQA semantics *iff $\langle \mathcal{R}, \mathcal{T} \rangle \models q(\vec{a})$ for every $\mathcal{R} \in S\text{-}Rep(\mathcal{K})$.*

Under-approximations of CQA. Several semantics that under-approximate CQA, in the sense that they provide a subset of the consistent query answers, have been considered, either with the goal of achieving a lower computational complexity of reasoning or to obtain answers that are even more "sure" than the consistent query answers. The first one, introduced under the name of *IAR (Intersection of ABox Repairs)* semantics [98, 99] and sometimes called *intersection* semantics, evaluates the queries over the intersection of the repairs.

▶ **Definition 28** (Intersection semantics). *A tuple $\vec{a}$ is an answer to the query $q(\vec{x})$ over $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ under the* intersection semantics *iff $\langle \mathcal{R}_\cap, \mathcal{T} \rangle \models q(\vec{a})$ where $\mathcal{R}_\cap = \bigcap_{\mathcal{R} \in S\text{-}Rep(\mathcal{K})} \mathcal{R}$.*

The *ICR (Intersection of Closed Repairs)* semantics achieves a finer approximation of CQA than the intersection semantics by logically closing repairs with respect to the ontology before intersecting them [27]. Given a $\mathcal{T}$-consistent dataset $\mathcal{B}$, the closure of $\mathcal{B}$ w.r.t. $\mathcal{T}$, denoted $\mathsf{cl}_{\mathcal{T}}(\mathcal{B})$, is the set of all facts entailed by $\langle \mathcal{B}, \mathcal{T} \rangle$.

▶ **Definition 29** (ICR semantics). *A tuple $\vec{a}$ is an answer to the query $q(\vec{x})$ over $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ under the* ICR semantics *iff $\langle \mathcal{R}^{cl}_{\cap}, \mathcal{T} \rangle \models q(\vec{a})$ where $\mathcal{R}^{cl}_{\cap} = \bigcap_{\mathcal{R} \in S\text{-}Rep(\mathcal{K})} \mathsf{cl}_{\mathcal{T}}(\mathcal{R})$.*

▶ Remark 30. If $\mathcal{K}$ is a database, $\mathsf{cl}_{\mathcal{T}}(\mathcal{B}) = \mathcal{B}$ so the ICR and intersection semantics coincide.

The family of *k-support* semantics was introduced to generalize the intersection semantics and obtain more fine-grained under-approximations of CQA, such that increasing the value of k yields a closer approximation [41]. A *support* of a Boolean query q in $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ is a $\mathcal{T}$-consistent subset $\mathcal{S} \subseteq \mathcal{D}$ such that $\langle \mathcal{S}, \mathcal{T} \rangle \models q$.

▶ **Definition 31** (k-support semantics). *A tuple $\vec{a}$ is an answer to the query $q(\vec{x})$ over $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ under the* k-support semantics *iff there exist k (not necessarily distinct) supports $\mathcal{S}_1, \dots, \mathcal{S}_k$ of $q(\vec{a})$ in $\mathcal{K}$ such that for every $\mathcal{R} \in S\text{-}Rep(\mathcal{K})$, there is some $\mathcal{S}_i \subseteq \mathcal{R}$.*

Over-approximations of CQA. Instead of approximating CQA from below, we can over-approximate it, by computing a super-set of the consistent query answers. The *brave* semantics considers all answers that hold in some repair [41].

▶ **Definition 32** (Brave semantics). *A tuple $\vec{a}$ is an answer to the query $q(\vec{x})$ over $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ under the* brave semantics *iff $\langle \mathcal{R}, \mathcal{T} \rangle \models q(\vec{a})$ for some $\mathcal{R} \in S\text{-}Rep(\mathcal{K})$.*

The *non-objection* semantics refines the brave semantics by retaining only the brave query answers that are consistent with all repairs [23].

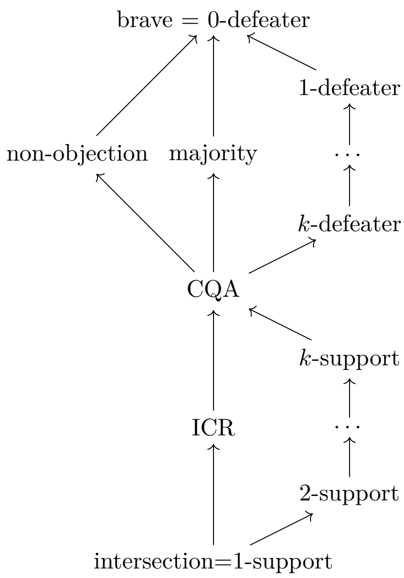
▶ **Definition 33** (Non-objection semantics). *A tuple $\vec{a}$ is an answer to the query $q(\vec{x})$ over $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ under the* non-objection semantics *iff (i) $\langle \mathcal{R}, \mathcal{T} \rangle \models q(\vec{a})$ for some $\mathcal{R} \in S\text{-}Rep(\mathcal{K})$ and (ii) for every $\mathcal{R}' \in S\text{-}Rep(\mathcal{K})$, there is a model $\mathcal{I}$ of $\langle \mathcal{R}', \mathcal{T} \rangle$ such that $\mathcal{I} \models q(\vec{a})$.*

The *majority-based* semantics is another refinement of brave that requires that the answer holds in more than half of the repairs [19]. This semantics can be adapted to use any threshold instead of $\frac{1}{2}$ and is closely related to the semantics that associates to each brave query answer the percentage (or number) of repairs in which it holds, which attracted quite a lot of attention in the database setting, with theoretical results on the precise complexity of the counting problem depending on the classes of constraints and queries [118, 48, 52] as well as approximation schemes [49].

▶ **Definition 34** (Majority-based semantics). *A tuple $\vec{a}$ is an answer to the query $q(\vec{x})$ over $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ under the* majority-based semantics *iff $\frac{|\mathcal{R} \mid \mathcal{R} \in S\text{-}Rep(\mathcal{K}), \langle \mathcal{R}, \mathcal{T} \rangle \models q(\vec{a})|}{|S\text{-}Rep(\mathcal{K})|} > \frac{1}{2}$.*

Finally, the family of *k-defeater* semantics provides fine-grained over-approximations of CQA, where increasing the value of k yields a closer approximation [41] (note that even if the following definition does not mention a repair, the set $\mathcal{B}$ is a subset of a repair that does not entail the query, so we consider this semantics as "repair-based").

▶ **Definition 35** (k-defeater semantics). *A tuple $\vec{a}$ is an answer to the query $q(\vec{x})$ over $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ under the* k-defeater semantics *iff there does not exist a $\mathcal{T}$-consistent subset $\mathcal{B}$ of $\mathcal{D}$ with $|\mathcal{B}| \leq k$ such that $\langle \mathcal{B} \cup \mathcal{S}, \mathcal{T} \rangle \models \bot$ for every minimal support $\mathcal{S}$ of $q(\vec{a})$ in $\mathcal{K}$.*



	Semantics with the property
CONSISTENT SUPPORT	all
CONSISTENT RESULTS	intersection, ICR, k-support, CQA, non-objection
UNIQUE BASE	intersection, ICR

- CONSISTENT SUPPORT: for every $\langle \mathcal{D}, \mathcal{T} \rangle$, $q(\vec{x})$ and $\vec{a}$, if $\vec{a}$ is an answer to $q(\vec{x})$ over $\langle \mathcal{D}, \mathcal{T} \rangle$ under Sem, then there exists a $\mathcal{T}$-consistent subset $\mathcal{S}$ of $\mathcal{D}$ such that $\langle \mathcal{S}, \mathcal{T} \rangle \models q(\vec{a})$.
- CONSISTENT RESULTS: for every $\langle \mathcal{D}, \mathcal{T} \rangle$, there exists a model $\mathcal{I}$ of $\mathcal{T}$ such that for every $q(\vec{x})$ and $\vec{a}$, if $\vec{a}$ is an answer to $q(\vec{x})$ over $\langle \mathcal{D}, \mathcal{T} \rangle$ under Sem, then $\mathcal{I} \models q(\vec{a})$.
- UNIQUE BASE: for every $\langle \mathcal{D}, \mathcal{T} \rangle$, there exists a $\mathcal{T}$-consistent dataset $\mathcal{D}'$ such that for every $q(\vec{x})$ and $\vec{a}$, $\vec{a}$ is an answer to $q(\vec{x})$ over $\langle \mathcal{D}, \mathcal{T} \rangle$ under Sem iff $\langle \mathcal{D}', \mathcal{T} \rangle \models q(\vec{a})$.

Figure 2 (left) Relationships between repair-based semantics (adapted from [28, Fig. 1]). An arrow Sem $\rightarrow$ Sem$'$ means that if $\vec{a}$ is an answer to $q(\vec{x})$ over $\mathcal{K}$ under Sem, then it is also an answer under Sem$'$. (right) Properties of repair-based semantics (adapted from [28, Fig. 2]).

Properties and relationships between repair-based semantics. Figure 2 (left) shows the relationships between the repair-based inconsistency-tolerant semantics. Following [29, 28], we also recall in Figure 2 (right) which desirable properties these semantics have (note that all repair-based semantics have the CONSISTENT SUPPORT property, contrary to some non repair-based inconsistency-tolerant semantics, cf. Section 4.2). We conclude with an example to illustrate the differences between these semantics.

▶ **Example 36.** Consider the following DL-Lite$_{\mathsf{core}}$ KB. The ontology expresses that associate and full professors are professors, professors and postdoctoral fellows are PhD holders and belong to some department, postdoctoral fellows have some supervisor, that one cannot be both an associate and a full professor or a postdoctoral fellow and a professor, and that the range of the roles Chair and Teach are disjoint.

$$\begin{aligned}\mathcal{T} = \{&\mathsf{AssociateProf} \sqsubseteq \mathsf{Prof},\ \mathsf{FullProf} \sqsubseteq \mathsf{Prof},\ \mathsf{Prof} \sqsubseteq \mathsf{Phd},\ \mathsf{Postdoc} \sqsubseteq \mathsf{Phd},\\ &\mathsf{Prof} \sqsubseteq \exists \mathsf{Department},\ \mathsf{Postoc} \sqsubseteq \exists \mathsf{Department},\ \mathsf{Postdoc} \sqsubseteq \exists \mathsf{Supervisor},\\ &\mathsf{AssociateProf} \sqsubseteq \neg \mathsf{FullProf},\ \mathsf{Postdoc} \sqsubseteq \neg \mathsf{Prof},\ \exists \mathsf{Chair}^- \sqsubseteq \neg \exists \mathsf{Teach}^- \}\\ \mathcal{D} = \{&\mathsf{AssociateProf}(\mathit{Carol}), \mathsf{FullProf}(\mathit{Carol}), \mathsf{Postdoc}(\mathit{Carol}),\\ &\mathsf{Teach}(\mathit{Carol}, \mathit{Algo}), \mathsf{Teach}(\mathit{Carol}, \mathit{DB}), \mathsf{Teach}(\mathit{Carol}, \mathit{KR}),\\ &\mathsf{Chair}(\mathit{Carol}, \mathit{DB}), \mathsf{Chair}(\mathit{Carol}, \mathit{KR})\}\end{aligned}$$

The repairs of $\langle \mathcal{D}, \mathcal{T} \rangle$ are as follows:

$$\begin{aligned}\mathcal{R}_1 &= \{\mathsf{AssociateProf}(\mathit{Carol}), \mathsf{Teach}(\mathit{Carol}, \mathit{Algo}), \mathsf{Teach}(\mathit{Carol}, \mathit{DB}), \mathsf{Teach}(\mathit{Carol}, \mathit{KR})\}\\ \mathcal{R}_2 &= \{\mathsf{AssociateProf}(\mathit{Carol}), \mathsf{Teach}(\mathit{Carol}, \mathit{Algo}), \mathsf{Chair}(\mathit{Carol}, \mathit{DB}), \mathsf{Chair}(\mathit{Carol}, \mathit{KR})\}\\ \mathcal{R}_3 &= \{\mathsf{AssociateProf}(\mathit{Carol}), \mathsf{Teach}(\mathit{Carol}, \mathit{Algo}), \mathsf{Teach}(\mathit{Carol}, \mathit{DB}), \mathsf{Chair}(\mathit{Carol}, \mathit{KR})\}\end{aligned}$$

$$\mathcal{R}_4 = \{\mathsf{AssociateProf}(Carol), \mathsf{Teach}(Carol, Algo), \mathsf{Teach}(Carol, KR), \mathsf{Chair}(Carol, DB)\}$$
$$\mathcal{R}_5 = \{\mathsf{FullProf}(Carol), \mathsf{Teach}(Carol, Algo), \mathsf{Teach}(Carol, DB), \mathsf{Teach}(Carol, KR)\}$$
$$\mathcal{R}_6 = \{\mathsf{FullProf}(Carol), \mathsf{Teach}(Carol, Algo), \mathsf{Chair}(Carol, DB), \mathsf{Chair}(Carol, KR)\}$$
$$\mathcal{R}_7 = \{\mathsf{FullProf}(Carol), \mathsf{Teach}(Carol, Algo), \mathsf{Teach}(Carol, DB), \mathsf{Chair}(Carol, KR)\}$$
$$\mathcal{R}_8 = \{\mathsf{FullProf}(Carol), \mathsf{Teach}(Carol, Algo), \mathsf{Teach}(Carol, KR), \mathsf{Chair}(Carol, DB)\}$$
$$\mathcal{R}_9 = \{\mathsf{Postdoc}(Carol), \mathsf{Teach}(Carol, Algo), \mathsf{Teach}(Carol, DB), \mathsf{Teach}(Carol, KR)\}$$
$$\mathcal{R}_{10} = \{\mathsf{Postdoc}(Carol), \mathsf{Teach}(Carol, Algo), \mathsf{Chair}(Carol, DB), \mathsf{Chair}(Carol, KR)\}$$
$$\mathcal{R}_{11} = \{\mathsf{Postdoc}(Carol), \mathsf{Teach}(Carol, Algo), \mathsf{Teach}(Carol, DB), \mathsf{Chair}(Carol, KR)\}$$
$$\mathcal{R}_{12} = \{\mathsf{Postdoc}(Carol), \mathsf{Teach}(Carol, Algo), \mathsf{Teach}(Carol, KR), \mathsf{Chair}(Carol, DB)\}$$

The KB $\langle \mathcal{D}, \mathcal{T} \rangle$ entails the BCQ:

- $\mathsf{Teach}(Carol, Algo)$ under the intersection semantics, since $\bigcap_{i=1}^{12} \mathcal{R}_i = \{\mathsf{Teach}(Carol, Algo)\}$;
- $\mathsf{Phd}(Carol)$ under
  - the ICR semantics, since $\bigcap_{i=1}^{12} \mathsf{cl}_{\mathcal{T}}(\mathcal{R}_i) = \{\mathsf{Teach}(Carol, Algo), \mathsf{Phd}(Carol)\}$,
  - the 3-support semantics, since the three supports $\{\mathsf{AssociateProf}(Carol)\}$, $\{\mathsf{FullProf}(Carol)\}$ and $\{\mathsf{Postdoc}(Carol)\}$ cover all repairs,

  but not under the intersection semantics nor under the 2-support semantics;
- $\exists y \mathsf{Department}(Carol, y)$ under the CQA semantics, since it is entailed by every repair, but not under the ICR semantics;
- $\mathsf{Prof}(Carol)$ under the majority semantics (it holds in $\frac{2}{3}$ of the repairs), but not under
  - the CQA semantics, since it does not hold in $\mathcal{R}_9 - \mathcal{R}_{12}$,
  - the non-objection semantics, since for $i \in \{9, 12\}$, $\langle \mathcal{R}_i \cup \{\mathsf{Prof}(Carol)\}, \mathcal{T} \rangle \models \bot$,
  - the 1-defeater semantics, since one fact ($\mathsf{Postdoc}(Carol)$) is sufficient to contradict the two minimal supports of $\mathsf{Prof}(Carol)$ ($\{\mathsf{AssociateProf}(Carol)\}$ and $\{\mathsf{FullProf}(Carol)\}$);
- $\exists y \mathsf{Supervisor}(Carol, y)$ under the non-objection semantics since it holds in repairs $\mathcal{R}_9 - \mathcal{R}_{12}$ and is consistent with all repairs, but not under
  - the CQA semantics, since it does not hold in $\mathcal{R}_1 - \mathcal{R}_8$,
  - the majority semantics, since it holds only in $\frac{1}{3}$ of the repairs,
  - the 1-defeater semantics, since one fact (e.g., $\mathsf{FullProf}(Carol)$) is sufficient to contradict the only minimal support of $\exists y \mathsf{Supervisor}(Carol, y)$ ($\{\mathsf{Postdoc}(Carol)\}$);
- $\exists y \mathsf{Chair}(Carol, y)$ under the 1-defeater semantics, since we need two facts ($\mathsf{Teach}(Carol, DB)$ and $\mathsf{Teach}(Carol, KR)$) to contradict the minimal supports $\{\mathsf{Chair}(Carol, DB)\}$ and $\{\mathsf{Chair}(Carol, KR)\}$, but not under the CQA semantics (cf. $\mathcal{R}_1$, $\mathcal{R}_5$ and $\mathcal{R}_9$);
- $\mathsf{AssociateProf}(Carol)$ under the brave semantics but not under the non-objection, majority or 1-defeater semantics.

4.2 Related work: Semantics that are not based on dataset repairs

Not all inconsistency-tolerant semantics that have been proposed for KBs or databases are based on dataset repairs. We review them only briefly since they are not the main focus of this chapter and received comparatively less attention than the main repair-based semantics.

$\mathcal{T}$-consistent subsets of some closure of $\mathcal{D}$. In the KB context, several semantics based on repairs of some closure of the original dataset w.r.t. the ontology were proposed, with the motivation of retaining more consequences and being independent from the KB syntax. The first ones, called the *CAR (Closed ABox Repair)* and *ICAR (Intersection of Closed ABox Repairs)* semantics, are the counterparts of the AR (CQA) and IAR (intersection) semantics that use closed repairs defined below instead of dataset repairs [98]. The *consistent closure* of $\mathcal{D}$ w.r.t. $\mathcal{T}$, denoted $\mathsf{conscl}_{\mathcal{T}}(\mathcal{D})$, is the set of facts entailed by $\mathcal{T}$-consistent subsets of $\mathcal{D}$.

▶ **Definition 37** (Closed repairs). *A* closed repair *of $\langle\mathcal{D},\mathcal{T}\rangle$ is a $\mathcal{T}$-consistent $\mathcal{R}\subseteq\mathsf{conscl}_{\mathcal{T}}(\mathcal{D})$ for which there is no $\mathcal{T}$-consistent $\mathcal{R}'\subseteq\mathsf{conscl}_{\mathcal{T}}(\mathcal{D})$ such that either (i) $\mathcal{R}\cap\mathcal{D}\subsetneq\mathcal{R}'\cap\mathcal{D}$ or (ii) $\mathcal{R}\cap\mathcal{D}=\mathcal{R}'\cap\mathcal{D}$ and $\mathcal{R}\subsetneq\mathcal{R}'$.*

The closed repairs of $\langle\mathcal{D},\mathcal{T}\rangle$ may differ from the (dataset) repairs of $\mathsf{conscl}_{\mathcal{T}}(\mathcal{D})$ (see [29, Example 13] for an example) so we get different semantics by using the latter. Another notion of closure of the original dataset was proposed in the context of Datalog$^\pm$ KBs: the *positive closure* of $\mathcal{D}$ w.r.t. $\mathcal{T}$, denoted $\mathsf{poscl}_{\mathcal{T}}(\mathcal{D})$, is the set of facts entailed by $\langle\mathcal{D},\mathcal{T}^+\rangle$ where $\mathcal{T}^+$ is the set of TGDs in $\mathcal{T}$ [19]. This yields yet another family of semantics based on the repairs of $\mathsf{poscl}_{\mathcal{T}}(\mathcal{D})$. However, it is not clear how to define positive closure for expressive DLs such as $\mathcal{ALC}$ and $\mathcal{SHIQ}$, since they cannot be written as Datalog$^\pm$ rules.

Note that in contrast with the repair-based semantics, these semantics do not satisfy the Consistent Support property (cf. [29, Example 15]).

$\mathcal{T}$-consistent quantified datasets. A long line of work in DL is concerned by modifying a (consistent) KB so that some unwanted consequences do not hold anymore, while preserving as much as possible the other consequences. Some recent work targets in particular dataset modifications, using *quantified ABoxes*, which allow variables to be used in facts [16, 15]. By identifying sets of atoms with their conjunctions, we can view such a quantified ABox $\mathcal{D}$ as a BCQ $\exists\vec{y}\mathcal{D}$ where $\exists\vec{y}$ contains all variables in $\mathcal{D}$. The extension of this framework to *inconsistent* KBs gives rise to an alternative notion of syntax-independent, consequence-preserving, repairs [17, 18]. These repairs have been defined for *Horn-DL ontologies*, which are such that $\mathcal{T}$ can be split into a "positive part" $\mathcal{T}^+$, which is such that any dataset is $\mathcal{T}^+$-consistent, and a "negative part" $\mathcal{T}^\perp$ which contains only concept inclusions of the form $C\sqsubseteq\perp$. Intuitively, a repair of a dataset $\mathcal{D}$ w.r.t. an ontology $\mathcal{T}$ is a quantified ABox $\exists\vec{y}\mathcal{R}$ such that $\langle\mathcal{D},\mathcal{T}^+\rangle\models\exists\vec{y}\mathcal{R}$, $\langle\exists\vec{y}\mathcal{R},\mathcal{T}^+\rangle\not\models\exists xC(x)$ for every $C\sqsubseteq\perp\in\mathcal{T}^\perp$ (and $\langle\exists\vec{y}\mathcal{R},\mathcal{T}^+\rangle\not\models q$ for other unwanted consequences q, if any). An optimal consequence-preserving repair is then a repair $\exists\vec{y}\mathcal{R}$ such that there is no repair $\exists\vec{z}\mathcal{R}'$ such that $\langle\exists\vec{z}\mathcal{R}',\mathcal{T}^+\rangle\models\exists\vec{y}\mathcal{R}$ but $\langle\exists\vec{y}\mathcal{R},\mathcal{T}^+\rangle\not\models\exists\vec{z}\mathcal{R}'$. We omit the formal definitions and details on this framework and simply illustrate how it can be applied in our context.

▶ **Example 38.** Let $\langle\mathcal{D},\mathcal{T}\rangle$ be as follows. $\mathcal{T}^+$ contains the two first axioms, $\mathcal{T}^\perp$ the last one.

$$\mathcal{T}=\{\exists\mathsf{Teach}\sqsubseteq\mathsf{Person},\ \exists\mathsf{Teach}^-\sqsubseteq\mathsf{Course},\ \mathsf{Person}\sqcap\mathsf{Course}\sqsubseteq\perp\}$$
$$\mathcal{D}=\{\mathsf{Teach}(\mathit{Dan},\mathit{DB}),\mathsf{Teach}(\mathit{Dan},\mathit{KR}),\mathsf{Teach}(\mathit{Algo},\mathit{Dan})\}$$

It holds that the quantified ABoxes $\exists x\,\{\mathsf{Teach}(\mathit{Dan},\mathit{DB}),\mathsf{Teach}(\mathit{Dan},\mathit{KR}),\mathsf{Teach}(\mathit{Algo},x)\}$ and $\exists x\,\{\mathsf{Teach}(x,\mathit{DB}),\mathsf{Teach}(x,\mathit{KR}),\mathsf{Teach}(\mathit{Algo},\mathit{Dan})\}$ are optimal consequence-preserving repairs of $\mathcal{D}$ w.r.t. $\mathcal{T}$, as well as all those obtained by adding some consequences w.r.t. $\mathcal{T}^+$, such as $\exists x\,\{\mathsf{Teach}(\mathit{Dan},\mathit{DB}),\mathsf{Teach}(\mathit{Dan},\mathit{KR}),\mathsf{Teach}(\mathit{Algo},x),\mathsf{Person}(\mathit{Dan}),\mathsf{Course}(\mathit{DB})\}$. Indeed, two equivalent (quantified) datasets are both optimal repairs or not. Using such repairs allows us to keep the information that *KR* and *DB* courses are taught by the same person, so that $\exists x \mathsf{Teach}(x, DB) \wedge \mathsf{Teach}(x, KR)$ is entailed under the CQA semantics based on this kind of repair. We also obtain that $\exists xy \mathsf{Teach}(x, KR) \wedge \mathsf{Teach}(Algo, y)$ holds under CQA based on such repairs, which shows that the Consistent Support property is not satisfied.

$\mathcal{T}$-consistent subsets of $\mathcal{D}$ that may not be $\subseteq$-maximal. Some semantics relax the notion of repairs by allowing "too many" facts to be removed to restore consistency. Motivated by computational complexity considerations, the family of *k-lazy* semantics restores consistency in each connected component of the conflict hypergraph by either removing at most k facts, or by removing the whole connected component if removing k facts is not sufficient to achieve consistency [109]. To get a database counterpart of conflict-free sets of arguments considered in the context of abstract argumentation, a recent work defines "repairs" as $\mathcal{T}$-consistent subsets of $\mathcal{D}$, calling "maximal repairs" the standard $\subseteq$-repairs [112]. This very relaxed notion of repairs is the basis of the definition of *maximally covering* (resp. *fully covering*) repairs which are $\mathcal{T}$-consistent subsets of $\mathcal{D}$ that retain as much as possible (resp. all) the constants occurring in $\mathcal{D}$ at some predicate positions.

Repairs modifying $\mathcal{T}$. While most inconsistency-tolerant semantics consider that the logical theory is reliable and repair only the dataset, *generalized* repairs allow to modify $\mathcal{T}$ as well as $\mathcal{D}$ to restore consistency [78]. The definition also allows for splitting $\mathcal{T}$ and $\mathcal{D}$ into a hard and a soft part, so that only the soft parts can be modified by a repair.

Attribute-based repairs. In the database area, *attribute-based* repairs, which modify the tuples instead of removing them, have been proposed. Many different kinds of attribute-based repairs have been considered: the attribute values can be taken from a database-independent domain, the database domain, or be "nulls" with a specific semantics, and minimality of change can be measured by a numerical aggregation function over differences of attribute values, or set-minimality of the sequence of updates yielding the repair [43, 102, 88, 26].

Operational repairs. Also in the database area, *operational* repairs have been defined as the result of applying a repairing sequence of update operations that has to fulfill some conditions [51, 53]. This kind of repair also allows for taking into account the probability of the updates, hence of the repairs and query answers.

5 Preferred repairs

In many scenarios, one can define preferred repairs based on some preference information, such as the relative or absolute reliability of the facts, or some rules modeling the user preferences. In this section, we present the different kinds of preferred (dataset) repair that have been proposed in the literature, their properties, relationships and links with other formalisms for inconsistency handling. Preferred repairs have mostly been defined for $\subseteq$-repairs, since it is less natural to consider, e.g., the reliability of some missing facts, but we also consider preferred Δ-repairs when they have been studied. As in Section 4, we conclude with a brief overview of other notions of "optimal repairs" which are not dataset repairs.

Before delving into the specific preferred repair definitions, we start with two general observations on the impact of using preferred (dataset) repairs on repair-based semantics: (1) it increases the number of query answers under CQA and intersection semantics while it decreases the number of answers under brave semantics, and (2) it preserves the relationships between the semantics presented in Figure 2.

▶ **Proposition 39.** *Let X-CQA, X-intersection and X-brave be the counterparts of the CQA, intersection and brave semantics that use preferred repairs from some X-Rep$(\mathcal{K}) \subseteq S$-Rep$(\mathcal{K})$ instead of standard ($\subseteq$- or Δ-)repairs. If q is entailed by $\mathcal{K}$ under:*

- *CQA (resp. intersection), then it is also entailed under X-CQA (resp. X-intersection);*
- *X-brave, then it is also entailed under brave.*

Moreover, if q is entailed by $\mathcal{K}$ under:

- *X-intersection, then it is entailed under X-CQA;*
- *X-CQA, then it is entailed under X-brave.*

5.1 Preferred repairs based on a preorder over datasets ($\leq, \leq_w, \subseteq_P, \leq_P$)

The first category of preferred repairs is based on a generalization of the definition of dataset repairs (cf. Definition 10) using a preorder (i.e., a reflexive and transitive binary relation) $\preceq$ over datasets instead of set-inclusion. We write $\mathcal{B} \prec \mathcal{B}'$ to indicate that $\mathcal{B} \preceq \mathcal{B}'$ and $\mathcal{B}' \not\preceq \mathcal{B}$.

▶ **Definition 40** (Strictly $\subseteq$-monotone preorder). *A preorder $\preceq$ over datasets is* strictly $\subseteq$-monotone *iff $\mathcal{B} \subset \mathcal{B}'$ implies $\mathcal{B} \prec \mathcal{B}'$.*

▶ **Definition 41** ($\preceq$-optimal repairs). *Let $\preceq$ be a strictly $\subseteq$-monotone preorder over datasets.*

- *A $\preceq$-optimal $\subseteq$-repair of $\mathcal{D}$ w.r.t. $\mathcal{T}$ is a $\mathcal{T}$-consistent dataset $\mathcal{R} \subseteq \mathcal{D}$ such that there is no $\mathcal{T}$-consistent $\mathcal{R}' \subseteq \mathcal{D}$ such that $\mathcal{R} \prec \mathcal{R}'$.*
- *A $\preceq$-optimal Δ-repair of $\mathcal{D}$ w.r.t. $\mathcal{T}$ is a $\mathcal{T}$-consistent dataset $\mathcal{R}$ such that there is no $\mathcal{T}$-consistent $\mathcal{R}'$ such that $\mathcal{R}'\Delta\mathcal{D} \prec \mathcal{R}\Delta\mathcal{D}$.*

We write $\preceq$-Rep$(\mathcal{K})$ (resp. $\preceq$-Rep$_{\subseteq}(\mathcal{K})$, $\preceq$-Rep$_{\Delta}(\mathcal{K})$) to denote the set of $\preceq$-optimal (resp. $\subseteq$-, Δ-) repairs of $\mathcal{K}$.

▶ Remark 42. The condition that $\preceq$ is strictly $\subseteq$-monotone implies that $\preceq$-optimal $\subseteq$-repairs are indeed $\subseteq$-repairs, and $\preceq$-optimal Δ-repairs are Δ-repairs (i.e., $\preceq$-*Rep*$_{\subseteq}(\mathcal{K}) \subseteq$ *S-Rep*$_{\subseteq}(\mathcal{K})$ and $\preceq$-*Rep*$_{\Delta}(\mathcal{K}) \subseteq$ *S-Rep*$_{\Delta}(\mathcal{K})$). Hence Proposition 39 applies.

We next introduce the four types of $\preceq$-optimal repairs that received the most attention. Their properties and relationships as well as examples are given at the end of this section.

Cardinality ($\leq$). *Cardinality-based* repairs use the preorder $\leq$ that compares datasets based on their cardinality:

$\mathcal{B} \leq \mathcal{B}'$ iff $|\mathcal{B}| \leq |\mathcal{B}'|$.

The $\leq$-optimal Δ-repairs are those that make the fewest updates. In particular, $\leq$-optimal $\subseteq$-repairs drop the fewest facts, and are thus more appropriate when it is believed that all the facts in the dataset have the same (small) probability of being erroneous. Cardinality-based repairs have been studied for DL and Datalog$^{\pm}$ KBs [33, 108] as well as for both Δ-repairs and $\subseteq$-repairs in the database setting [9, 102, 4, 74].

Weight ($\leq_w$). *Weight-based* repairs require a function w that assigns a (positive) weight to each fact and induces a preorder $\leq_w$ over datasets:

$\mathcal{B} \leq_w \mathcal{B}'$ iff $\Sigma_{\alpha\in\mathcal{B}} w(\alpha) \leq \Sigma_{\alpha\in\mathcal{B}'} w(\alpha)$.

Weights can be used to model the reliability of different facts in $\mathcal{D}$ (e.g., if it is built using information extraction techniques, the weights may be derived from the confidence levels output by the extraction tool). In the case of Δ-repairs, one needs to assign a weight to all possible facts (since $\leq_w$-optimal Δ-repairs minimize the weight of the symmetric difference between $\mathcal{D}$ and the repair). This can be done, e.g., by assigning the same weight to all facts that do not belong to $\mathcal{D}$, to penalize every addition in the same way while taking into account different reliability levels for facts from $\mathcal{D}$. This kind of repair has been studied for DL and Datalog$^{\pm}$ KBs [73, 33, 107] and for databases [102, 100, 122].

Prioritized set inclusion ($\subseteq_P$). Repairs based on *priority levels and set-inclusion* require that the dataset $\mathcal{D}$ is partitioned into priority levels: a *prioritization* of $\mathcal{D}$ is a tuple $P = \langle \mathcal{P}_1, \dots, \mathcal{P}_n \rangle$ of disjoint datasets such that $\mathcal{D} = \bigcup_{i=1}^{n} \mathcal{P}_i$. Facts in $\mathcal{P}_1$ are considered the most reliable and those in $\mathcal{P}_n$ the least reliable. A prioritization $P = \langle \mathcal{P}_1, \dots, \mathcal{P}_n \rangle$ induces a set-inclusion-based preorder $\subseteq_P$:

$$
\begin{aligned}
\mathcal{B} \subseteq_P \mathcal{B}' \text{ iff either } & \mathcal{B} \cap \mathcal{P}_i = \mathcal{B}' \cap \mathcal{P}_i \text{ for every } 1 \leq i \leq n, \\
& \text{or there is some } 1 \leq i \leq n \text{ such that} \\
& \mathcal{B} \cap \mathcal{P}_i \subsetneq \mathcal{B}' \cap \mathcal{P}_i \text{ and } \mathcal{B} \cap \mathcal{P}_j = \mathcal{B}' \cap \mathcal{P}_j \text{ for } 1 \leq j < i.
\end{aligned}
$$

Prioritizations are a natural way to distinguish facts coming from different sources, separate a part of the dataset that has already been validated from recent additions, or to take into account that some predicates are known to be more reliable than others. $\subseteq_P$-optimal repairs have been mostly studied for DL and Datalog$^{\pm}$ KBs [33, 107] but also extended to Δ-repairs of databases with universal constraints [32]. In this case, we consider a prioritization $P = \langle \mathcal{P}_1, \dots, \mathcal{P}_n \rangle$ of the set of relevant literals $\mathit{Lits}^{\mathcal{T}}_{\mathcal{D}}$ (cf. Definition 16) so that we keep the intuition that the more reliable a literal λ (which expresses whether a fact is true or false w.r.t. $\mathcal{D}$), the smaller the index of $\mathcal{P}_i$ that contains λ. We then simply drop the negation in literals to get the corresponding prioritization of all relevant facts $\mathit{Facts}^{\mathcal{T}}_{\mathcal{D}}$, $P' = \langle \mathcal{P}'_1, \dots, \mathcal{P}'_n \rangle$ with $\mathcal{P}'_i = (\mathcal{P}_i \cap \mathcal{D}) \cup \{\alpha \mid \neg\alpha \in \mathcal{P}_i\}$, and use P' instead of P in the definition of $\subseteq_P$ (additionally setting $\mathcal{B} \subsetneq_P \mathcal{B}'$ for every $\mathcal{B} \subseteq \mathit{Facts}^{\mathcal{T}}_{\mathcal{D}}$ and $\mathcal{B}' \not\subseteq \mathit{Facts}^{\mathcal{T}}_{\mathcal{D}}$). We can show that the definition of $\subseteq_P$-optimal Δ-repairs given in [32] is indeed equivalent to that of Definition 41 using this preorder by observing that for all $\mathcal{R}, \mathcal{R}' \subseteq \mathit{Facts}^{\mathcal{T}}_{\mathcal{D}}$, $\mathcal{R}' \Delta \mathcal{D} \subsetneq_P \mathcal{R} \Delta \mathcal{D}$ iff there is some $1 \leq i \leq n$ such that

- $\mathit{Int}_{\mathcal{D}}(\mathcal{R}) \cap \mathcal{P}_i \subsetneq \mathit{Int}_{\mathcal{D}}(\mathcal{R}') \cap \mathcal{P}_i$ (recall from Proposition 17 that $\mathit{Int}_{\mathcal{D}}(\mathcal{B}) = (\mathcal{B} \cap \mathcal{D}) \cup \{\neg\alpha \mid \alpha \in \mathit{Facts}^{\mathcal{T}}_{\mathcal{D}} \setminus (\mathcal{B} \cup \mathcal{D})\}$ is the set of literals upon which $\mathcal{B}$ and $\mathcal{D}$ agree) and
- $\mathit{Int}_{\mathcal{D}}(\mathcal{R}) \cap \mathcal{P}_j = \mathit{Int}_{\mathcal{D}}(\mathcal{R}') \cap \mathcal{P}_j$ for $1 \leq j < i$.

Prioritized cardinality ($\leq_P$). A prioritization $P = \langle \mathcal{P}_1, \dots, \mathcal{P}_n \rangle$ also induces a cardinality-based preorder $\leq_P$:

$$
\begin{aligned}
\mathcal{B} \leq_P \mathcal{B}' \text{ iff either } & |\mathcal{B} \cap \mathcal{P}_i| = |\mathcal{B}' \cap \mathcal{P}_i| \text{ for every } 1 \leq i \leq n, \\
& \text{or there is some } 1 \leq i \leq n \text{ such that} \\
& |\mathcal{B} \cap \mathcal{P}_i| < |\mathcal{B}' \cap \mathcal{P}_i| \text{ and } |\mathcal{B} \cap \mathcal{P}_j| = |\mathcal{B}' \cap \mathcal{P}_j| \text{ for } 1 \leq j < i.
\end{aligned}
$$

$\leq_P$-optimal repairs have been studied for DL and Datalog$^{\pm}$ KBs [33, 107] but we can extend their definition to Δ-repairs of databases with universal constraints in the same way as for $\subseteq_P$-optimal repairs. Note that with $\leq_P$ and $\subseteq_P$, a single fact on level $\mathcal{P}_i$ is preferred to any number of facts from $\mathcal{P}_{i+1}$. This makes these preorders best suited for cases in which there is a significant difference in the perceived reliability of adjacent priority levels.

Properties and relationships. In this paragraph, we use $\preceq$-$\mathit{Rep}(\mathcal{K})$ ($\preceq \in \{\leq, \leq_w, \subseteq_P, \leq_P\}$) and S-$\mathit{Rep}(\mathcal{K})$ to denote the ($\preceq$-optimal) $\subseteq$- or Δ-repairs of a KB or database $\mathcal{K}$.

- If w assigns the same weight to every fact, then $\leq_w$-*Rep*$(\mathcal{K})$ and $\leq$-*Rep*$(\mathcal{K})$ coincide, i.e., weight-based repairs generalize cardinality-based repairs.
- Given a prioritization $P = \langle \mathcal{P}_1, \dots, \mathcal{P}_n \rangle$, let $u = (\max_{i=1}^{n} |\mathcal{P}_i|) + 1$, and let w be defined by $w(\alpha) = u^{n-i}$ for $\alpha \in \mathcal{P}_i$ (or in case of Δ-repairs, $w(\alpha) = u^{n-i}$ for $\alpha \in \mathcal{P}_i$ or $\neg\alpha \in \mathcal{P}_i$). Then for every $\mathcal{T}$, $\leq_P$-*Rep*$(\mathcal{D}, \mathcal{T})$ and $\leq_w$-*Rep*$(\mathcal{D}, \mathcal{T})$ coincide [44, Lemma 6.2.5]. Hence weight-based repairs also generalize repairs based on prioritized cardinality.
- If a prioritization assigns all facts (or literals) to the same priority level, then $\subseteq_P$-*Rep*$(\mathcal{K})$ and *S*-*Rep*$(\mathcal{K})$ coincide, and $\leq_P$-*Rep*$(\mathcal{K})$ and $\leq$-*Rep*$(\mathcal{K})$ coincide.

Examples. Examples 43 and 44 illustrate the four kinds of $\preceq$-optimal repair, for KBs and Δ-repairs of databases with UCs, respectively.

▶ **Example 43** **([44]).** Consider the DL-Lite$_{\mathsf{core}}$ KB $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ defined as follows.

$$\begin{aligned}\mathcal{T} = \{&\mathsf{Student} \sqsubseteq \mathsf{Person}, \ \mathsf{Prof} \sqsubseteq \mathsf{Person}, \ \mathsf{Student} \sqsubseteq \neg\mathsf{Prof},\\ &\exists\mathsf{Teach} \sqsubseteq \mathsf{Prof}, \ \exists\mathsf{Teach}^- \sqsubseteq \mathsf{Course}, \ \mathsf{Course} \sqsubseteq \neg\mathsf{Person}\}\\ \mathcal{D} = \{&\mathsf{Student}(\mathit{Ann}), \mathsf{Teach}(\mathit{Ann}, \mathit{DB}), \mathsf{Teach}(\mathit{Ann}, \mathit{KR}),\\ &\mathsf{Student}(\mathit{Bob}), \mathsf{Prof}(\mathit{Bob}), \mathsf{Teach}(\mathit{KR}, \mathit{Bob}), \mathsf{Teach}(\mathit{Bob}, \mathit{Algo})\}\end{aligned}$$

The set of repairs *S*-*Rep*$(\mathcal{K})$ contains:

$$\begin{aligned}\mathcal{R}_1 =&\{\mathsf{Student}(\mathit{Ann}), \mathsf{Student}(\mathit{Bob})\}\\ \mathcal{R}_2 =&\{\mathsf{Teach}(\mathit{Ann}, \mathit{DB}), \mathsf{Teach}(\mathit{Ann}, \mathit{KR}), \mathsf{Student}(\mathit{Bob})\}\\ \mathcal{R}_3 =&\{\mathsf{Student}(\mathit{Ann}), \mathsf{Prof}(\mathit{Bob}), \mathsf{Teach}(\mathit{Bob}, \mathit{Algo})\}\\ \mathcal{R}_4 =&\{\mathsf{Teach}(\mathit{Ann}, \mathit{DB}), \mathsf{Teach}(\mathit{Ann}, \mathit{KR}), \mathsf{Prof}(\mathit{Bob}), \mathsf{Teach}(\mathit{Bob}, \mathit{Algo})\}\\ \mathcal{R}_5 =&\{\mathsf{Student}(\mathit{Ann}), \mathsf{Teach}(\mathit{KR}, \mathit{Bob})\}\\ \mathcal{R}_6 =&\{\mathsf{Teach}(\mathit{Ann}, \mathit{DB}), \mathsf{Teach}(\mathit{KR}, \mathit{Bob})\}\end{aligned}$$

Observe that $\mathsf{Person}(\mathit{Ann})$ is entailed under CQA semantics, but not under intersection semantics, as the intersection of the repairs is empty. On the other hand, every fact from $\mathcal{D}$ is entailed under brave. By using $\preceq$-optimal repairs, we can obtain further answers under $\preceq$-CQA and $\preceq$-intersection semantics, and eliminate some answers under $\preceq$-brave semantics.

- We have $\leq$-*Rep*$(\mathcal{K}) = \{\mathcal{R}_4\}$, so $\mathsf{Prof}(\mathit{Ann})$ and $\mathsf{Prof}(\mathit{Bob})$ are entailed under $\leq$-intersection, while they were not entailed under (plain) CQA semantics. On the other hand, using $\leq$-optimal repairs rules out $\mathsf{Student}(\mathit{Ann})$, $\mathsf{Student}(\mathit{Bob})$ and $\mathsf{Teach}(\mathit{KR}, \mathit{Bob})$, which do not hold under the $\leq$-brave semantics.
- Suppose we have the prioritization $P = \langle \mathcal{P}_1, \mathcal{P}_2 \rangle$, based on the reliability of the predicates:

  $$\begin{aligned}\mathcal{P}_1 =&\{\mathsf{Student}(\mathit{Ann}), \mathsf{Student}(\mathit{Bob}), \mathsf{Prof}(\mathit{Bob})\}\\ \mathcal{P}_2 =&\{\mathsf{Teach}(\mathit{Ann}, \mathit{DB}), \mathsf{Teach}(\mathit{Ann}, \mathit{KR}), \mathsf{Teach}(\mathit{KR}, \mathit{Bob}), \mathsf{Teach}(\mathit{Bob}, \mathit{Algo})\}\end{aligned}$$

  We obtain $\subseteq_P$-*Rep*$(\mathcal{K}) = \{\mathcal{R}_1, \mathcal{R}_3\}$ and $\leq_P$-*Rep*$(\mathcal{K}) = \{\mathcal{R}_3\}$. Now $\mathsf{Student}(\mathit{Ann})$ is entailed under the $\subseteq_P$-intersection and $\leq_P$-intersection semantics, whereas it was not entailed under (plain) CQA semantics, and it conflicts with a fact entailed under the $\leq$-intersection semantics. Note that $\mathsf{Prof}(\mathit{Bob})$ is entailed under $\leq_P$-intersection semantics, but only the less specific $\mathsf{Person}(\mathit{Bob})$ is entailed under $\subseteq_P$-CQA semantics.
- If we assign facts in $\mathcal{P}_1$ a weight of 2, and facts of $\mathcal{P}_2$ a weight of 1, $\leq_w$-*Rep*$(\mathcal{K}) = \{\mathcal{R}_3, \mathcal{R}_4\}$. Under $\leq_w$-CQA semantics, neither $\mathsf{Prof}(\mathit{Ann})$ nor $\mathsf{Student}(\mathit{Ann})$ is entailed, but only $\mathsf{Person}(\mathit{Ann})$. Under $\leq_w$-intersection semantics, $\mathsf{Prof}(\mathit{Bob})$ is entailed.

▶ **Example 44.** Consider the database (with UCs) $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ defined as follows.

$$\begin{aligned}\mathcal{T} = \{&\mathsf{Name}(x, y) \rightarrow \mathsf{Person}(x),\ \mathsf{Teach}(x, y) \rightarrow \mathsf{Person}(x),\\ &\mathsf{Teach}(x, y) \rightarrow \mathsf{Course}(y),\ \mathsf{Course}(x) \wedge \mathsf{Person}(x) \rightarrow \bot\}\\ \mathcal{D} = \{&\mathsf{Name}(123, \mathit{Ann}), \mathsf{Teach}(\mathit{Bob}, 123)\}\end{aligned}$$

The set of Δ-repairs $S\text{-}\mathit{Rep}_\Delta(\mathcal{K})$ contains:

$$\begin{aligned}\mathcal{R}_1 &= \{\mathsf{Name}(123, \mathit{Ann}), \mathsf{Person}(123)\}\\ &\qquad\qquad \text{i.e., } \mathcal{R}_1 \Delta \mathcal{D} = \{\mathsf{Teach}(\mathit{Bob}, 123), \mathsf{Person}(123)\}\\ \mathcal{R}_2 &= \{\mathsf{Teach}(\mathit{Bob}, 123), \mathsf{Person}(\mathit{Bob}), \mathsf{Course}(123)\}\\ &\qquad\qquad \text{i.e., } \mathcal{R}_2 \Delta \mathcal{D} = \{\mathsf{Name}(123, \mathit{Ann}), \mathsf{Person}(\mathit{Bob}), \mathsf{Course}(123)\}\\ \mathcal{R}_3 &= \emptyset\\ &\qquad\qquad \text{i.e., } \mathcal{R}_3 \Delta \mathcal{D} = \{\mathsf{Name}(123, \mathit{Ann}), \mathsf{Teach}(\mathit{Bob}, 123)\}\end{aligned}$$

- $\leq\text{-}\mathit{Rep}_\Delta(\mathcal{K}) = \{\mathcal{R}_1, \mathcal{R}_3\}$ since $\mathcal{R}_1$ and $\mathcal{R}_3$ have the smallest symmetric differences with $\mathcal{D}$.
- If we take the prioritization $P = \langle \mathcal{P}_1, \mathcal{P}_2 \rangle$ where $\mathcal{P}_1$ contains $\mathsf{Teach}(\mathit{Bob}, 123)$ and all negative literals (to strongly penalize fact additions) and $\mathcal{P}_2 = \{\mathsf{Name}(123, \mathit{Ann})\}$, then $\subseteq_P\text{-}\mathit{Rep}_\Delta(\mathcal{K}) = \{\mathcal{R}_2, \mathcal{R}_3\}$ and $\leq_P\text{-}\mathit{Rep}_\Delta(\mathcal{K}) = \{\mathcal{R}_3\}$. Indeed, $\mathcal{R}_3 \Delta \mathcal{D} \subsetneq_P \mathcal{R}_1 \Delta \mathcal{D}$ while $\mathcal{R}_2 \Delta \mathcal{D}$ and $\mathcal{R}_3 \Delta \mathcal{D}$ are incomparable w.r.t. $\subseteq_P$. However, $\mathcal{R}_3 \Delta \mathcal{D} <_P \mathcal{R}_2 \Delta \mathcal{D}$ since $\mathcal{R}_2 \Delta \mathcal{D}$ contains two facts of level 1 while $\mathcal{R}_3 \Delta \mathcal{D}$ only one.
- If we assign to literals in $\mathcal{P}_1$ a weight of 2, and to those in $\mathcal{P}_2$ a weight of 1, we obtain that $\leq_w\text{-}\mathit{Rep}_\Delta(\mathcal{D}, \mathcal{T}) = \{\mathcal{R}_3\}$ since $\mathcal{R}_3 \Delta \mathcal{D}$ has weight 3, $\mathcal{R}_1 \Delta \mathcal{D}$ 4 and $\mathcal{R}_2 \Delta \mathcal{D}$ 5.

5.2 Optimal repairs based on a priority relation

The second category of preferred repairs we consider is based on a *priority relation* defined over conflicting facts (or literals in the case of Δ-repairs of databases with UCs, even if most work on these kinds of repairs has focused on databases with anti-monotone ICs and KBs).

▶ **Definition 45** (Priority relation [129]). *A* priority relation $\succ$ *for a dataset* $\mathcal{D}$ *w.r.t. an ontology or set of anti-monotone ICs (resp. UCs)* $\mathcal{T}$ *is an acyclic binary relation over the facts (resp. literals) of* $\mathit{Conf}(\mathcal{D}, \mathcal{T})$ *such that if* $\alpha \succ \beta$*, there is* $\mathcal{C} \in \mathit{Conf}(\mathcal{D}, \mathcal{T})$ *s.t.* $\{\alpha, \beta\} \subseteq \mathcal{C}$.

We say that $\succ$ *is* total *if for every pair* $\alpha \neq \beta$ *such that* $\{\alpha, \beta\} \subseteq \mathcal{C}$ *for some* $\mathcal{C} \in \mathit{Conf}(\mathcal{D}, \mathcal{T})$*, either* $\alpha \succ \beta$ *or* $\beta \succ \alpha$*. A* completion *of* $\succ$ *is a total priority relation* $\succ' \supseteq \succ$.

A priority relation $\succ$ *is* score-structured *if there is a function* $s : \bigcup_{\mathcal{C} \in \mathit{Conf}(\mathcal{D}, \mathcal{T})} \mathcal{C} \rightarrow \mathbb{N}$ *such that for every* $\{\alpha, \beta\} \subseteq \mathcal{C}$ *with* $\mathcal{C} \in \mathit{Conf}(\mathcal{D}, \mathcal{T})$*,* $\alpha \succ \beta$ *iff* $s(\alpha) > s(\beta)$.

▶ **Definition 46** (Prioritized KB/database). *A* prioritized KB (resp. database) $\mathcal{K}_\succ = (\mathcal{K}, \succ)$ *consists of a KB (resp. database)* $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ *and a priority relation* $\succ$ *for* $\mathcal{D}$ *w.r.t.* $\mathcal{T}$.

Specifying the priority relation. Before defining the optimal repairs based on a priority relation, let us start by briefly explaining how one can specify such a relation, since it is not realistic to expect users to manually input a binary relation between facts or literals. This question has been recently tackled by a rule-based approach to specifying preferences over conflicting facts [37]. In a nutshell, the idea is that the user defines a few rules that are evaluated on $\mathcal{D}$, possibly taking into account $\mathcal{T}$ and/or some metadata (such as the sources of the facts or the dates they have been added to the dataset) to produce preference statements over facts (using fact identifiers). If these statements yield a cyclic relation, cycle removal techniques are used to obtain a proper priority relation (possibly taking into account a stratification of the preference rules into more or less important ones). While this framework has been defined for KBs, we can in principle extend preference rules to specify a priority relation over literals in the case of databases with UCs.

▶ **Example 47** (Adapted from [37]). The following preference rules express that (i) more recently added facts are preferred, (ii) if $\mathcal{D}$ contains both $\mathsf{FullProf}(c)$ and $\mathsf{AssociateProf}(c)$ for some c, then the former is preferred, and (iii) if a person is declared to belong to $\mathsf{Student}$ and to a subclass Y of Prof, but there is no Teach-fact about this person in $\mathcal{D}$, then the $\mathsf{Student}$-fact is deemed more reliable.

$$\text{(i)} \quad \mathsf{Date}(x_1, y_1) \wedge \mathsf{Date}(x_2, y_2) \wedge y_2 < y_1 \rightarrow \mathsf{pref}(x_1, x_2)$$
$$\text{(ii)} \quad x_1 = \mathbf{id}(\mathsf{FullProf}(y)) \wedge x_2 = \mathbf{id}(\mathsf{AssociateProf}(y)) \rightarrow \mathsf{pref}(x_1, x_2)$$
$$\text{(iii)} \quad Y \sqsubseteq \mathsf{Prof} \wedge x_1 = \mathbf{id}(\mathsf{Student}(y)) \wedge x_2 = \mathbf{id}(Y(y)) \wedge \neg(\exists z \mathsf{Teach}(y, z)) \rightarrow \mathsf{pref}(x_1, x_2)$$

Assume that

$$\mathcal{T} = \{\mathsf{AssociateProf} \sqsubseteq \mathsf{Prof}, \mathsf{FullProf} \sqsubseteq \mathsf{Prof}, \mathsf{AssociateProf} \sqsubseteq \neg\mathsf{FullProf}, \mathsf{Prof} \sqsubseteq \neg\mathsf{Student}\}$$
$$\mathcal{D} = \{\mathsf{Student}(a), \mathsf{AssociateProf}(a), \mathsf{FullProf}(a)\}$$

and that $\mathsf{AssociateProf}(a)$ is associated to a more recent date than $\mathsf{FullProf}(a)$. The rules yield the following preference statements: (i) $\mathsf{AssociateProf}(a)$ is preferred to $\mathsf{FullProf}(a)$, (ii) $\mathsf{FullProf}(a)$ is preferred to $\mathsf{AssociateProf}(a)$, and (iii) $\mathsf{Student}(a)$ is preferred to the two other facts. Since there is a cycle in the preference statements, we need to either remove the whole cycle, which produces the priority relation $\mathsf{Student}(a) \succ \mathsf{AssociateProf}(a)$, $\mathsf{Student}(a) \succ \mathsf{FullProf}(a)$, or to rely on some information on the relative importance of the rules. For example, if rule (ii) has priority over (i), we additionally obtain $\mathsf{FullProf}(a) \succ \mathsf{AssociateProf}(a)$.

5.2.1 Pareto-, globally- and completion-optimal repairs

We now present the three kinds of optimal repair for a prioritized database or KB [129, 30, 32].

▶ **Definition 48** (Optimal repairs of a KB or database with anti-monotone constraints). *Consider a prioritized KB or database with anti-monotone ICs $\mathcal{K}_\succ = (\mathcal{K}, \succ)$ with $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$, and let $\mathcal{R} \in S\text{-}Rep(\mathcal{K})$.*

- *A* Pareto improvement *of $\mathcal{R}$ (w.r.t. $\succ$) is a $\mathcal{T}$-consistent $\mathcal{B} \subseteq \mathcal{D}$ such that there exists $\beta \in \mathcal{B} \setminus \mathcal{R}$ such that $\beta \succ \alpha$ for every $\alpha \in \mathcal{R} \setminus \mathcal{B}$.*
- *A* global improvement *of $\mathcal{R}$ (w.r.t. $\succ$) is a $\mathcal{T}$-consistent $\mathcal{B} \subseteq \mathcal{D}$ such that $\mathcal{B} \neq \mathcal{R}$ and for every $\alpha \in \mathcal{R} \setminus \mathcal{B}$, there exists $\beta \in \mathcal{B} \setminus \mathcal{R}$ such that $\beta \succ \alpha$.*

The repair $\mathcal{R}$ is a:

- Pareto-optimal repair of $\mathcal{K}_\succ$ *if there is no Pareto improvement of $\mathcal{R}$ w.r.t. $\succ$.*
- Globally-optimal repair of $\mathcal{K}_\succ$ *if there is no global improvement of $\mathcal{R}$ w.r.t. $\succ$.*
- Completion-optimal repair of $\mathcal{K}_\succ$ *if $\mathcal{R}$ is a globally-optimal repair of $\mathcal{K}_{\succ'}$, for some completion $\succ'$ of $\succ$.*

We denote by $P\text{-}Rep(\mathcal{K}_\succ)$, $G\text{-}Rep(\mathcal{K}_\succ)$, and $C\text{-}Rep(\mathcal{K}_\succ)$ the sets of Pareto-, globally-, and completion-optimal repairs.

For Δ-repairs, recall from Section 3.2 that $Int_{\mathcal{D}}(\mathcal{B}) = (\mathcal{B} \cap \mathcal{D}) \cup \{\neg\alpha \mid \alpha \in Facts^{\mathcal{T}}_{\mathcal{D}} \setminus (\mathcal{B} \cup \mathcal{D})\}$ is the set of literals upon which $\mathcal{B}$ and $\mathcal{D}$ agree.

▶ **Definition 49** (Optimal repairs of a database with universal constraints)**.** *Consider a prioritized database $\mathcal{K}_{\succ} = (\mathcal{K}, \succ)$ with $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ where $\mathcal{T}$ is a set of UCs, and let $\mathcal{R} \in S\text{-}Rep_{\Delta}(\mathcal{K})$.*

- *A* Pareto improvement *of $\mathcal{R}$ (w.r.t. $\succ$) is a $\mathcal{T}$-consistent dataset $\mathcal{B}$ such that there is $\mu \in Int_{\mathcal{D}}(\mathcal{B}) \setminus Int_{\mathcal{D}}(\mathcal{R})$ such that $\mu \succ \lambda$ for every $\lambda \in Int_{\mathcal{D}}(\mathcal{R}) \setminus Int_{\mathcal{D}}(\mathcal{B})$.*
- *A* global improvement *of $\mathcal{R}$ (w.r.t. $\succ$) is a $\mathcal{T}$-consistent dataset $\mathcal{B}$ s.t. $Int_{\mathcal{D}}(\mathcal{B}) \neq Int_{\mathcal{D}}(\mathcal{R})$ and for every $\lambda \in Int_{\mathcal{D}}(\mathcal{R}) \setminus Int_{\mathcal{D}}(\mathcal{B})$, there exists $\mu \in Int_{\mathcal{D}}(\mathcal{B}) \setminus Int_{\mathcal{D}}(\mathcal{R})$ such that $\mu \succ \lambda$.*

The Pareto-, globally-, and completion-optimal Δ-repairs of $\mathcal{K}_{\succ}$ ($P\text{-}Rep_{\Delta}(\mathcal{K}_{\succ})$, $G\text{-}Rep_{\Delta}(\mathcal{K}_{\succ})$ and $C\text{-}Rep_{\Delta}(\mathcal{K}_{\succ})$) are then defined as in Definition 48.

A Pareto improvement is a global improvement, and a global improvement w.r.t. $\succ$ is a global improvement w.r.t. any completion $\succ'$ of $\succ$, so optimal repairs are related as follows.

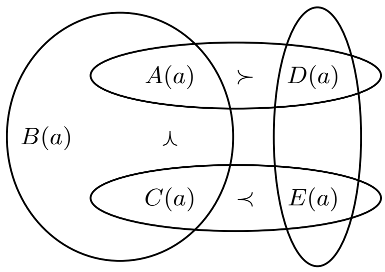
▶ **Proposition 50** ([129, 32])**.** *For every prioritized KB or database $\mathcal{K}_{\succ}$, it holds that*

$$C\text{-}Rep(\mathcal{K}_{\succ}) \subseteq G\text{-}Rep(\mathcal{K}_{\succ}) \subseteq P\text{-}Rep(\mathcal{K}_{\succ}) \subseteq S\text{-}Rep(\mathcal{K}),$$

and similarly for $X\text{-}Rep_{\Delta}(\mathcal{K}_{\succ})$ in the case of Δ-repairs.

Examples 51 and 52 illustrate the three kinds of optimal repair for prioritized KBs and Δ-repairs of prioritized databases with UCs, respectively.

▶ **Example 51.** Consider the KB $\mathcal{K}$ from Example 15, whose conflict hypergraph and repairs are recalled below, and assume that $E(a) \succ C(a)$, $C(a) \succ A(a)$ and $A(a) \succ D(a)$.



$S\text{-}Rep(\mathcal{K}) = \{\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3\}$ with:
$\mathcal{R}_1 = \{A(a), B(a), E(a)\}$
$\mathcal{R}_2 = \{A(a), C(a)\}$
$\mathcal{R}_3 = \{B(a), C(a), D(a)\}$

We show that $C\text{-}Rep(\mathcal{K}_{\succ}) = G\text{-}Rep(\mathcal{K}_{\succ}) = \{\mathcal{R}_1\}$ and $P\text{-}Rep(\mathcal{K}_{\succ}) = \{\mathcal{R}_1, \mathcal{R}_3\}$.

- $\mathcal{R}_2$ is not Pareto-optimal (hence not globally- nor completion-optimal) because $\{A(a), E(a)\}$ is a Pareto improvement of $\mathcal{R}_2$ (since $E(a) \succ C(a)$).
- $\mathcal{R}_3$ is not globally-optimal (hence not completion-optimal) since $\{B(a), E(a), A(a)\}$ is a global improvement of $\mathcal{R}_3$ (since $E(a) \succ C(a)$ and $A(a) \succ D(a)$). However, it is Pareto-optimal, since replacing only $C(a)$ by $E(a)$ leads to a contradiction between $D(a)$ and $E(a)$, and similarly replacing only $D(a)$ with $A(a)$ leads to a conflict $\{A(a), B(a), C(a)\}$.
- $\mathcal{R}_1$ is completion-optimal (hence globally- and Pareto-optimal), since every completion $\succ'$ of $\succ$ is such that $E(a) \succ' D(a)$ (to avoid cycle $E(a) \succ' C(a) \succ' A(a) \succ' D(a) \succ' E(a)$). Indeed, this implies that for every completion $\succ'$, a globally-optimal repair $\mathcal{R}$ of $\mathcal{K}_{\succ'}$ must contain $E(a)$ (otherwise $\mathcal{R} \setminus \{C(a), D(a)\} \cup \{E(a)\}$ would be a global improvement of $\mathcal{R}$ w.r.t. $\succ'$), hence does not contain $C(a)$ nor $D(a)$, so that $\mathcal{R} = \mathcal{R}_1$.

▶ **Example 52** ([32])**.** Let $\mathcal{D} = \{S(a,b), S(a,c), R(d,b), R(d,c)\}$ and $\mathcal{T}$ contains the UCs:

$$S(x,y) \wedge S(x,z) \wedge y \neq z \rightarrow \bot \qquad S(x,y) \rightarrow A(x) \qquad R(y,x) \wedge S(z,x) \rightarrow \bot$$
$$R(x,y) \wedge R(x,z) \wedge y \neq z \rightarrow \bot \qquad S(x,y) \rightarrow B(x)$$

The conflicts are all binary, so the conflict hypergraph is a graph, whose edges are pictured below. Assume that $R(d,b) \succ S(a,b)$, $S(a,b) \succ \neg A(a)$, $S(a,c) \succ R(d,c)$, $S(a,c) \succ \neg B(a)$. We use an arrow $\lambda \rightarrow \mu$ when $\lambda \succ \mu$ and plain lines for conflicting literals with no priority.

$$\begin{tikzcd}
R(d,b) \arrow[r] \arrow[d, dash] & S(a,b) \arrow[r] \arrow[d, dash] \arrow[dr, dash] & \neg A(a) \arrow[dl, dash] \\
R(d,c) & S(a,c) \arrow[l] \arrow[r] & \neg B(a)
\end{tikzcd}$$

It can be verified that the optimal repairs are as follows:

$$\begin{aligned}
C\text{-}Rep_\Delta(\mathcal{K}_\succ) =& \{\{R(d,b), S(a,c), A(a), B(a)\}\} \\
G\text{-}Rep_\Delta(\mathcal{K}_\succ) =& C\text{-}Rep_\Delta(\mathcal{K}_\succ) \cup \{\{R(d,b)\}, \{R(d,c)\}\} \\
P\text{-}Rep_\Delta(\mathcal{K}_\succ) =& G\text{-}Rep_\Delta(\mathcal{K}_\succ) \cup \{\{R(d,c), S(a,b), A(a), B(a)\}\}
\end{aligned}$$

and that $S\text{-}Rep_\Delta(\mathcal{K}) = P\text{-}Rep_\Delta(\mathcal{K}_\succ)$.

There always exists at least one completion-(hence Pareto- and globally-)optimal repair, which can be obtained by the following greedy procedure: while some fact of $\mathcal{D}$ (or literal from $Lits^{\mathcal{T}}_{\mathcal{D}}$) has not been considered, pick one that is maximal w.r.t. $\succ$ among the facts (or literals) not yet considered (i.e., pick some α such that there is no $\beta \succ \alpha$), and add it to the current set if it does not lead to a contradiction [129, 32].

▶ Remark 53 ([129, 32]). When $\succ$ is total, $|P\text{-}Rep(\mathcal{K}_\succ)| = 1$ (hence $C\text{-}Rep(\mathcal{K}_\succ) = G\text{-}Rep(\mathcal{K}_\succ) = P\text{-}Rep(\mathcal{K}_\succ)$). It follows that we may replace "globally-optimal" by "Pareto-optimal" in the definition of completion-optimal repairs.

The following example shows that the binary relation $\preceq_{P\text{-imp}}$ between datasets defined by "$\mathcal{B} \preceq_{P\text{-imp}} \mathcal{B}'$ iff $\mathcal{B} = \mathcal{B}'$ or $\mathcal{B}'$ is a Pareto improvement of $\mathcal{B}$" is *not* a preorder over datasets, so that we cannot recast Pareto-optimal repairs in the terms of Section 5.1 in this way, and similarly for globally-optimal repairs.

▶ **Example 54.** Let $\mathcal{D} = \{\alpha, \beta, \gamma, \delta\}$ and assume that $\mathcal{T}$ is such that $Conf(\mathcal{D}, \mathcal{T}) = \{\{\alpha,\beta\}, \{\alpha,\gamma\}, \{\delta,\beta\}, \{\delta,\gamma\}, \{\beta,\gamma\}\}$. Let $\beta \succ \alpha$, $\beta \succ \delta$ and $\gamma \succ \beta$. There are three repairs: $\mathcal{R}_1 = \{\alpha, \delta\}$, $\mathcal{R}_2 = \{\beta\}$, and $\mathcal{R}_3 = \{\gamma\}$. $\mathcal{R}_2$ is a Pareto improvement of $\mathcal{R}_1$ and $\mathcal{R}_3$ is a Pareto improvement of $\mathcal{R}_2$, but $\mathcal{R}_3$ is a not a Pareto improvement of $\mathcal{R}_1$: the relation "is a Pareto improvement of" is not transitive, so $\preceq_{P\text{-imp}}$ is not a preorder. This example also shows that the relation "is a global improvement of" is not transitive.

The case of score-structured priority relations. When $\succ$ is score-structured with scoring function $s : \bigcup_{\mathcal{C} \in Conf(\mathcal{D},\mathcal{T})} \mathcal{C} \to \mathbb{N}$, the *corresponding prioritization of* $\bigcup_{\mathcal{C} \in Conf(\mathcal{D},\mathcal{T})} \mathcal{C}$ is $P = \langle \mathcal{P}_1, \dots, \mathcal{P}_n \rangle$ such that for every $1 \leq i \leq n$, there exists $m \in \mathbb{N}$ such that $\mathcal{P}_i = \{\alpha \mid s(\alpha) = m\}$, $\mathcal{P}_i \neq \emptyset$, and for every $\alpha_i \in \mathcal{P}_i$, $\alpha_j \in \mathcal{P}_j$, $i < j$ iff $s(\alpha_i) > s(\alpha_j)$. The *corresponding prioritization* of the whole dataset $\mathcal{D}$ (or set of literals $Lits^{\mathcal{T}}_{\mathcal{D}}$ in the case of Δ-repairs of databases with UCs) is obtained by adding the facts from $\mathcal{D} \setminus \bigcup_{\mathcal{C} \in Conf(\mathcal{D},\mathcal{T})} \mathcal{C}$ (or literals from $Lits^{\mathcal{T}}_{\mathcal{D}} \setminus \bigcup_{\mathcal{C} \in Conf(\mathcal{D},\mathcal{T})} \mathcal{C}$) to $\mathcal{P}_1$ (this is an arbitrary choice as we could add them to any of the priority levels and obtain the following result). In this case, all three notions of optimal repairs coincide with $\subseteq_P$-optimal repairs.

▶ **Proposition 55** ([44, Prop. 6.1.14] and [32, Prop. 5])**.** *If $\mathcal{K}_\succ$ is a prioritized KB or database with a score-structured priority relation $\succ$, and P is the corresponding prioritization, then*

$$C\text{-}Rep(\mathcal{K}_\succ) = G\text{-}Rep(\mathcal{K}_\succ) = P\text{-}Rep(\mathcal{K}_\succ) = \subseteq_P\text{-}Rep(\mathcal{K}),$$

and similarly for $X\text{-}Rep_\Delta(\mathcal{K}_\succ)$ in the case of Δ-repairs.

5.2.2 Links with abstract argumentation and active integrity constraints

We now present some correspondences between Pareto-optimal repairs and other notions defined in frameworks that deal with inconsistent information, which speaks in favor of the naturalness of Pareto-optimal repairs.

5.2.2.1 Abstract argumentation

Recall from Section 3.3 that given a KB (or database with anti-monotone ICs) $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$, its associated SETAF $F_\mathcal{K} = (\mathcal{D}, \rightsquigarrow_\mathcal{K})$, with $\rightsquigarrow_\mathcal{K} = \{(\mathcal{C} \setminus \{\alpha\}, \alpha) \mid \mathcal{C} \in Conf(\mathcal{K}), \alpha \in \mathcal{C}\}$, is such that $\mathcal{R} \in S\text{-}Rep(\mathcal{K})$ iff $\mathcal{R}$ is a naive/stable/preferred extension of $F_\mathcal{K}$. To capture *prioritized* KBs (or databases), we need to use *preference-based SETAFs* (PSETAFs) [30].

▶ **Definition 56** (Preference-based set-based argumentation framework)**.** *A PSETAF is a triple $(Args, \rightsquigarrow, \succ)$, where $(Args, \rightsquigarrow)$ is a SETAF, and $\succ$ is an acyclic binary relation over Args called the preference relation.*

The semantics of PSETAFs is given by a reduction to SETAFs:

▶ **Definition 57** (PSETAF extensions)**.** *Given a PSETAF $(Args, \rightsquigarrow, \succ)$, its corresponding SETAF is $(Args, \rightsquigarrow_\succ)$, where the relation $\rightsquigarrow_\succ \subseteq (2^{Args} \setminus \{\emptyset\}) \times Args$ is defined as follows: $S \rightsquigarrow_\succ \alpha$ iff $S \rightsquigarrow \alpha$ and $\alpha \not\succ \beta$ for every $\beta \in S$. A subset $E \subseteq Args$ is a* stable (resp. preferred) extension of a PSETAF $(Args, \rightsquigarrow, \succ)$ *iff it is a stable (resp. preferred) extension of the corresponding SETAF.*

From Pareto-optimal repairs to PSETAF extensions. Let $\mathcal{K}_\succ$ be a prioritized KB (or database with anti-monotone ICs) such that $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ and $\mathcal{D}$ does not contain any self-conflicting fact. The PSETAF $F_{\mathcal{K},\succ}$ associated to $\mathcal{K}_\succ$ is obtained from the SETAF associated to $\mathcal{K}$ by using the priority relation $\succ$ as the preference relation, i.e., $F_{\mathcal{K},\succ} = (\mathcal{D}, \rightsquigarrow_\mathcal{K}, \succ)$ [30].

▶ **Proposition 58** ([30, Th. 35, 36 and 37])**.** *Let $\mathcal{K}_\succ = (\mathcal{K}, \succ)$ be a prioritized KB (or database with anti-monotone ICs) such that $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ and $\mathcal{D}$ does not contain any self-conflicting fact, $F_{\mathcal{K},\succ} = (\mathcal{D}, \rightsquigarrow_\mathcal{K}, \succ)$ be its associated PSETAF, and $\mathcal{R} \subseteq \mathcal{D}$.*

- *$\mathcal{R} \in P\text{-}Rep(\mathcal{K}_\succ)$ iff $\mathcal{R}$ is a stable extension of $F_{\mathcal{K},\succ}$.*
- *If $\succ$ is transitive, then $\mathcal{R} \in P\text{-}Rep(\mathcal{K}_\succ)$ iff $\mathcal{R}$ is a preferred extension of $F_{\mathcal{K},\succ}$.*
- *If every $\mathcal{C} \in Conf(\mathcal{K})$ is such that $|\mathcal{C}| = 2$, then $\mathcal{R} \in P\text{-}Rep(\mathcal{K}_\succ)$ iff $\mathcal{R}$ is a preferred extension of $F_{\mathcal{K},\succ}$.*

Note that it follows that globally-optimal and completion-optimal repairs correspond to proper subsets of the stable extensions of $F_{\mathcal{K},\succ}$, but they do not at present have any analogue in the argumentation setting.

▶ **Example 59.** Consider again the prioritized KB $\mathcal{K}_\succ$ of Example 51. The PSETAF associated to $\mathcal{K}_\succ$ is $F_{\mathcal{K},\succ} = (\mathcal{D}, \rightsquigarrow_\mathcal{K}, \succ)$ where $\rightsquigarrow_\mathcal{K}$ is the attack relation defined in Example 24. The SETAF corresponding to the PSETAF $F_{\mathcal{K},\succ}$ is $(\{A(a), B(a), C(a), D(a), E(a)\}, \rightsquigarrow_\succ)$ where $\rightsquigarrow_\succ$ is as follows.

$$\begin{array}{rlrl}
\{A(a), C(a)\} & \rightsquigarrow_\succ B(a) & \{B(a), C(a)\} & \rightsquigarrow_\succ A(a) \\
\{A(a)\} & \rightsquigarrow_\succ D(a) & & \\
\{E(a)\} & \rightsquigarrow_\succ C(a) & & \\
\{D(a)\} & \rightsquigarrow_\succ E(a) & \{E(a)\} & \rightsquigarrow_\succ D(a)
\end{array}$$

As in the case of standard repairs, it is not difficult to extend this reduction to the case of Pareto-optimal Δ-repairs of databases with UCs.

5.2.2.2 Active integrity constraints

Active integrity constraints (AICs) provide a framework in which universal constraints are enriched with information on what are the allowed update actions (fact deletions or additions) to resolve a given constraint violation [80, 61, 42].

Background on AICs. In this section, we consider that UCs are written in the form $\beta[\vec{x}] \wedge \nu[\vec{x}] \wedge \epsilon[\vec{x}] \rightarrow \bot$, with $\beta[\vec{x}]$ a conjunction of relational atoms, $\nu[\vec{x}]$ a conjunction of *negated* relational atoms and $\epsilon[\vec{x}]$ a conjunction of inequalities (cf. Section 2.1).

▶ **Definition 60** (Update actions). *An* update atom *is of the form* $+P(\vec{x})$ *or* $-P(\vec{x})$ *where* $P(\vec{x})$ *is a relational atom. We use a function* fix *to map relational literals to the corresponding update atoms:* $\mathsf{fix}(P(\vec{x})) = -P(\vec{x})$ *and* $\mathsf{fix}(\neg P(\vec{x})) = +P(\vec{x})$*. An* update action *is a ground update atom, i.e., is of the form* $-\alpha$ *or* $+\alpha$ *with* α *a fact. A set of update actions* $\mathcal{U}$ *is consistent if* $\mathcal{U}$ *does not contain both* $-\alpha$ *and* $+\alpha$ *for some fact* α*. The result of applying a consistent set of update actions* $\mathcal{U}$ *on a dataset* $\mathcal{D}$ *is* $\mathcal{D} \circ \mathcal{U} := \mathcal{D} \setminus \{\alpha \mid -\alpha \in \mathcal{U}\} \cup \{\alpha \mid +\alpha \in \mathcal{U}\}$*.*

▶ **Definition 61** (Active integrity constraints). *An* active integrity constraint *takes the form* $r = \ell_1 \wedge \dots \wedge \ell_n \rightarrow \{A_1, \dots, A_k\}$*, where* $\tau_r := \ell_1 \wedge \dots \wedge \ell_n \rightarrow \bot$ *is a UC, and* $\{A_1, \dots, A_k\}$ *is a non-empty set of update atoms such that each* A_j *is equal to* $\mathsf{fix}(\ell_i)$ *for some* ℓ_i*. A literal* ℓ *of* r *is* non-updatable *if* $\mathsf{fix}(\ell) \notin \{A_1, \dots, A_k\}$*. A dataset* $\mathcal{D}$ *satisfies* r *(*$\mathcal{D} \models r$*) if it satisfies* τ_r*.*

A ground AIC is an AIC that contains no variables. The set $gr_{\mathcal{D}}(r)$ *contains all ground AICs obtained from* r *by (i) replacing variables by constants that occur in* $\mathcal{D}$*, (ii) removing all true* $c \neq d$ *atoms, and (iii) removing all ground AICs with an atom* $c \neq c$*. We let* $gr_{\mathcal{D}}(\eta) := \bigcup_{r \in \eta} gr_{\mathcal{D}}(r)$*, and observe that* $\mathcal{D} \models r$ *iff* $\mathcal{D} \models r_g$ *for every* $r_g \in gr_{\mathcal{D}}(r)$*.*

An AIC is called normal *if its head contains a single update atom. The* normalization *of* $r = \ell_1 \wedge \dots \wedge \ell_n \rightarrow \{A_1, \dots, A_k\}$ *is the set* $N(r) = \{\ell_1 \wedge \dots \wedge \ell_n \rightarrow \{A_i\} \mid 1 \leq i \leq k\}$*. The normalization of a set of AICs* η *is* $N(\eta) = \bigcup_{r \in \eta} N(r)$*.*

▶ **Definition 62** (Repair updates). *A* repair update[2] *of a dataset* $\mathcal{D}$ *w.r.t. a set of AICs* η *is a consistent* $\subseteq$*-minimal set of update actions* $\mathcal{U}$ *such that* $\mathcal{D} \circ \mathcal{U}$ *satisfies all AICs in* η*. We denote the set of repair updates of* $\mathcal{D}$ *w.r.t.* η *by* $Up(\mathcal{D}, \eta)$*.*

It is easy to check that $\{\mathcal{D} \circ \mathcal{U} \mid \mathcal{U} \in Up(\mathcal{D}, \eta)\} = S\text{-}Rep_{\Delta}(\mathcal{D}, \mathcal{T}_\eta)$ where $\mathcal{T}_\eta$ is the set of UCs that correspond to AICs in η. To take into account the restrictions on the possible update actions expressed by the AICs, several classes of repair updates have been defined.

▶ **Definition 63** (Founded, well-founded, grounded and justified repair updates). *A repair update* $\mathcal{U}$ *of* $\mathcal{D}$ *w.r.t.* η *is:*

- Founded *if for every* $A \in \mathcal{U}$*, there exists* $r \in gr_{\mathcal{D}}(\eta)$ *such that* A *is an update action of* r *and* $\mathcal{D} \circ \mathcal{U} \setminus \{A\} \not\models r$*.*
- Well-founded *if there is a sequence of actions* $A_1, \dots, A_n$ *such that* $\mathcal{U} = \{A_1, \dots, A_n\}$*, and for each* $1 \leq i \leq n$*, there exists* $r_i \in gr_{\mathcal{D}}(\eta)$ *such that* A_i *is an update action of* r_i *and* $\mathcal{D} \circ \{A_1, \dots, A_{i-1}\} \not\models r_i$*.*
- Grounded *if for every* $\mathcal{V} \subsetneq \mathcal{U}$*, there is* $r \in gr_{\mathcal{D}}(N(\eta))$ *such that* $\mathcal{D} \circ \mathcal{V} \not\models r$ *and the (only) update action of* r *is in* $\mathcal{U} \setminus \mathcal{V}$ *(recall that* $N(\eta)$ *is the normalization of* η*).*
- Justified *if* $ne(\mathcal{D}, \mathcal{D} \circ \mathcal{U}) \cup \mathcal{U}$ *is a minimal set of update actions closed under* η *that contains the set of no-effect actions* $ne(\mathcal{D}, \mathcal{D} \circ \mathcal{U})$ *where*
  - $ne(\mathcal{D}, \mathcal{D} \circ \mathcal{U}) = \{+\alpha \mid \alpha \in \mathcal{D} \cap (\mathcal{D} \circ \mathcal{U})\} \cup \{-\alpha \mid \alpha \notin \mathcal{D} \cup (\mathcal{D} \circ \mathcal{U}), \alpha \in Facts^{\mathcal{T}}_{\mathcal{D}}\}$
  - $\mathcal{V}$ *is closed under* η *if for every* $r \in gr_{\mathcal{D}}(\eta)$*, if* $\mathcal{V}$ *satisfies all the non-updatable literals of* r*, then* $\mathcal{V}$ *contains an update action of* r*.*

[2] Repair updates are usually called repairs in the AIC literature, we use this term to avoid confusion.

We denote the sets of Δ*-repairs obtained by applying some founded, well-founded, grounded or justified repair update of* $\mathcal{D}$ *w.r.t.* η *by* $\mathit{Found\text{-}Rep}_\Delta(\mathcal{D},\eta)$*,* $\mathit{WellFound\text{-}Rep}_\Delta(\mathcal{D},\eta)$*,* $\mathit{Ground\text{-}Rep}_\Delta(\mathcal{D},\eta)$ *and* $\mathit{Just\text{-}Rep}_\Delta(\mathcal{D},\eta)$*, respectively.*

The relationships between the various kinds of repairs are represented below, where a plain arrow from X to Y means $X \subseteq Y$ and the dotted arrow represents an inclusion that only holds when η is a set of normal AICs. All inclusions may be strict [62, 69, 68].

$$\begin{tikzcd}
\mathit{Ground\text{-}Rep}_\Delta(\mathcal{D},\eta) \arrow[r] \arrow[dr] & \mathit{Found\text{-}Rep}_\Delta(\mathcal{D},\eta) \\
\mathit{Just\text{-}Rep}_\Delta(\mathcal{D},\eta) \arrow[r] \arrow[ur] \arrow[u, dotted] & \mathit{WellFound\text{-}Rep}_\Delta(\mathcal{D},\eta)
\end{tikzcd}$$

From Pareto-optimal repairs to repairs w.r.t. AICs. Given a prioritized database $\mathcal{K}_\succ$ where $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ with $\mathcal{T}$ a set of UCs, its corresponding set of ground AICs [32] is

$$\eta_\succ^\mathcal{T} = \{ r_\mathcal{C} \mid \mathcal{C} \in \mathit{Conf}(\mathcal{K}) \} \text{ where } r_\mathcal{C} := \bigwedge_{\lambda \in \mathcal{C}} \lambda \to \{ \mathsf{fix}(\lambda) \mid \lambda \in \mathcal{C}, \forall \mu \in \mathcal{C}, \lambda \not\succ \mu \}.$$

Intuitively, $\eta_\succ^\mathcal{T}$ expresses that conflicts of $\mathcal{K}$ should be fixed by modifying their least preferred literals according to $\succ$. The Pareto-optimal repairs of $\mathcal{K}_\succ$ coincide with several kinds of repairs of $\mathcal{D}$ w.r.t. $\eta_\succ^\mathcal{T}$.

▶ **Proposition 64** ([32, Prop. 7])**.** *Let* $\mathcal{K}_\succ = (\langle \mathcal{D}, \mathcal{T} \rangle, \succ)$ *be a prioritized database such that* $\mathcal{T}$ *is a set of UCs, and* $\eta_\succ^\mathcal{T}$ *be its corresponding set of ground AICs.*

$$\begin{aligned}
\mathit{P\text{-}Rep}_\Delta(\mathcal{K}_\succ) = \mathit{Just\text{-}Rep}_\Delta(\mathcal{D}, \eta_\succ^\mathcal{T}) = \mathit{Ground\text{-}Rep}_\Delta(\mathcal{D}, \eta_\succ^\mathcal{T}) = \mathit{Found\text{-}Rep}_\Delta(\mathcal{D}, \eta_\succ^\mathcal{T}) \\
\subseteq \mathit{WellFound\text{-}Rep}_\Delta(\mathcal{D}, \eta_\succ^\mathcal{T})
\end{aligned}$$

▶ **Example 65.** We modify slightly the UCs of Example 52 so that some Δ-repairs are not Pareto-optimal. Let $\mathcal{D} = \{S(a,b), S(a,c), R(d,b), R(d,c)\}$ and $\mathcal{T}$ contains the UCs:

$$\begin{aligned}
&S(x,y) \land S(x,z) \land y \neq z \to \bot \qquad && S(x,y) \to A(y) \\
&R(x,y) \land R(x,z) \land y \neq z \to \bot && R(y,x) \land S(z,x) \to \bot
\end{aligned}$$

The edges of the conflict graph and the optimal repairs are shown below, assuming that $R(d,b) \succ S(a,b)$, $S(a,b) \succ \neg A(b)$, $S(a,c) \succ R(d,c)$, $S(a,c) \succ \neg A(c)$.

$$\begin{tikzcd}
R(d,b) \arrow[r] \arrow[d, dash] & S(a,b) \arrow[r] \arrow[d, dash] & \neg A(b) \\
R(d,c) & S(a,c) \arrow[l] \arrow[r] & \neg A(c)
\end{tikzcd}$$

$$\begin{aligned}
\mathit{C\text{-}Rep}_\Delta(\mathcal{K}_\succ) =& \mathit{G\text{-}Rep}_\Delta(\mathcal{K}_\succ) = \{\{R(d,b), S(a,c), A(c)\}\} \\
\mathit{P\text{-}Rep}_\Delta(\mathcal{K}_\succ) =& \mathit{G\text{-}Rep}_\Delta(\mathcal{K}_\succ) \cup \{\{R(d,c), S(a,b), A(b)\}\} \\
\mathit{S\text{-}Rep}_\Delta(\mathcal{K}) =& \mathit{P\text{-}Rep}_\Delta(\mathcal{K}_\succ) \cup \{\{R(d,b)\}, \{R(d,c)\}\}
\end{aligned}$$

The set of ground AICs $\eta_\succ^\mathcal{T}$ corresponding to $\mathcal{K}_\succ$ contains the following AICs:

$$\begin{aligned}
&r_1 : R(d,b) \land S(a,b) \to \{-S(a,b)\} \qquad && r_4 : S(a,b) \land S(a,c) \to \{-S(a,b), -S(a,c)\} \\
&r_2 : R(d,c) \land S(a,c) \to \{-R(d,c)\} && r_5 : R(d,b) \land R(d,c) \to \{-R(d,b), -R(d,c)\} \\
&r_3 : S(a,b) \land \neg A(b) \to \{+A(b)\} && r_6 : S(a,c) \land \neg A(c) \to \{+A(c)\}
\end{aligned}$$

We can check that the two Pareto-optimal Δ-repairs correspond exactly to the justified, grounded and founded repair updates:

- $\mathcal{U}_1 = \{-S(a,b), -S(a,c), -R(d,c)\}$ and $\mathcal{U}_2 = \{-S(a,b), -S(a,c), -R(d,b)\}$, which correspond respectively to the Δ-repairs $\mathcal{R}_1 = \{R(d,b)\}$ and $\mathcal{R}_2 = \{R(d,c)\}$, are not founded (hence not grounded nor justified). Indeed, $-S(a,c) \in \mathcal{U}_1$ while the only AIC with $-S(a,c)$ in the head (r_4) is satisfied by $\mathcal{R}_1 \cup \{S(a,c)\}$. Regarding $\mathcal{U}_2$, $-S(a,b) \in \mathcal{U}_2$ and the two AICs with $-S(a,b)$ in the head (r_1 and r_4) are satisfied by $\mathcal{R}_2 \cup \{S(a,b)\}$.
- The repair update $\mathcal{U}_3 = \{-R(d,b), -S(a,c), +A(b)\}$, which corresponds to the Pareto-optimal Δ-repair $\mathcal{R}_3 = \{R(d,c), S(a,b), A(b)\}$ is grounded (hence founded) and justified. Indeed, (1) for every $\mathcal{V} \subsetneq \mathcal{U}_3$, there is r in the normalization $N(\eta^{\mathcal{T}}_{\succ})$ of $\eta^{\mathcal{T}}_{\succ}$ such that $\mathcal{D} \circ \mathcal{V} \not\models r$ and the only update action of r is in $\mathcal{U}_3 \setminus \mathcal{V}$, so $\mathcal{U}_3$ is grounded, and (2) $ne(\mathcal{D}, \mathcal{D} \circ \mathcal{U}_3) \cup \mathcal{U}_3 = \{+R(d,c), +S(a,b), -A(c), -R(d,b), -S(a,c), +A(b)\}$ is a minimal set of update actions closed under $\eta^{\mathcal{T}}_{\succ}$ that contains $ne(\mathcal{D}, \mathcal{D} \circ \mathcal{U}_3) = \{+R(d,c), +S(a,b), -A(c)\}$ (since this set of update actions satisfies all the non-updatable literals of r_3, r_4 and r_5).
- The repair update $\mathcal{U}_4 = \{-R(d,c), -S(a,b), +A(c)\}$, which corresponds to the Pareto-optimal Δ-repair $\mathcal{R}_4 = \{R(d,b), S(a,c), A(c)\}$ is also grounded, founded and justified.

The above reduction is data-dependent and requires us to create potentially exponentially many ground AICs (one for every conflict). In the case of *denial constraints*, however, there exists an alternative data-independent reduction, provided that the priority relation $\succ$ is specified in the dataset [32]. It assumes the existence of a predicate $P_{\succ} \in \mathbf{P}$, that the first attribute of each relation in $\mathbf{P} \setminus \{P_{\succ}\}$ stores a unique fact identifier, and that $P_{\succ}$ stores pairs of such identifiers. Then given a set of DCs $\mathcal{T}$ over $\mathbf{P} \setminus \{P_{\succ}\}$, $min(\mathcal{T})$ is the set of DCs that is equivalent to $\mathcal{T}$ but has the property that for every dataset $\mathcal{D}$, the conflicts of $\mathcal{D}$ w.r.t. $\mathcal{T}$ are precisely the images of constraint bodies of $min(\mathcal{T})$ on $\mathcal{D}$. This can be achieved by replacing each $\varphi \to \bot \in \mathcal{T}$ with all refinements obtained by (dis)equating variables in φ with each other, then removing any subsumed constraints. For example, if $\mathcal{T} = \{R(x,x) \to \bot, R(x,y) \wedge S(y) \to \bot\}$, then $min(\mathcal{T})$ contains $R(x,x) \to \bot$ and $R(x,y) \wedge S(y) \wedge x \neq y \to \bot$, so $\{R(a,a), S(a)\}$ is no longer an image of a constraint body. The set of AICs corresponding to $\mathcal{T}$, $\eta^{\mathcal{T}}$, contains all AICs

$$\big(\ell_1 \wedge \ldots \wedge \ell_n \wedge \varepsilon \wedge \bigwedge_{\substack{j \in \{1,\ldots,n\},\\ j \neq i}} \neg P_{\succ}(id_i, id_j)\big) \to \{-\ell_i\}$$

such that $\ell_1 \wedge \ldots \wedge \ell_n \wedge \varepsilon \to \bot \in min(\mathcal{T})$, $i \in \{1,\ldots,n\}$, and for every $1 \leq k \leq n$, $\ell_k = R(id_k, \vec{t})$ for some $R, \vec{t}$.

▶ **Proposition 66** ([32, Prop. 8]). *Let $\mathcal{K}_{\succ} = (\langle \mathcal{D}, \mathcal{T} \rangle, \succ)$ be a prioritized database such that $\mathcal{T}$ is a set of DCs, and $\eta^{\mathcal{T}}$ be the set of AICs corresponding to $\mathcal{T}$.*

$$\begin{aligned} P\text{-}Rep(\mathcal{K}_{\succ}) = Just\text{-}Rep_{\Delta}(\mathcal{D}, \eta^{\mathcal{T}}) = Ground\text{-}Rep_{\Delta}(\mathcal{D}, \eta^{\mathcal{T}}) = Found\text{-}Rep_{\Delta}(\mathcal{D}, \eta^{\mathcal{T}}) \\ \subseteq WellFound\text{-}Rep_{\Delta}(\mathcal{D}, \eta^{\mathcal{T}}) \end{aligned}$$

From repairs w.r.t. AICs to Pareto-optimal repairs. Interestingly, certain "well-behaved" sets of AICs, for which, in particular, justified, grounded and founded repairs coincide, can be translated into prioritized databases whose Pareto-optimal repairs coincide with these repairs [32]. The conditions on the AICs are quite involved though.

5.3 Preferred repairs based on preference rules

The last category of preferred repairs we consider is inspired by work in logic programming [46] and uses *preference rules* to express context-dependent preferences. Preferred repairs are then those that maximally (w.r.t. some preorder) satisfy these preference rules [50].

▶ **Definition 67** (Preference program [50]). *A* preference rule ρ *is an expression of the form*

$$P[\vec{x}] \succ Q[\vec{y}] \leftarrow \exists \vec{z} \varphi[\vec{z}, \vec{w}],$$

where $P[\vec{x}]$ *and* $Q[\vec{y}]$ *are two relational atoms whose variables are* $\vec{x}$ *and* $\vec{y}$ *respectively (and may contain constants),* $\vec{x} \cap \vec{z} = \vec{y} \cap \vec{z} = \emptyset$*, and* $\varphi[\vec{z}, \vec{w}]$ *is a (possibly empty) conjunction of relational atoms with variables* $\vec{z} \cup \vec{w}$*. A set of preference rules is called a* preference program.

▶ **Definition 68** (Preference program-based prioritized KB or database [50]). *A* (preference program-based) prioritized KB or database $\mathcal{K}_\Pi = (\mathcal{K}, \Pi)$ *consists of a KB (or database with anti-monotone ICs)* $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ *and a preference program* Π.

A ground instance of a preference rule ρ *(w.r.t.* $\mathcal{K}$ *and* Π*) is a preference rule obtained from* ρ *by replacing all its free variables by constants that occur in* $\mathcal{K}$ *or* Π*. The set of all ground instances of all rules in* Π *is denoted by* $grnd_{\mathcal{K}}(\Pi)$.

A $\mathcal{T}$*-consistent dataset* $\mathcal{B}$ satisfies a ground instance $P(\vec{a}) \succ Q(\vec{b}) \leftarrow \exists \vec{z} \varphi[\vec{z}, \vec{c}]$ *(w.r.t.* $\mathcal{K}$*) if* $\langle \mathcal{B}, \mathcal{T} \rangle \models \exists \vec{z} \varphi[\vec{z}, \vec{c}]$ *implies that* $\langle \mathcal{B}, \mathcal{T} \rangle \not\models Q(\vec{b})$ *or* $\langle \mathcal{B}, \mathcal{T} \rangle \models P(\vec{a})$*, i.e., if the following holds:*

$$\langle \mathcal{B}, \mathcal{T} \rangle \models \exists \vec{z} \varphi[\vec{z}, \vec{c}] \Rightarrow \Big(\langle \mathcal{B}, \mathcal{T} \rangle \models Q(\vec{b}) \Rightarrow \langle \mathcal{B}, \mathcal{T} \rangle \models P(\vec{a}) \Big)$$

The set of preference rules from $grnd_{\mathcal{K}}(\Pi)$ *satisfied by* $\mathcal{B}$ *is denoted by* $GrSat(\mathcal{B}, \mathcal{K}_\Pi)$.

▶ **Definition 69** (Preference program-based preferred repairs [50]). *Let* $\mathcal{K}_\Pi = (\mathcal{K}, \Pi)$ *be a (preference program-based) prioritized KB (or database) and* $\trianglelefteq$ *be a preorder on the subsets of* $grnd_{\mathcal{K}}(\Pi)$*. A repair* $\mathcal{R}$ *of* $\mathcal{K}$ *is* $\Pi_\trianglelefteq$-preferred *iff there is no repair* $\mathcal{R}'$ *of* $\mathcal{K}$ *such that* $GrSat(\mathcal{R}, \mathcal{K}_\Pi) \triangleleft GrSat(\mathcal{R}', \mathcal{K}_\Pi)$*. We denote by* $Pr_\trianglelefteq\text{-}Rep(\mathcal{K}_\Pi)$ *the set of* $\Pi_\trianglelefteq$*-preferred repairs.*

Note that even if the binary relation over datasets $\preceq_{\Pi_\trianglelefteq}$ defined by $\mathcal{B} \preceq_{\Pi_\trianglelefteq} \mathcal{B}'$ iff $GrSat(\mathcal{B}, \mathcal{K}_\Pi) \trianglelefteq GrSat(\mathcal{B}', \mathcal{K}_\Pi)$ is a preorder, we cannot recast $\Pi_\trianglelefteq$-preferred repairs in the terms of Section 5.1. Indeed, $\preceq_{\Pi_\trianglelefteq}$ is not strictly $\subseteq$-monotone, since the empty set trivially satisfies every preference rule ground instance, hence is $\preceq_{\Pi_\trianglelefteq}$-maximal.

▶ **Example 70.** Consider the KB $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$, where $\mathcal{T}$ expresses that computer science (CS), economics (Eco) and literature (Lit) courses and professors, respectively, are disjoint.

$$\begin{aligned} \mathcal{T} = \{ & \mathsf{CSCourse} \sqsubseteq \neg \mathsf{EcoCourse},\ \mathsf{CSCourse} \sqsubseteq \neg \mathsf{LitCourse},\ \mathsf{LitCourse} \sqsubseteq \neg \mathsf{EcoCourse}, \\ & \mathsf{CSPr} \sqsubseteq \neg \mathsf{EcoPr},\ \mathsf{CSPr} \sqsubseteq \neg \mathsf{LitPr},\ \mathsf{LitPr} \sqsubseteq \neg \mathsf{EcoPr} \} \\ \mathcal{D} = \{ & \mathsf{CSCourse}(c), \mathsf{EcoCourse}(c), \mathsf{LitCourse}(c), \mathsf{CSPr}(p), \mathsf{EcoPr}(p), \mathsf{Teach}(p, c) \} \end{aligned}$$

There are six repairs:

$$\begin{aligned} &\mathcal{R}_1 = \{\mathsf{CSCourse}(c), \mathsf{CSPr}(p), \mathsf{Teach}(p, c)\} && \mathcal{R}_2 = \{\mathsf{CSCourse}(c), \mathsf{EcoPr}(p), \mathsf{Teach}(p, c)\} \\ &\mathcal{R}_3 = \{\mathsf{EcoCourse}(c), \mathsf{CSPr}(p), \mathsf{Teach}(p, c)\} && \mathcal{R}_4 = \{\mathsf{EcoCourse}(c), \mathsf{EcoPr}(p), \mathsf{Teach}(p, c)\} \\ &\mathcal{R}_5 = \{\mathsf{LitCourse}(c), \mathsf{CSPr}(p), \mathsf{Teach}(p, c)\} && \mathcal{R}_6 = \{\mathsf{LitCourse}(c), \mathsf{EcoPr}(p), \mathsf{Teach}(p, c)\} \end{aligned}$$

Assume that Π contains the following preference rules, which express that when a course is taught by a professor in a given subject, we prefer the fact that states that this course is on that subject to facts that state it is on another subject:

$$\begin{aligned} &\rho_1 : && \mathsf{CSCourse}(x) \succ \mathsf{EcoCourse}(x) \leftarrow \exists y \mathsf{Teach}(y, x) \land \mathsf{CSPr}(y) \\ &\rho_2 : && \mathsf{EcoCourse}(x) \succ \mathsf{CSCourse}(x) \leftarrow \exists y \mathsf{Teach}(y, x) \land \mathsf{EcoPr}(y) \\ &\rho_3 : && \mathsf{CSCourse}(x) \succ \mathsf{LitCourse}(x) \leftarrow \exists y \mathsf{Teach}(y, x) \land \mathsf{CSPr}(y) \\ &\rho_4 : && \mathsf{LitCourse}(x) \succ \mathsf{CSCourse}(x) \leftarrow \exists y \mathsf{Teach}(y, x) \land \mathsf{LitPr}(y) \\ &\rho_5 : && \mathsf{EcoCourse}(x) \succ \mathsf{LitCourse}(x) \leftarrow \exists y \mathsf{Teach}(y, x) \land \mathsf{EcoPr}(y) \\ &\rho_6 : && \mathsf{LitCourse}(x) \succ \mathsf{EcoCourse}(x) \leftarrow \exists y \mathsf{Teach}(y, x) \land \mathsf{LitPr}(y) \end{aligned}$$

Since all repairs trivially satisfy the ground instances obtained by replacing x by p, we identify each ρ_i with its ground instance obtained by replacing x by c in what follows.

- $\mathcal{R}_1$ and $\mathcal{R}_4$ satisfy all rules. For example, $\mathcal{R}_1$ satisfies ρ_1 because $\langle \mathcal{R}_1, \mathcal{T} \rangle \models \mathsf{CSCourse}(c)$, and ρ_2 because $\langle \mathcal{R}_1, \mathcal{T} \rangle \not\models \exists y \mathsf{Teach}(y, c) \wedge \mathsf{EcoPr}(y)$.
- $\mathcal{R}_2$ satisfies all rules but ρ_2: $\langle \mathcal{R}_2, \mathcal{T} \rangle \models \exists y \mathsf{Teach}(y, c) \wedge \mathsf{EcoPr}(y)$ and $\langle \mathcal{R}_2, \mathcal{T} \rangle \models \mathsf{CSCourse}(c)$ while $\langle \mathcal{R}_2, \mathcal{T} \rangle \not\models \mathsf{EcoCourse}(c)$. Note that it satisfies ρ_5 because $\langle \mathcal{R}_2, \mathcal{T} \rangle \not\models \mathsf{LitCourse}(c)$.
- $\mathcal{R}_3$ satisfies all rules but ρ_1; $\mathcal{R}_5$ satisfies all rules but ρ_3; and $\mathcal{R}_6$ satisfies all rules but ρ_5.

Hence, for $\trianglelefteq \in \{\subseteq, \leq\}$, $Pr_{\trianglelefteq}\text{-}Rep(\mathcal{K}_\Pi) = \{\mathcal{R}_1, \mathcal{R}_4\}$, which correspond to the two ways of repairing $\mathcal{D}$ by selecting a pair of a course and a professor with matching subject.

To illustrate how choosing $\subseteq$ or $\leq$ may lead to different $\Pi_{\trianglelefteq}$-preferred repairs, consider $\mathcal{K}' = \langle \mathcal{D}, \mathcal{T}' \rangle$ where $\mathcal{T}'$ keeps only the three first axioms of $\mathcal{T}$ (i.e., a course has only one subject but professors can teach several subjects). There are three repairs of $\mathcal{K}'$:

$$\mathcal{R}'_1 = \{\mathsf{CSCourse}(c), \mathsf{CSPr}(p), \mathsf{EcoPr}(p), \mathsf{Teach}(p, c)\}$$
$$\mathcal{R}'_2 = \{\mathsf{EcoCourse}(c), \mathsf{CSPr}(p), \mathsf{EcoPr}(p), \mathsf{Teach}(p, c)\}$$
$$\mathcal{R}'_3 = \{\mathsf{LitCourse}(c), \mathsf{CSPr}(p), \mathsf{EcoPr}(p), \mathsf{Teach}(p, c)\}$$

We obtain $Pr_{\subseteq}\text{-}Rep(\mathcal{K}'_\Pi) = \{\mathcal{R}'_1, \mathcal{R}'_2, \mathcal{R}'_3\}$ while $Pr_{\leq}\text{-}Rep(\mathcal{K}'_\Pi) = \{\mathcal{R}'_1, \mathcal{R}'_2\}$. Indeed $\mathcal{R}'_1$ satisfies all rules but ρ_2; $\mathcal{R}'_2$ satisfies all rules but ρ_1; and $\mathcal{R}'_3$ satisfies ρ_1, ρ_2, ρ_4, and ρ_6.

While the preference rules used to define a priority relation among conflicting facts in Section 5.2 (cf. Example 47) are quite similar to the ones considered here from a syntactic point of view, they have a different semantics: the former aim at defining a priority relation between conflicting facts globally, taking into account the whole dataset $\mathcal{D}$, while the latter aim at defining directly a preference relation between repairs, so that the relation $\succ$ between facts is specific to each repair. If we consider the counterparts of the preference rules of Example 70 that specify a priority relation as in Example 47 and evaluate them over $\mathcal{D}$, we obtain that $\mathsf{CSCourse}(c)$ is preferred to $\mathsf{EcoCourse}(c)$ (counterpart of ρ_1), $\mathsf{EcoCourse}(c)$ is preferred to $\mathsf{CSCourse}(c)$ (counterpart of ρ_2), $\mathsf{CSCourse}(c)$ is preferred to $\mathsf{LitCourse}(c)$ (counterpart of ρ_3), and $\mathsf{EcoCourse}(c)$ is preferred to $\mathsf{LitCourse}(c)$ (counterpart of ρ_5), which yields (assuming that all rules are equally important) the priority relation $\mathsf{CSCourse}(c) \succ \mathsf{LitCourse}(c)$, $\mathsf{EcoCourse}(c) \succ \mathsf{LitCourse}(c)$. The optimal repairs w.r.t. this priority relation are $\{\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3, \mathcal{R}_4\}$ for $\mathcal{K}_\succ$ and $\{\mathcal{R}'_1, \mathcal{R}'_2\}$ for $\mathcal{K}'_\succ$.

Capturing $\preceq$-optimal repairs for $\preceq \in \{\leq, \leq_w, \subseteq_P, \leq_P\}$. Given $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ and a preorder $\preceq \in \{\leq, \leq_w, \subseteq_P, \leq_P\}$ over the subsets of $\mathcal{D}$, it is possible to define a preference program Π, a fact α and a preorder $\trianglelefteq$ on the subsets of $grnd_{\mathcal{K}}(\Pi)$ such that there is a one-to-one correspondence between $\preceq$-optimal repairs of $\mathcal{K}$ and $\Pi_{\trianglelefteq}$-preferred repairs of $\mathcal{K}' = \langle \mathcal{D} \cup \{\alpha\}, \mathcal{T} \rangle$. More precisely:

$$Pr_{\trianglelefteq}\text{-}Rep(\mathcal{K}'_\Pi) = \{\mathcal{R} \cup \{\alpha\} \mid \mathcal{R} \in \preceq\text{-}Rep(\mathcal{K})\}.$$

The idea is simply to encode the preorder over the subsets of $\mathcal{D}$ in Π: take $\alpha = A(a)$ such that A and a do not occur in $\mathcal{K}$, $\Pi = \{\beta \succ \alpha \leftarrow \mid \beta \in \mathcal{D}\}$, and for all subsets S and S' of $grnd_{\mathcal{K}}(\Pi)$ (which is equal to Π in this case) let $S \trianglelefteq S'$ iff $\{\beta \mid \beta \succ \alpha \leftarrow \in S\} \preceq \{\beta \mid \beta \succ \alpha \leftarrow \in S'\}$. Then for all $\mathcal{T}$-consistent $\mathcal{B}$ and $\mathcal{B}'$ subsets of $\mathcal{D}$, $\mathcal{B} \preceq \mathcal{B}'$ iff $GrSat(\mathcal{B} \cup \{\alpha\}, \mathcal{K}'_\Pi) \trianglelefteq GrSat(\mathcal{B}' \cup \{\alpha\}, \mathcal{K}'_\Pi)$. Since all repairs of $\mathcal{K}'$ contain α (so that subsets of $\mathcal{D}$, which trivially satisfy all rules from Π, are not repairs of $\mathcal{K}'$, hence not $\Pi_{\trianglelefteq}$-preferred repairs), it follows that $\mathcal{R}$ is a $\preceq$-optimal repair of $\mathcal{K}$ iff $\mathcal{R} \cup \{\alpha\}$ is a $\Pi_{\trianglelefteq}$-preferred repair of $\mathcal{K}'$. In the case of $\leq$, this reduction can be adapted into a data-independent one [50, Prop. 42].

Comparison with optimal repairs based on a priority relation. Given a prioritized KB or database $\mathcal{K}_{\succ}$, we can consider the preference program $\Pi = \{\alpha \succ \beta \leftarrow \mid \alpha \succ \beta\}$. However, the $\Pi_{\trianglelefteq}$-preferred repairs of $\mathcal{K}_{\Pi}$ are incomparable with the optimal repairs of $\mathcal{K}_{\succ}$ [50]. For example, assume that $\mathcal{D} = \{\alpha, \beta, \gamma\}$, $\mathit{Conf}(\mathcal{D}, \mathcal{T}) = \{\{\alpha, \beta\}, \{\alpha, \gamma\}\}$ and $\alpha \succ \beta$. There are two repairs, $\{\alpha\}$ and $\{\beta, \gamma\}$. The priority relation $\succ$ is score-structured so Pareto-, globally-, completion- and $\subseteq_P$-optimal repairs coincide and both $\{\alpha\}$ and $\{\beta, \gamma\}$ are optimal. However, only $\{\alpha\}$ is a $\Pi_{\subseteq}$-preferred (resp. $\Pi_{\leq}$-preferred) repair. For an example where there are more $\Pi_{\subseteq}$-preferred repairs, consider $\mathcal{D} = \{\alpha, \beta, \gamma\}$, $\mathit{Conf}(\mathcal{D}, \mathcal{T}) = \{\{\alpha, \beta\}, \{\beta, \gamma\}\}$ and $\alpha \succ \beta$, $\beta \succ \gamma$. Again $\succ$ is score-structured, and the only optimal repair of $\mathcal{K}_{\succ}$ is $\{\alpha, \gamma\}$. However both $\{\alpha, \gamma\}$ and $\{\beta\}$ are $\Pi_{\subseteq}$-preferred repairs of $\mathcal{K}_{\Pi}$ because $\{\beta\}$ satisfies $\beta \succ \gamma \leftarrow$ while $\{\alpha, \gamma\}$ does not. More examples and discussion can be found in [50, Sec. 6].

5.4 Related work: "Optimal repairs" that are not dataset repairs

We conclude this section with a brief presentation of other notions of "optimal repairs" which are not dataset repairs (i.e., do not form a subset of $S\text{-}\mathit{Rep}(\mathcal{K})$).

5.4.1 Repairs based on non strictly $\subseteq$-monotone weight-based preorders

A generalization of $\leq_w$-optimal repairs based on aggregation functions has been defined for databases with anti-monotone ICs [122]. An aggregation function g maps every dataset $\mathcal{B}$ (associated with a weight function w) to a non-negative rational number $g(\mathcal{B})$ (cf. [122] for the precise definition). Prominent examples of such functions are $\Sigma(\mathcal{B}) = \Sigma_{\alpha \in \mathcal{B}} w(\alpha)$ (which yields $\leq_w$-optimal repairs), $\max(\mathcal{B}) = \max_{\alpha \in \mathcal{B}} w(\alpha)$, or $\mathsf{average}(\mathcal{B}) = \frac{\Sigma_{\alpha \in \mathcal{B}} w(\alpha)}{|\mathcal{B}|}$.

▶ **Definition 71** (Aggregation-based repairs). *Given an aggregation function g and a KB (or database with anti-monotone ICs) $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ with a weight function w, a g-repair of $\mathcal{K}$ is a $\mathcal{T}$-consistent $\mathcal{R} \subseteq \mathcal{D}$ such that (i) there is no $\mathcal{T}$-consistent $\mathcal{R}' \subseteq \mathcal{D}$ with $g(\mathcal{R}) < g(\mathcal{R}')$ and (ii) for every $\mathcal{T}$-consistent $\mathcal{R}' \subseteq \mathcal{D}$, if $g(\mathcal{R}') = g(\mathcal{R})$ and $\mathcal{R} \subseteq \mathcal{R}'$, then $\mathcal{R} = \mathcal{R}'$.*

Note that if we define the preorder corresponding to an aggregation function g as $\mathcal{B} \preceq_g \mathcal{B}'$ iff $g(\mathcal{B}) \leq g(\mathcal{B}')$, condition (i) of Definition 71 can be written as "there is no $\mathcal{T}$-consistent $\mathcal{R}' \subseteq \mathcal{D}$ with $\mathcal{R} \prec_g \mathcal{R}'$". This definition thus modifies Definition 41 by (1) dropping the condition that the preorder is strictly $\subseteq$-monotone and (2) partially compensating the effects of dropping this requirement with condition (ii), which expresses that a g-repair needs to be $\subseteq$-maximal among the $\preceq_g$-maximal $\mathcal{T}$-consistent subsets of $\mathcal{D}$. The following example shows that this is not sufficient to ensure that g-repairs are dataset repairs.

▶ **Example 72** ([122]). Let $\mathcal{D} = \{\beta, \gamma\}$ be such that $\mathcal{D}$ is $\mathcal{T}$-consistent, so that $S\text{-}\mathit{Rep}(\mathcal{D}, \mathcal{T}) = \{\mathcal{D}\}$, and assume that $w(\beta) = 1$ and $w(\gamma) = 2$. Consider the aggregation function min defined by $\min(\mathcal{B}) = \min_{\alpha \in \mathcal{B}} w(\alpha)$. Then $\{\gamma\}$ is the only min-repair of $\mathcal{D}$ w.r.t. $\mathcal{T}$ since $\min(\{\beta, \gamma\}) = 1 < \min(\{\gamma\}) = 2$. Hence the set of min-repairs is not a subset of $S\text{-}\mathit{Rep}(\mathcal{D}, \mathcal{T})$.

5.4.2 Selecting a single $\mathcal{T}$-consistent dataset from a prioritized dataset

Instead of considering all optimal repairs of a prioritized KB or database $\mathcal{K}_{\succ}$, one may be interested in selecting a *single* $\mathcal{T}$-consistent dataset. Several such "repairs" have been proposed, depending on whether $\succ$ is based on a prioritization $P = \langle \mathcal{P}_1, \dots, \mathcal{P}_n \rangle$ of $\mathcal{D}$ (i.e., $\alpha \succ \beta$ iff $\{\alpha, \beta\} \subseteq \mathcal{C}$ for some conflict $\mathcal{C}$, $\alpha \in \mathcal{P}_i$, $\beta \in \mathcal{P}_j$ and $i < j$), a preorder $\trianglelefteq$ over $\mathcal{D}$ (i.e., $\alpha \succ \beta$ iff $\{\alpha, \beta\} \subseteq \mathcal{C}$ for some conflict $\mathcal{C}$ and $\alpha \triangleright \beta$), or is an arbitrary priority relation. These notions of repair stem from the KB area, so we give the definitions for KBs or databases with anti-monotone ICs, but we could in principle extend them to handle database with universal constraints by considering literals as in Section 5.2. We provide below all definitions before presenting their relationships, properties, and an example.

The next definition presents the different kinds of "unique repair" that have been proposed when the dataset is partitioned into priority levels, i.e., the most restricted case.

▶ **Definition 73** (Unique repairs based on a prioritization [24]). *Let $\mathcal{K}_\succ$ be a prioritized KB (or database with anti-monotone ICs) such that $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ and $\succ$ is based on a prioritization $P = \langle \mathcal{P}_1, \dots, \mathcal{P}_n \rangle$ of $\mathcal{D}$.*

- *The* possibilistic repair *of $\mathcal{K}_\succ$ is $\pi(\mathcal{K}_\succ) = \mathcal{P}_1 \cup \dots \cup \mathcal{P}_{inc(\mathcal{K}_\succ)-1}$, where $inc(\mathcal{K}_\succ)$ is the* inconsistency degree *of $\mathcal{K}_\succ$, which is such that $\mathcal{P}_1 \cup \dots \cup \mathcal{P}_{inc(\mathcal{K}_\succ)}$ is $\mathcal{T}$-inconsistent and $\mathcal{P}_1 \cup \dots \cup \mathcal{P}_{inc(\mathcal{K}_\succ)-1}$ is $\mathcal{T}$-consistent.*
- *The* linear repair *of $\mathcal{K}_\succ$ is $\ell(\mathcal{K}_\succ) = \ell(\mathcal{D})$ with:*
  - *$\ell(\mathcal{P}_1) = \mathcal{P}_1$ if $\mathcal{P}_1$ is $\mathcal{T}$-consistent, $\ell(\mathcal{P}_1) = \emptyset$ otherwise;*
  - *for $i > 1$, $\ell(\mathcal{P}_1 \cup \dots \cup \mathcal{P}_i) = \ell(\mathcal{P}_1 \cup \dots \cup \mathcal{P}_{i-1}) \cup \mathcal{P}_i$ if $\ell(\mathcal{P}_1 \cup \dots \cup \mathcal{P}_{i-1}) \cup \mathcal{P}_i$ is $\mathcal{T}$-consistent, $\ell(\mathcal{P}_1 \cup \dots \cup \mathcal{P}_i) = \ell(\mathcal{P}_1 \cup \dots \cup \mathcal{P}_{i-1})$ otherwise.*
- *The* non-defeated repair *of $\mathcal{K}_\succ$ is $nd(\mathcal{K}_\succ) = \bigcup_{i=1}^n \mathcal{R}^i_\cap$ where $\mathcal{R}^i_\cap = \bigcap_{\mathcal{R} \in S\text{-}Rep(\mathcal{P}_1 \cup \dots \cup \mathcal{P}_i, \mathcal{T})} \mathcal{R}$.*
- *The* linear-based non-defeated repair *of $\mathcal{K}_\succ$ is $\ell nd(\mathcal{K}_\succ) = \ell nd(\mathcal{D})$ with:*
  - *$\ell nd(\mathcal{P}_1) = \bigcap_{\mathcal{R} \in S\text{-}Rep(\mathcal{P}_1, \mathcal{T})} \mathcal{R}$;*
  - *for $i > 1$, $\ell nd(\mathcal{P}_1 \cup \dots \cup \mathcal{P}_i) = \ell nd(\mathcal{P}_1 \cup \dots \cup \mathcal{P}_{i-1}) \cup \bigcap_{\mathcal{R} \in S\text{-}Rep(\ell nd(\mathcal{P}_1 \cup \dots \cup \mathcal{P}_{i-1}) \cup \mathcal{P}_i, \mathcal{T})} \mathcal{R}$.*
- *The* prioritized inclusion-based non-defeated repair *of $\mathcal{K}_\succ$ is $pind(\mathcal{K}_\succ) = \bigcup_{i=1}^n \mathcal{R}^i_{P,\cap}$ where $\mathcal{R}^i_{P,\cap} = \bigcap_{\mathcal{R} \in \subseteq_P\text{-}Rep(\mathcal{P}_1 \cup \dots \cup \mathcal{P}_i, \mathcal{T})} \mathcal{R}$. Equivalently, $pind(\mathcal{K}_\succ) = \bigcap_{\mathcal{R} \in \subseteq_P\text{-}Rep(\mathcal{K})} \mathcal{R}$.*

The "unique repairs" of Definition 74, for the case where only a preorder over the facts is available, rely on completing the preorder in all possible ways, yielding prioritizations used to apply notions from Definition 73. Two additional proposals, called *CElect* [22] and $C\pi$ [97], add facts using the logical closure w.r.t. the ontology, thus are typically not subsets of $\mathcal{D}$.

▶ **Definition 74** (Unique repairs based on a preorder [22]). *Let $\mathcal{K}_\succ$ be a prioritized KB (or database with anti-monotone ICs) such that $\succ$ corresponds to a preorder $\trianglelefteq$ over $\mathcal{D}$. For each total preorder $\trianglelefteq'$ extending $\trianglelefteq$, let $\succ'$ correspond to the prioritization induced by $\trianglelefteq'$.*

- *$Elect(\mathcal{K}_\succ) = \bigcap_{\trianglelefteq' \text{ total extension of } \trianglelefteq} nd(\mathcal{K}_{\succ'})$.*
- *$PartialPR(\mathcal{K}_\succ) = \bigcap_{\trianglelefteq' \text{ total extension of } \trianglelefteq} pind(\mathcal{K}_{\succ'})$.*

The main "unique repair" that has been proposed in the general case of an arbitrary priority relation stems from abstract argumentation. Hence, we need to define the *grounded extension* of a SETAF. Given a (SET)AF $F = (Args, \rightsquigarrow)$, recall that the characteristic function of F is defined by $\Gamma_F(A) = \{\alpha \mid \alpha \text{ is defended by } A\}$. Then $E \subseteq Args$ is a *complete extension* iff E is conflict-free and $E = \Gamma_F(E)$, and the *grounded extension* of F is the $\subseteq$-minimal complete extension, or equivalently, the least fixpoint of Γ_F.

▶ **Definition 75** (Unique repairs based on an arbitrary priority relation [30]). *Let $\mathcal{K}_\succ$ be a prioritized KB (or database with anti-monotone ICs).*

- *The* grounded repair *of $\mathcal{K}_\succ$, $grounded(\mathcal{K}_\succ)$, is the grounded extension of the PSETAF $F_{\mathcal{K},\succ}$ associated with $\mathcal{K}_\succ$ (cf. Section 5.2.2.1).*
- *For $X \in \{C, G, P\}$, the analogue of $pind(\mathcal{K}_\succ)$ is $\bigcap_{\mathcal{R} \in X\text{-}Rep(\mathcal{K}_\succ)} \mathcal{R}$.*

Figure 3 shows the inclusion relationships between these types of repairs in the most specific case, when $\succ$ is induced by a prioritization. To obtain the relationships between repairs for preordered datasets, simply remove from the picture those that are not defined in this case.

$$\begin{tikzcd}[column sep=small]
 & \mathit{pind}(\mathcal{K}_{\succ}) = \mathit{PartialPR}(\mathcal{K}_{\succ}) = \bigcap_{\mathcal{R}\in X\text{-}\mathit{Rep}(\mathcal{K}_{\succ})} \mathcal{R} & \\
 & \mathit{grounded}(\mathcal{K}_{\succ}) \arrow[u] & \\
 & \mathit{nd}(\mathcal{K}_{\succ}) = \mathit{Elect}(\mathcal{K}_{\succ}) \arrow[u] & \\
\ell(\mathcal{K}_{\succ}) & & \ell\mathit{nd}(\mathcal{K}_{\succ}) \\
 & \pi(\mathcal{K}_{\succ}) \arrow[ul] \arrow[ur] \arrow[uu] &
\end{tikzcd}$$

- Polynomial computation:
  given the conflict hypergraph and $\succ$, the repair can be computed in polynomial time
  $\Longrightarrow$ holds for all but $\mathit{pind}(\mathcal{K}_{\succ})$
- Complete w.r.t. isolated vertices:
  contains all facts that do not belong to any conflict (i.e., includes $\mathcal{R}_{\cap} = \bigcap_{\mathcal{R}\in S\text{-}\mathit{Rep}(\mathcal{K})} \mathcal{R}$)
  $\Longrightarrow$ holds for all but $\pi(\mathcal{K}_{\succ})$, $\ell(\mathcal{K}_{\succ})$
- Sound w.r.t. prioritized intersection:
  included in the intersection of optimal repairs
  $\Longrightarrow$ holds for all but $\ell(\mathcal{K}_{\succ})$, $\ell\mathit{nd}(\mathcal{K}_{\succ})$

Figure 3 (left) Relationships between "unique repairs" when $\succ$ is induced by a prioritization of the dataset [24, 22, 30]. An arrow $X \to Y$ means that $X \subseteq Y$. (right) Relevant properties [24, 30].

Finally, for arbitrary priority relations, it holds that $\mathit{grounded}(\mathcal{K}_{\succ}) \subseteq \bigcap_{\mathcal{R}\in P\text{-}\mathit{Rep}(\mathcal{K}_{\succ})} \mathcal{R} \subseteq \bigcap_{\mathcal{R}\in G\text{-}\mathit{Rep}(\mathcal{K}_{\succ})} \mathcal{R} \subseteq \bigcap_{\mathcal{R}\in C\text{-}\mathit{Rep}(\mathcal{K}_{\succ})} \mathcal{R}$ (the last two inclusions directly come from the relationship between completion-, globally- and Pareto-optimal repairs). Figure 3 also indicates relevant properties these repairs may have. In particular, they can be computed in polynomial time when the conflict hypergraph is given, except for the intersections of optimal repairs. We conclude with an example to illustrate the difference between all these "unique repairs".

▶ **Example 76.** Assume that $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ has the conflict graph depicted below and let $P = \langle \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_4 \rangle$ be the prioritization of $\mathcal{D}$ defined by: $\mathcal{P}_1 = \{\alpha, \beta, \gamma\}$, $\mathcal{P}_2 = \{\delta, \epsilon, \zeta\}$, $\mathcal{P}_3 = \{\eta, \theta, \kappa\}$, $\mathcal{P}_4 = \{\lambda, \mu\}$. Arrows on the conflict graph indicate the priority relation $\succ$ induced by P. The optimal repairs ($\subseteq_P\text{-}\mathit{Rep}(\mathcal{K}_{\succ})$) and "unique repairs" are given below. This example shows that all inclusions given in Figure 3 can be strict and that $\pi(\mathcal{K}_{\succ})$ and $\ell(\mathcal{K}_{\succ})$ do not satisfy Complete w.r.t. isolated vertices, since they do not contain μ, while $\ell(\mathcal{K}_{\succ})$ and $\ell\mathit{nd}(\mathcal{K}_{\succ})$ do not satisfy Sound w.r.t. prioritized intersection, since they contain η which does not belong to the intersection of the optimal repairs (actually, η does not belong to *any* of the optimal repairs).

$$\begin{tikzcd}
\alpha \arrow[d, dash] & \delta \arrow[d, dash] \arrow[dr] & & \\
\beta & \epsilon \arrow[r] & \eta \arrow[r] & \lambda \\
\gamma \arrow[r] \arrow[drr] & \zeta \arrow[r] & \theta & \mu \\
 & & \kappa &
\end{tikzcd}$$

$$\begin{aligned}
\subseteq_P\text{-}\mathit{Rep}(\mathcal{K}_{\succ}) = \{ & \\
&\{\alpha, \gamma, \delta, \theta, \lambda, \mu\}, \\
&\{\alpha, \gamma, \epsilon, \theta, \lambda, \mu\}, \\
&\{\beta, \gamma, \delta, \theta, \lambda, \mu\}, \\
&\{\beta, \gamma, \epsilon, \theta, \lambda, \mu\}\}
\end{aligned}$$

$$\begin{aligned}
\mathit{pind}(\mathcal{K}_{\succ}) &= \{\gamma, \theta, \lambda, \mu\} \\
\mathit{grounded}(\mathcal{K}_{\succ}) &= \{\gamma, \theta, \mu\} \\
\mathit{nd}(\mathcal{K}_{\succ}) &= \{\gamma, \mu\} \\
\pi(\mathcal{K}_{\succ}) &= \emptyset \\
\ell(\mathcal{K}_{\succ}) &= \{\eta, \theta, \kappa\} \\
\ell\mathit{nd}(\mathcal{K}_{\succ}) &= \{\gamma, \eta, \theta, \mu\}
\end{aligned}$$

▶ Remark 77. It is easy to define variants of $\pi(\mathcal{K}_{\succ})$ and $\ell(\mathcal{K}_{\succ})$ that satisfy Complete w.r.t. isolated vertices: simply start by modifying the prioritization to add a level of highest priority that contains all facts that do not belong to any conflict.

5.4.3 Soft repairs for weighted constraints

In the database setting, *soft constraints*, which are expected to generally hold in the dataset but not necessarily in a perfect manner, have been considered. In this context, weights are used to represent the cost of deleting some fact or violating some constraint [60].

▶ **Definition 78** (Soft repairs of a database with anti-monotone constraints [60]). *Let $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ be a database where $\mathcal{T}$ is a set of anti-monotone ICs and let $w : \mathcal{D} \cup \mathcal{T} \mapsto \mathbb{N} \cup \{\infty\}$. A* soft repair *of $\mathcal{D}$ w.r.t. $\mathcal{T}$ is a dataset $\mathcal{R} \subseteq \mathcal{D}$ such that*

$$\mathsf{cost}^w_{\mathcal{K}}(\mathcal{R}) = \Sigma_{\alpha \in \mathcal{D} \setminus \mathcal{R}} w(\alpha) + \Sigma_{\tau \in \mathcal{T}} w(\tau) |\mathit{Conf}(\mathcal{R}, \{\tau\})|$$

is minimal, i.e., there is no $\mathcal{R}' \subseteq \mathcal{D}$ such that $\mathsf{cost}^w_{\mathcal{K}}(\mathcal{R}') < \mathsf{cost}^w_{\mathcal{K}}(\mathcal{R})$.

Soft repairs are not dataset repairs since they may be $\mathcal{T}$-inconsistent. However, they generalize $\leq_w$-optimal repairs.

▶ **Proposition 79** ([60]). *If w is a function that assigns a positive weight to each fact of $\mathcal{D}$, then $\leq_w$-Rep$(\mathcal{D}, \mathcal{T})$ is equal to the set of soft repairs of $\mathcal{D}$ w.r.t. $\mathcal{T}$ using w' where $w'(\tau) = \infty$ for each $\tau \in \mathcal{T}$ and $w'(\alpha) = w(\alpha)$ for each $\alpha \in \mathcal{D}$.*

The generalization of soft constraints to DL KBs does not give rise to "soft repairs" but rather to "soft models", since the optimization is done at the level of the interpretation instead of the dataset [36]. In this context, the cost of an interpretation $\mathcal{I}$ is $\mathsf{cost}^w_{\mathcal{K}}(\mathcal{I}) = \Sigma_{\alpha \in \mathcal{D}, \mathcal{I} \not\models \alpha} w(\alpha) + \Sigma_{\tau \in \mathcal{T}} w(\tau) |\mathsf{vio}_\tau(\mathcal{I})|$ with $\mathsf{vio}_{C \sqsubseteq D}(\mathcal{I}) = (C \sqcap \neg D)^{\mathcal{I}}$ for every concept inclusion $C \sqsubseteq D$. Interpretations with minimal (or bounded) cost are then used to define variants of the certain and possible semantics.

6 Computational complexity of reasoning with (preferred) repairs

In this section, we present some complexity results on the following decision problems:

- *Repair checking* takes as input a database or KB $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ and a dataset $\mathcal{R}$, and outputs "yes" if $\mathcal{R}$ is a (preferred) repair of $\mathcal{K}$, and "no" otherwise;
- *BCQ entailment under CQA (resp. intersection, brave) semantics* takes as input a database or KB $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ and a BCQ q and output "yes" if $\mathcal{K}$ entails q under the (preferred repair-based) CQA (resp. intersection, brave) semantics, and "no" otherwise.

We focus on these three semantics since CQA is the most studied inconsistency-tolerant semantics and intersection and brave provide the most and least cautious under- and over-approximations (cf. Figure 2). The complexity of reasoning under most of the other inconsistency-tolerant semantics mentioned in Section 4.1 is summarized in [28, Fig. 3] for DL-Lite KBs, with more details in [29]. Other relevant tasks that received significant attention include, e.g., the problem of enumerating the (preferred) repairs, counting them, or deciding whether there is a unique preferred repair [93, 30, 101].

6.1 The case of standard repairs

We first present the complexity of reasoning with standard dataset repairs.

6.1.1 Repair checking

Table 4 shows the data and combined complexity of repair checking (in the database case we focus on Δ-repairs but the complexity of $\subseteq$- and $\supseteq$-repair checking has also been studied [4, 130]). In Table 4, results for Datalog$^\pm$ and database ICs are mainly taken from [107, Table 3] and [11, Table 3], respectively, where the interested reader can find results for different complexity measures, additional languages, and proofs or pointers to the literature. Propositions 80, 81 and 82 state the results for DLs and refined data complexity results for Datalog$^\pm$. Most of these results are known but they are not always explicitly stated in the literature, and the proofs illustrate the use of classical algorithms.

Table 4 Complexity of repair checking. All non-"in" entries are completeness results.

		Data complexity		Combined complexity	
DLs $S\text{-}Rep(\mathcal{K})$	DL-Lite$_\mathcal{R}$	in L	Prop. 82	NL	Prop. 82
	DL-Lite$_{\mathcal{R},\sqcap}$	in L		P	Prop. 81
	$\mathcal{EL}_\bot$	P	Prop. 81	P	
	$\mathcal{ALC}$	DP	Prop. 80	Exp	Prop. 80
	$\mathcal{SHIQ}$	DP		Exp	
Datalog$^\pm$ $S\text{-}Rep(\mathcal{K})$	$\mathsf{L}_\bot$	in L	Prop. 82	PSpace	[107, Tab. 3]
	$\mathsf{A}_\bot$	in L		DExp	
	$\mathsf{G}_\bot$	P	Prop. 81	2Exp	
	$\mathsf{S}_\bot$	in L	Prop. 82	Exp	
	$\mathsf{F}_\bot$	P	Prop. 81	Exp	
ICs $S\text{-}Rep(\mathcal{K})$	FD	in L	[11, Tab. 3]	in L	[11, Tab. 3]
	DC	in L		DP	
ICs $S\text{-}Rep_\Delta(\mathcal{K})$	full TGD	P	[11, Tab. 3]	DP	[11, Tab. 3]
	UC	coNP		Π^p_2	
	TGD	coNP		Π^p_3	

Algorithms for repair checking. The complexity upper bounds are shown using two classical algorithms for repair checking. Let $\mathcal{K} = \langle\mathcal{D}, \mathcal{T}\rangle$ be a KB or a database and $\mathcal{R}$ be a dataset (repair candidate). The *non-deterministic algorithm* is as follows:

1. check that $\langle\mathcal{R}, \mathcal{T}\rangle \not\models \bot$,
2. check that $\mathcal{R}$ minimally differs from $\mathcal{D}$ by considering the complement (co-2): to show that $\mathcal{R}$ does not minimally differ from $\mathcal{D}$, guess $\mathcal{R}'$ such that $\langle\mathcal{R}', \mathcal{T}\rangle \not\models \bot$ and $\mathcal{R}'\Delta\mathcal{D} \subsetneq \mathcal{R}\Delta\mathcal{D}$.

The *deterministic algorithm* works for KBs or databases with anti-monotone ICs, since in these cases, $\mathcal{R}$ must be a subset of $\mathcal{D}$ and is not maximal iff there is an immediate superset of $\mathcal{R}$ in $\mathcal{D}$ which is $\mathcal{T}$-consistent:

1. check that $\mathcal{R} \subseteq \mathcal{D}$ and that $\langle\mathcal{R}, \mathcal{T}\rangle \not\models \bot$,
2. check that $\mathcal{R}$ is maximal by checking that for each $\alpha \in \mathcal{D} \setminus \mathcal{R}$, $\langle\mathcal{R} \cup \{\alpha\}, \mathcal{T}\rangle \models \bot$.

Basic lower bounds. Complexity lower bounds for consistency checking (Table 3) immediately transfer to repair checking since $\mathcal{D}$ is a repair of $\langle\mathcal{D}, \mathcal{T}\rangle$ iff $\mathcal{D}$ is $\mathcal{T}$-consistent.

▶ **Proposition 80** ($\mathcal{ALC}$ **and** $\mathcal{SHIQ}$). *For $\mathcal{ALC}$ and $\mathcal{SHIQ}$, repair checking is DP-complete w.r.t. data complexity, and* Exp*-complete w.r.t. combined complexity.*

Proof. The combined complexity upper bound follows from the deterministic algorithm for repair checking and the fact that consistency checking is in Exp. The matching lower bound comes from the Exp-hardness of consistency checking.

For the data complexity upper bound, we use the non-deterministic algorithm. Since consistency checking is in NP, step 1 can be done with a call to an NP oracle and step 2 with a call to a coNP oracle: indeed, (co-2) is in NP since we can guess a certificate of polynomial size that consists of (i) $\mathcal{R}' \subseteq \mathcal{D}$ and (ii) a certificate that $\langle\mathcal{R}', \mathcal{T}\rangle \not\models \bot$, and check this certificate in P. We obtain the DP lower bound via a reduction from the DP-complete problem SAT-UNSAT: given a pair (φ_1, φ_2) of propositional formulas, decide whether φ_1 is satisfiable and φ_2 is unsatisfiable. We assume that φ_1 and φ_2 are sets of clauses such that each clause has exactly 2 positive and 2 negative literals, and can use truth constants true (1) and false (0) instead of variables (this is w.l.o.g. since 2+2SAT is NP-complete [128]). Let φ_1 and φ_2 be two sets of clauses $\{c^1_1, \dots, c^1_{m_1}\}$ and $\{c^2_1, \dots, c^2_{m_2}\}$ over variables $x^1_1, \dots, x^1_{n_1}$ and $x^2_1, \dots, x^2_{n_2}$ (and truth values 1, 0), respectively. We define an $\mathcal{ALC}$ KB $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ by:

$$\mathcal{T} = \{V \sqsubseteq T \sqcup F,\;\; T \sqcap F \sqsubseteq \bot,\;\; \exists P_1.F \sqcap \exists P_2.F \sqcap \exists N_1.T \sqcap \exists N_2.T \sqsubseteq F,\;\; \exists Cl.F \sqsubseteq F\}$$

and $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 \cup \{F(0), T(1)\}$ with:

$$\begin{aligned}\mathcal{D}_i =& \{T(\varphi_i)\} \cup \{Cl(\varphi_i, c^i_k) \mid 1 \leq k \leq m_i\} \cup \{V(x^i_j) \mid 1 \leq j \leq n_i\} \cup \\ & \{P_\ell(c^i_k, x^i_j) \mid \ell \in \{1,2\}, x^i_j \text{ is the } \ell^{th} \text{ positive literal of } c^i_k\} \cup \\ & \{N_\ell(c^i_k, x^i_j) \mid \ell \in \{1,2\}, \neg x^i_j \text{ is the } \ell^{th} \text{ negative literal of } c^i_k\}\end{aligned}$$

We claim that $\mathcal{D} \setminus \{T(\varphi_2)\} \in S\text{-}Rep(\mathcal{K})$ iff φ_1 is satisfiable and φ_2 is unsatisfiable. Indeed, it is easy to show that for $i \in \{1,2\}$, $\mathcal{D}_i \cup \{F(0), T(1)\}$ is $\mathcal{T}$-consistent iff φ_i is satisfiable, since there is a one-to-one correspondence between models of $\langle \mathcal{D}_i \cup \{F(0), T(1)\}, \mathcal{T} \rangle$, which assign each x^i_j to T or F in a way that does not enforce any c^i_k to be assigned to F, and valuations of the x^i_j's that satisfy φ_i. Moreover, $\mathcal{D}_i \cup \{F(0), T(1)\} \setminus \{T(\varphi_i)\}$ is $\mathcal{T}$-consistent, since in this case there is a model that assigns all individuals but 1 to F. Since $\mathcal{D}_1$ and $\mathcal{D}_2$ are disjoint, the claim follows easily. ◀

▶ **Proposition 81** (P bounds). *For all ontology languages from Table 4 but $\mathcal{ALC}$ and $\mathcal{SHIQ}$, repair checking is in P w.r.t. data complexity. Matching lower bounds hold for $\mathcal{EL}_\bot$, $\mathsf{G}_\bot$ and $\mathsf{F}_\bot$. For DL-Lite$_\mathcal{R}$, DL-Lite$_{\mathcal{R},\sqcap}$ and $\mathcal{EL}_\bot$, repair checking is also in P w.r.t. combined complexity. Matching lower bounds hold for DL-Lite$_{\mathcal{R},\sqcap}$ and $\mathcal{EL}_\bot$.*

Proof. For the upper bounds, we use the deterministic algorithm and the fact that consistency checking is in P for the considered languages and complexity measures. The lower bounds come from P-hardness of consistency checking in data or combined complexity. ◀

▶ **Proposition 82** (Below P). *Repair checking is in L w.r.t. data complexity for DL-Lite$_\mathcal{R}$, DL-Lite$_{\mathcal{R},\sqcap}$, $\mathsf{L}_\bot$, $\mathsf{A}_\bot$ and $\mathsf{S}_\bot$, and NL-complete w.r.t. combined complexity for DL-Lite$_\mathcal{R}$.*

Proof. For the data complexity upper bound, the DL-Lite$_\mathcal{R}$ and DL-Lite$_{\mathcal{R},\sqcap}$ cases follow from the $\mathsf{S}_\bot$ case (cf. Figure 1). For $\mathsf{L}_\bot$, $\mathsf{A}_\bot$ and $\mathsf{S}_\bot$, we use a data-independent reduction from repair checking in any of these languages to repair checking for a database with DCs, which is in L w.r.t. data complexity (cf. Table 4). Indeed, by Proposition 7, since L, A and S are UCQ-rewritable (Fact 5), for every ontology $\mathcal{T}$, there exists a BUCQ $q^\mathcal{T}_\bot$ such that for every dataset $\mathcal{B}$, $\langle \mathcal{B}, \mathcal{T} \rangle \models \bot$ iff $\mathcal{I}_\mathcal{B} \models q^\mathcal{T}_\bot$ iff $\langle \mathcal{B}, \mathcal{T}_{\mathsf{DC}} \rangle \models \bot$, where $\mathcal{T}_{\mathsf{DC}}$ is the set of DCs that contains $q \rightarrow \bot$ for each BCQ q in $q^\mathcal{T}_\bot$. Hence, $\mathcal{R} \in S\text{-}Rep(\mathcal{D}, \mathcal{T})$ iff $\mathcal{R} \in S\text{-}Rep(\mathcal{D}, \mathcal{T}_{\mathsf{DC}})$.

Regarding the combined complexity for DL-Lite$_\mathcal{R}$, the lower bound comes from the NL-hardness of consistency checking. For the upper bound, observe that since consistency checking is in NL w.r.t. combined complexity, and NL = coNL, inconsistency checking is also in NL. The deterministic algorithm for repair checking can thus be adapted as follows: guess a certificate that consists of (i) a certificate that $\langle \mathcal{R}, \mathcal{T} \rangle \not\models \bot$ and (ii) certificates that $\langle \mathcal{R} \cup \{\alpha\}, \mathcal{T} \rangle \models \bot$ for each $\alpha \in \mathcal{D} \setminus \mathcal{R}$. The certificate is of polynomial size and can be verified by a deterministic logarithmic-space bounded Turing machine that has an additional read-only read-once input tape, since each certificate from (i)-(ii) is of polynomial size and verifiable by such a machine, and all certificates can be verified independently. ◀

Table 5 Complexity of BCQ entailment under CQA, intersection, and brave semantics. All non-“in” entries are completeness results (except undecidability results).

		Data complexity			Combined complexity		
		CQA	Int.	brave	CQA	Int.	brave
DLs $S\text{-}Rep(\mathcal{K})$	DL-Lite$_\mathcal{R}$	coNP	in AC^0	in AC^0	Π_2^p	NP	NP
	DL-Lite$_{\mathcal{R},\sqcap}$	coNP	in AC^0	in AC^0	Π_2^p	Θ_2^p	NP
	$\mathcal{EL}_\bot$	coNP	coNP	NP	Π_2^p	Θ_2^p	NP
	$\mathcal{ALC}$	Π_2^p	Π_2^p	Σ_2^p	Exp	Exp	Exp
	$\mathcal{SHIQ}$	Π_2^p	Π_2^p	Σ_2^p	2Exp	2Exp	2Exp
Datalog$^\pm$ $S\text{-}Rep(\mathcal{K})$	$\mathsf{L}_\bot$	coNP	in AC^0	in AC^0	PSpace	PSpace	PSpace
	$\mathsf{A}_\bot$	coNP	in AC^0	in AC^0	$\mathrm{P}^{\mathrm{NExp}}$	$\mathrm{P}^{\mathrm{NExp}}$	$\mathrm{P}^{\mathrm{NExp}}$
	$\mathsf{G}_\bot$	coNP	coNP	NP	2Exp	2Exp	2Exp
	$\mathsf{S}_\bot$	coNP	in AC^0	in AC^0	Exp	Exp	Exp
	$\mathsf{F}_\bot$	coNP	coNP	NP	Exp	Exp	Exp
ICs $S\text{-}Rep(\mathcal{K})$	FD	coNP	in AC^0	in AC^0	Π_2^p		
	DC	coNP	in AC^0	in AC^0	Π_2^p		
ICs $S\text{-}Rep_\Delta(\mathcal{K})$	full TGD	coNP	coNP	P	Exp		
	UC	Π_2^p	Π_2^p	Σ_2^p	Π_2^{Exp}		
	TGD	undec.			undec.		

6.1.2 BCQ entailment under CQA, intersection, and brave semantics

Table 5 shows the data and combined complexity of deciding BCQ entailment under the CQA, intersection, and brave semantics. The results given in Table 5 are taken from:

- [11, Tab. 1] for the CQA semantics over databases with ICs,
- [104, Tab. 2, 3 & 6] for the CQA and intersection semantics over Datalog$^\pm$ KBs,
- [29, Fig. 2, 3 & 4] for DL-Lite$_\mathcal{R}$, $\mathcal{EL}_\bot$ and $\mathcal{ALC}$, [41, Th. 3 & Fig. 1] for DL-Lite$_{\mathcal{R},\sqcap}$, and [126, Fig. 2] for $\mathcal{SHIQ}$ (plus straightforward upper bound for brave).

The complexity of BCQ entailment under the intersection and brave semantics over databases and brave semantics in Datalog$^\pm$ has not been systematically investigated. In the remaining of this section, we present some classical algorithms for BCQ entailment under the three semantics and provide proofs for the few new results on Datalog$^\pm$ and databases (for which we consider data complexity, on which we focus in the next section on preferred repairs).

Classical algorithms. Most upper complexity bounds are shown using the *non-deterministic* versions of Algorithms 1, 2 and 3, where one guesses a repair that does not entail the query to show that it does not hold under CQA semantics, a repair that entails the query to show that it holds under brave semantics, and the appropriate $\mathcal{B}$ and $\mathcal{R}_1, \dots, \mathcal{R}_n$ to show that the query is not entailed under intersection semantics. Note that in the case of brave semantics, since every $\mathcal{T}$-consistent subset of $\mathcal{D}$ can be extended into a standard repair, it is sufficient to check that $\langle \mathcal{R}, \mathcal{T} \rangle \not\models \bot$ instead of $\mathcal{R} \in S\text{-}Rep(\mathcal{K})$ in Algorithm 2 when $X = S$. These algorithms can be adapted for semantics based on Δ-repairs of databases with UCs, by considering subsets of $Facts_\mathcal{D}^\mathcal{T}$ (whose size is exponential w.r.t. the maximal arity of the predicates in $\mathcal{T}$ and polynomial w.r.t. the size of $\mathcal{D}$) instead of subsets of $\mathcal{D}$. This does not extend to ICs with existential quantifiers in the head such as TGDs, since in this case repairs may use constants that do not occur in $\mathcal{K}$.

Algorithm 1 X-CQA.

Input: a KB or database with anti-monotone ICs $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ and a BCQ q
Output: true iff $\mathcal{K}$ entails q under CQA semantics based on $X\text{-}Rep(\mathcal{K})$

```
for R ⊆ D do
  if R ∈ X-Rep(K) and ⟨R, T⟩ ⊭ q then
    return false
return true
```

Algorithm 2 X-brave.

Input: a KB or database with anti-monotone ICs $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ and a BCQ q
Output: true iff $\mathcal{K}$ entails q under brave semantics based on $X\text{-}Rep(\mathcal{K})$

```
for R ⊆ D do
  if R ∈ X-Rep(K) and ⟨R, T⟩ ⊨ q then
    return true
return false
```

When the conflicts can be computed efficiently, Algorithm 4 provides an efficient way of deciding BCQ entailment under intersection semantics.

Basic lower bounds. Complexity lower bounds for BCQ entailment under the classical semantics (Table 2) immediately transfer to BCQ entailment under CQA, intersection and brave semantics. Indeed, the hardness results for BCQ entailment under the classical semantics hold in the case where $\mathcal{D}$ is assumed to be $\mathcal{T}$-consistent, and in this case the classical, CQA, intersection and brave semantics all coincide.

AC^0 upper bounds. The AC^0 upper bounds in Table 5 follow from the following result [41].

▶ **Proposition 83** ([41, Theorems 1, 2, 3]). *If CQ answering under $\mathcal{L}$ is FO-rewritable and for every ontology $\mathcal{T}$ in $\mathcal{L}$ and BCQ q, there exist ℓ and m such that for every $\mathcal{D}$, (i) every minimal $\mathcal{T}$-consistent subset $\mathcal{S}$ of $\mathcal{D}$ such that $\langle \mathcal{S}, \mathcal{T} \rangle \models q$ has cardinality at most ℓ and (ii) every conflict of $\langle \mathcal{D}, \mathcal{T} \rangle$ has cardinality at most m, then CQ answering under the intersection and brave semantics in $\mathcal{L}$ is FO-rewritable: for every ontology $\mathcal{T}$ and BCQ q, there exists an FO-sentence $\varphi^{\cap}_{\mathcal{T},q}$ (resp. $\varphi^{brave}_{\mathcal{T},q}$) such that for every $\mathcal{D}$, q is entailed by $\langle \mathcal{D}, \mathcal{T} \rangle$ under intersection (resp. brave) semantics iff $\mathcal{I}_{\mathcal{D}} \models \varphi^{\cap}_{\mathcal{T},q}$ (resp. $\mathcal{I}_{\mathcal{D}} \models \varphi^{brave}_{\mathcal{T},q}$).*

▶ **Corollary 84** (AC^0 bounds). *BCQ entailment under intersection and brave semantics is in AC^0 w.r.t. data complexity for DL-Lite$_\mathcal{R}$, DL-Lite$_{\mathcal{R},\sqcap}$, $\mathsf{L}_\perp$, $\mathsf{A}_\perp$, $\mathsf{S}_\perp$, FDs and DCs.*

Algorithm 3 X-intersection.

Input: a KB or database with anti-monotone ICs $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ and a BCQ q
Output: true iff $\mathcal{K}$ entails q under intersection semantics based on $X\text{-}Rep(\mathcal{K})$

```
for B ⊆ D with B = {α_1, ..., α_n} do
  for R_1 ⊆ D, ..., R_n ⊆ D such that α_i ∉ R_i (1 ≤ i ≤ n) do
    if R_i ∈ X-Rep(K) for 1 ≤ i ≤ n and ⟨D \ B, T⟩ ⊭ q then
      return false
return true
```

Algorithm 4 Intersection semantics from conflicts.

Input: a KB or database with anti-monotone ICs $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ and a BCQ q
Output: true iff $\mathcal{K}$ entails q under (plain) intersection semantics

if $\langle \mathcal{T}, \mathcal{D} \setminus \bigcup_{\mathcal{C} \in Conf(\mathcal{K})} \mathcal{C} \rangle \models q$ **then**
  return true
return false

Proof. These languages satisfy the conditions of Proposition 83. Indeed, CQ answering is UCQ-rewritable (cf. Fact 5), hence FO-rewritable and:

(i) for DCs (hence also FDs) the size of minimal $\mathcal{T}$-consistent subsets $\mathcal{S}$ of $\mathcal{D}$ such that $\langle \mathcal{S}, \mathcal{T} \rangle \models q$ is bounded by $|q|$, and for $\mathsf{L}_\bot$, $\mathsf{A}_\bot$ and $\mathsf{S}_\bot$ (hence also DL-Lite$_\mathcal{R}$ and DL-Lite$_{\mathcal{R},\sqcap}$), it is bounded by the maximal number of atoms in some BCQ in $\mathsf{Rew}(q, \mathcal{T})$;

(ii) for DCs (hence also FDs) the size of conflicts is bounded by the maximal number of atoms in some DC body, and for $\mathsf{L}_\bot$, $\mathsf{A}_\bot$ and $\mathsf{S}_\bot$ (hence also DL-Lite$_\mathcal{R}$ and DL-Lite$_{\mathcal{R},\sqcap}$), by Proposition 7, for every ontology $\mathcal{T}$, there exists a BUCQ $q_\bot^\mathcal{T}$ such that for every dataset $\mathcal{D}$, $\langle \mathcal{D}, \mathcal{T} \rangle \models \bot$ iff $\mathcal{I}_\mathcal{D} \models q_\bot^\mathcal{T}$ so the size of conflicts is bounded by the maximal number of atoms in some BCQ of $q_\bot^\mathcal{T}$. ◀

Example 85 illustrates why Proposition 83 does not apply to the other ontology languages considered in Table 5: the size of the conflicts cannot be bounded independently from $\mathcal{D}$.

▶ **Example 85.** Let $\mathcal{T} = \{R(x, y) \wedge A(x) \rightarrow A(y),\ A(x) \wedge B(x) \rightarrow \bot\}$, which belongs to $\mathsf{F}_\bot$, $\mathsf{G}_\bot$ and $\mathcal{EL}_\bot$. For every $n \geq 1$, we can build a dataset $\mathcal{D}$ of size $n + 2$ which is a conflict (of itself) w.r.t. $\mathcal{T}$: $\mathcal{D} = \{A(a_0), R(a_0, a_1), \dots, R(a_{n-1}, a_n), B(a_n)\}$.

Note that if we interpret $\mathcal{T}$ as a set of UCs in Example 85, it also shows that the size of the conflicts cannot be bounded independently from $\mathcal{D}$ for databases with UCs. It is actually undecidable to determine whether there exists $\ell \in \mathbb{N}$ such that for every dataset $\mathcal{D}$, $\max_{\mathcal{C} \in Conf(\mathcal{D}, \mathcal{T})} |\mathcal{C}| \leq \ell$ when $\mathcal{T}$ is a set of full TGDs [32, Theorem 5][3].

Brave semantics in Datalog$^\pm$. For brave semantics and Datalog$^\pm$, the AC^0 upper bounds come from Corollary 84 and the other bounds in Table 5 follows easily from known results:

▶ **Proposition 86.** *BCQ entailment under brave semantics is PSPACE-complete for $\mathsf{L}_\bot$, P^{NExp}-complete for $\mathsf{A}_\bot$, 2EXP-complete for $\mathsf{G}_\bot$, and EXP-complete for $\mathsf{S}_\bot$ and $\mathsf{F}_\bot$ w.r.t. combined complexity, and NP-complete for $\mathsf{G}_\bot$ and $\mathsf{F}_\bot$ w.r.t. data complexity.*

Proof. The NP data complexity upper bounds follow from the non-deterministic version of Algorithm 2, since both repair checking and BCQ entailment are in P w.r.t. data complexity for $\mathsf{G}_\bot$ and $\mathsf{F}_\bot$. For the matching lower bound, we remark that the reduction used in the proof of NP-hardness of BCQ entailment under brave semantics in $\mathcal{EL}_\bot$ [29, Theorem 29] builds an ontology without existential on the right-hand side, which is thus in $\mathsf{G}_\bot$ and $\mathsf{F}_\bot$.

The 2EXP and EXP combined complexity upper bounds for $\mathsf{G}_\bot$, $\mathsf{S}_\bot$ and $\mathsf{F}_\bot$ follow from Algorithm 2 (since repair checking and BCQ entailment are both in 2EXP or EXP, respectively) and the lower bounds from the respective combined complexity of BCQ entailment.

The PSPACE combined complexity upper bound for $\mathsf{L}_\bot$ follows from the non-deterministic version of Algorithm 2, since repair checking and BCQ entailment are in PSPACE for $\mathsf{L}_\bot$ and NPSPACE = PSPACE. The lower bound comes from that of BCQ entailment in $\mathsf{L}_\bot$.

[3] Theorem 5 in [32] is stated for universal constraints but the proof uses only full TGDs.

The $\mathrm{P}^{\mathrm{NExp}}$ combined complexity upper bound for $\mathsf{A}_\bot$ follows from the non-deterministic version of Algorithm 2, since repair checking is in DExp and BCQ entailment in NExp for $\mathsf{A}_\bot$, so that the "check" phase can be done in $\mathrm{P}^{\mathrm{NExp}}$, and the fact that $\mathrm{NP}^{\mathrm{NExp}} = \mathrm{P}^{\mathrm{NExp}}$ [89, 104]. The lower bound follows from the reduction from the $\mathrm{P}^{\mathrm{NExp}}$-hard extended exponential tiling problem used to show the lower bound for intersection semantics in [104, Theorem 6.2]: given an extended tiling system $\mathcal{E}$, a KB $\langle \mathcal{D}_\mathcal{E}, \mathcal{T}_\mathcal{E}\rangle$ is built such that $\mathcal{E}$ is valid iff $\mathsf{Yes}()$ is entailed by $\langle \mathcal{D}_\mathcal{E}, \mathcal{T}_\mathcal{E}\rangle$ under intersection semantics, and since (i) $\mathsf{Yes}() \in \mathcal{D}_\mathcal{E}$, (ii) $\mathsf{Yes}()$ does not occur in the body of any TGD and (iii) the only NC that contains $\mathsf{Yes}()$ is $\phi(\vec{x}) \wedge \mathsf{Yes}() \wedge \mathsf{No}() \rightarrow \bot$, it follows that $\mathcal{E}$ is valid iff $q = \exists \vec{x} \phi(\vec{x}) \wedge \mathsf{No}()$ is not entailed under brave semantics. ◀

Data complexity of brave and intersection semantics in databases. For brave and intersection BCQ entailment over databases, the AC^0 upper bounds come from Corollary 84, the data complexity results for UCs are taken from [32, Tab. 1] (they follow from the proof of [32, Theorem 3]), and the other results given in Table 5 are proved below.

▶ **Proposition 87** (Full TGD). *For the IC language of full TGDs and data complexity, BCQ entailment under brave semantics is P-complete and BCQ entailment under intersection semantics is coNP-complete.*

Proof. For brave semantics, we remark that for every database $\mathcal{K} = \langle \mathcal{D}, \mathcal{T}\rangle$ such that $\mathcal{T}$ is a set of full TGDs, for every BCQ q, q is entailed by $\mathcal{K}$ under brave semantics iff q is (classically) entailed by the F (a.k.a. Datalog) KB $\mathcal{K}_\mathsf{F} = \langle \mathcal{D}, \mathcal{T}\rangle$. Indeed, it is known that for full TGDs, there exists a unique $\supseteq$-repair $\mathcal{R}$, which can be computed in polynomial time w.r.t. the size of the dataset via the chase procedure [130], and we can show that every $\mathcal{R}' \in \mathit{S\text{-}Rep}_\Delta(\mathcal{K})$ is such that $\mathcal{R}' \subseteq \mathcal{R}$. Since BCQ evaluation is monotone, it follows that there exists $\mathcal{R}' \in \mathit{S\text{-}Rep}_\Delta(\mathcal{K})$ such that q holds in $\mathcal{R}'$ iff q holds in $\mathcal{R}$. Finally, since $\mathcal{K}_\mathsf{F} \models q$ iff q holds in $\mathcal{R}$ (by the fixpoint semantics of Datalog, cf. [1, Sec. 12.3]), we obtain that q is entailed by $\mathcal{K}$ under brave semantics iff q is classically entailed by $\mathcal{K}_\mathsf{F}$. The upper bound then follows from the fact that BCQ entailment in F is in P and the lower bound from the reduction that shows P-hardness of BCQ entailment in F [70, Th. 4.4].

For intersection semantics, the coNP upper bound follows from the non-deterministic version of Algorithm 3 (adapted for databases with UCs), since repair checking and BCQ entailment are in P and the repair size is polynomially bounded w.r.t. data complexity. For the lower bound, we use a reduction from UNSAT. Let φ be a conjunction of clauses $c_1, \dots, c_m$ over variables $x_1, \dots, x_n$ and define $\mathcal{K} = \langle \mathcal{D}, \mathcal{T}\rangle$ by:

$$\begin{aligned}\mathcal{T} = \{&\mathsf{Next}(z_1, z_2) \wedge \mathsf{Check}(z_2) \wedge \mathsf{Pos}(z_2, x_1) \wedge \mathsf{True}(x_1) \rightarrow \mathsf{Check}(z_1),\\ &\mathsf{Next}(z_1, z_2) \wedge \mathsf{Check}(z_2) \wedge \mathsf{Neg}(z_2, x_1) \wedge \mathsf{False}(x_1) \rightarrow \mathsf{Check}(z_1),\\ &\mathsf{Check}(z) \wedge \mathsf{Unsat}(z) \rightarrow \mathsf{B}(),\ \ \mathsf{True}(x) \wedge \mathsf{False}(x) \rightarrow \mathsf{B}()\}\\ \mathcal{D} = \{&\mathsf{Unsat}(c_0)\} \cup \{\mathsf{Check}(c_m)\} \cup \{\mathsf{Next}(c_{j-1}, c_j) \mid 1 \leq j \leq m\} \cup \{\mathsf{Pos}(c_j, x_i) \mid x_i \in c_j\} \cup\\ &\{\mathsf{Neg}(c_j, x_i) \mid \neg x_i \in c_j\} \cup \{\mathsf{True}(x_i), \mathsf{False}(x_i) \mid 1 \leq i \leq n\}\end{aligned}$$

We claim that $\mathsf{Unsat}(c_0)$ is entailed by $\mathcal{K}$ under intersection semantics iff φ is unsatisfiable. Indeed, if there exists a valuation ν of $x_1, \dots, x_n$ that satisfies φ, the following dataset is a Δ-repair of $\mathcal{D}$ w.r.t. $\mathcal{T}$, so $\mathsf{Unsat}(c_0)$ does not hold under intersection semantics.

$$\begin{aligned}\mathcal{R}_\nu = &\{\mathsf{Check}(c_j) \mid 0 \leq j \leq m\} \cup \{\mathsf{Next}(c_{j-1}, c_j) \mid 1 \leq j \leq m\} \cup \{\mathsf{Pos}(c_j, x_i) \mid x_i \in c_j\} \cup\\ &\{\mathsf{Neg}(c_j, x_i) \mid \neg x_i \in c_j\} \cup \{\mathsf{True}(x_i) \mid \nu(x_i) = 1\} \cup \{\mathsf{False}(x_i) \mid \nu(x_i) = 0\}\end{aligned}$$

In the other direction, if there is a Δ-repair $\mathcal{R}$ of $\mathcal{D}$ w.r.t. $\mathcal{T}$ such that $\mathsf{Unsat}(c_0) \notin \mathcal{R}$, then $\mathsf{B}() \notin \mathcal{R}$ and $\mathsf{Check}(c_0) \in \mathcal{R}$ (otherwise $\mathcal{R} \cup \{\mathsf{Unsat}(c_0)\}$ would be $\mathcal{T}$-consistent). Since $\mathsf{Check}(c_0) \in \mathcal{R}$, there must exist x_i such that $\{\mathsf{Next}(c_0, c_1), \mathsf{Check}(c_1), \mathsf{Pos}(c_1, x_i), \mathsf{True}(x_i)\}$ or $\{\mathsf{Next}(c_0, c_1), \mathsf{Check}(c_1), \mathsf{Neg}(c_1, x_i), \mathsf{False}(x_i)\}$ is included in $\mathcal{R}$ (otherwise $\mathcal{R} \setminus \{\mathsf{Check}(c_0)\}$ would be $\mathcal{T}$-consistent), and we can show that for every $1 \leq j \leq m$, there must exist x_i such that either $\{\mathsf{Next}(c_{j-1}, c_j), \mathsf{Check}(c_j), \mathsf{Pos}(c_j, x_i), \mathsf{True}(x_i)\}$ or $\{\mathsf{Next}(c_{j-1}, c_j), \mathsf{Check}(c_j), \mathsf{Neg}(c_j, x_i), \mathsf{False}(x_i)\}$ is included in $\mathcal{R}$. Moreover, since $\mathsf{B}() \notin \mathcal{R}$, for every x_i, either $\mathsf{True}(x_i)$ or $\mathsf{False}(x_i)$ does not belong to $\mathcal{R}$ so the valuation $\nu_{\mathcal{R}}$ defined by $\nu_{\mathcal{R}}(x_i) = 1$ iff $\mathsf{True}(x_i) \in \mathcal{R}$ satisfies all clauses of φ. ◀

6.2 Reasoning with preferred repairs

In this section, we present the impact of using preferred repairs on the computational complexity. Table 6 shows the data complexity of reasoning with X-optimal repairs for $\mathrm{X} \in \{\leq, \leq_w, \leq_P, \subseteq_P, \text{Pareto}, \text{globally}, \text{completion}\}$, for languages for which repairs are $\subseteq$-repairs and consistency checking and BCQ entailment are in P. The lower bounds apply to languages that extend either DL-Lite$_{\mathsf{core}}$, FD, or NC, yielding tight data complexity bounds for all semantics based on optimal repairs for DL-Lite$_{\mathsf{core}}$, DL-Lite$_{\mathcal{R}}$, DL-Lite$_{\mathcal{R},\sqcap}$, $\mathsf{L}_\bot$, $\mathsf{A}_\bot$, $\mathsf{G}_\bot$, $\mathsf{S}_\bot$, $\mathsf{F}_\bot$, FD and DC. Similarly, Table 7 shows the data complexity of reasoning with preference program-based prioritized KBs, for languages for which repairs are $\subseteq$-repairs and consistency checking and BCQ entailment are in P, but since such repairs have not been considered in as many settings as the others, we only report the lower bounds for the languages considered in the original paper [50]. For the results given in Table 6, we present the algorithms that yield the upper bounds [33, 107, 102, 129] and give references or show the lower bounds in Sections 6.2.1 and 6.2.2. We then discuss these results and mention some additional complexity results that do not fall into the setting considered in Table 6 (e.g., on combined complexity or for more expressive languages) in Section 6.2.3.

6.2.1 Repair checking

This section presents the proofs for the results on repair checking given in Table 6.

▶ **Proposition 88.** *For every ontology language or anti-monotone IC language for which consistency checking and BCQ entailment are in P w.r.t. data complexity, the data complexity of repair checking is in coNP for X-optimal repairs for $X \in \{\leq, \leq_w, \leq_P, \textit{globally}\}$ and in P for X-optimal repairs for $X \in \{\subseteq_P, \textit{Pareto}, \textit{completion}\}$. Matching coNP lower bounds hold for every ontology language or IC language that extends DL-Lite$_{\mathsf{core}}$, NC or FD.*

Upper bounds. The upper bounds are obtained via Algorithms 5 to 9, using the non-deterministic versions of Algorithms 5 and 6 where one guesses an appropriate $\mathcal{B} \subseteq \mathcal{D}$ to show that $\mathcal{R} \notin X\text{-}Rep(\mathcal{K})$. The correctness of the algorithms straightforwardly follows from the respective definitions of preferred repairs, except for the completion-optimal repair case, which was shown in [129, Corollary 4].

Lower bounds. The coNP lower bound for globally-optimal repair checking has been shown for a logical theory that contains four FDs [129, Theorem 2] as well as for a DL-Lite$_{\mathsf{core}}$ ontology that contains only axioms that can be written as NCs [30, Theorem 10]. The coNP lower bound for $\leq$-optimal repair checking (hence also for $\leq_w$- and $\leq_P$-optimal repair checking, since $\leq$-optimal repairs can be obtained as special cases of $\leq_w$- or $\leq_P$-optimal repairs) has been shown for a logical theory that contains a single NC [103, Corollary 1].

Table 6 Data complexity of repair checking and BCQ entailment under CQA, intersection, and brave semantics with $\preceq$-optimal repairs, or Pareto-, globally- or completion-optimal repairs. Upper bounds hold for every ontology or anti-monotone IC language for which consistency checking and BCQ entailment are in P. Lower bounds hold for every language that extends either DL-Lite$_{\mathsf{core}}$, FD, or NC. †: Θ_2^{P} if there is a data-independent bound on the weights or number or priority levels.

	Repair checking	CQA	Intersection	Brave
$\leq$-*Rep*$(\mathcal{K})$	coNP	Θ_2^{P}	Θ_2^{P}	Θ_2^{P}
$\leq_w$-*Rep*$(\mathcal{K})$	coNP	Δ_2^{P} †	Δ_2^{P} †	Δ_2^{P} †
$\leq_P$-*Rep*$(\mathcal{K})$	coNP	Δ_2^{P} †	Δ_2^{P} †	Δ_2^{P} †
$\subseteq_P$-*Rep*$(\mathcal{K})$	in P	coNP	coNP	NP
P-*Rep*$(\mathcal{K}_\succ)$	in P	coNP	coNP	NP
C-*Rep*$(\mathcal{K}_\succ)$	in P	coNP	coNP	NP
G-*Rep*$(\mathcal{K}_\succ)$	coNP	Π_2^{P}	Π_2^{P}	Σ_2^{P}

Table 7 Data complexity of repair checking and BCQ entailment under CQA, intersection, and brave semantics with $\Pi_{\trianglelefteq}$-preferred repairs. Upper bounds hold for every ontology or anti-monotone IC language for which consistency checking and BCQ entailment are in P. Lower bounds hold for all Datalog$^{\pm}$ fragments we consider [50] (note that [50] allows for constants in TGDs and NCs). The results for brave semantics are not explicitly given in [50] but both the upper and lower bounds can be obtained by adapting straightforwardly the proofs for the other semantics.

	Repair checking	CQA	Intersection	Brave
$Pr_{\subseteq}$-*Rep*$(\mathcal{K}_\Pi)$	coNP	Π_2^{P}	Π_2^{P}	Σ_2^{P}
$Pr_{\leq}$-*Rep*$(\mathcal{K}_\Pi)$	coNP	Θ_2^{P}	Θ_2^{P}	Θ_2^{P}

Algorithm 5 $\preceq$-optimal repair checking for $\preceq \in \{\leq, \leq_w, \leq_P\}$.

Input: $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ and $\mathcal{R} \subseteq \mathcal{D}$ (and weight function w or prioritization P when relevant)
Output: true iff $\mathcal{R} \in \preceq$-*Rep*$(\mathcal{K})$

```
for B ⊆ D do
    if B is T-consistent and R ≺ B then
        return false
return true
```

Algorithm 6 Globally-optimal repair checking.

Input: $\mathcal{K}_\succ = (\langle \mathcal{D}, \mathcal{T} \rangle, \succ)$ and $\mathcal{R} \subseteq \mathcal{D}$
Output: true iff $\mathcal{R} \in G$-*Rep*$(\mathcal{K}_\succ)$

```
for B ⊆ D such that B ≠ R do
    if B is T-consistent and ∀α ∈ R \ B, ∃β ∈ B \ R such that β ≻ α then
        return false
return true
```

Algorithm 7 $\subseteq_P$-optimal repair checking.

Input: $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ with prioritization $P = \langle \mathcal{P}_1, \ldots, \mathcal{P}_n \rangle$ and $\mathcal{R} \subseteq \mathcal{D}$
Output: true iff $\mathcal{R} \in \subseteq_P\text{-}Rep(\mathcal{K})$

```
if R is not T-consistent then
  return false
for 1 ≤ i ≤ n and α ∈ (D \ R) ∩ P_i do
  if R ∩ (P_1 ∪ ··· ∪ P_i) ∪ {α} is T-consistent then
    return false
return true
```

Algorithm 8 Pareto-optimal repair checking.

Input: $\mathcal{K}_\succ = (\langle \mathcal{D}, \mathcal{T} \rangle, \succ)$ and $\mathcal{R} \subseteq \mathcal{D}$
Output: true iff $\mathcal{R} \in P\text{-}Rep(\mathcal{K}_\succ)$

```
if R is not T-consistent then
  return false
for β ∈ (D \ R) do
  if R ∪ {β} \ {α | β ≻ α} is T-consistent then
    return false
return true
```

Algorithm 9 Completion-optimal repair checking.

Input: $\mathcal{K}_\succ = (\langle \mathcal{D}, \mathcal{T} \rangle, \succ)$ and $\mathcal{R} \subseteq \mathcal{D}$
Output: true iff $\mathcal{R} \in C\text{-}Rep(\mathcal{K}_\succ)$

```
if R is not T-consistent then
  return false
D' ← D, R' ← ∅
while S := {α | α ∈ D', ∀β ∈ D', β ⊁ α} ≠ ∅ do
  if S ∩ R = ∅ then
    return false
  choose α ∈ S ∩ R
  R' ← R' ∪ {α}, D' ← D' \ ({α} ∪ {β | ⟨R' ∪ {β}, T⟩ ⊨ ⊥})
return true
```

We show next that it also holds for FD and for DL-Lite$_{\mathsf{core}}$ (using only axioms that can be written as NCs) by two reductions from 3-SAT. Let $\varphi = c_1 \wedge \cdots \wedge c_m$ be a conjunction of clauses of three literals over variables $x_1, \dots, x_n$.

- For the FD case, define $\mathcal{K}_{\mathsf{FD}} = \langle \mathcal{D}_{\mathsf{FD}}, \mathcal{T}_{\mathsf{FD}} \rangle$ with

$$\begin{aligned}\mathcal{T}_{\mathsf{FD}} = \{&P(x_1,x_2,x_3,x_4,x_5) \wedge P(y_1,y_2,y_3,y_4,y_5) \wedge x_1 = y_1 \rightarrow x_2 = y_2,\\ &P(x_1,x_2,x_3,x_4,x_5) \wedge P(y_1,y_2,y_3,y_4,y_5) \wedge x_2 = y_2 \rightarrow x_3 = y_3,\\ &P(x_1,x_2,x_3,x_4,x_5) \wedge P(y_1,y_2,y_3,y_4,y_5) \wedge x_4 = y_4 \rightarrow x_5 = y_5\}\\ \mathcal{D}_{\mathsf{FD}} = \{&P(c_j,x_i,1,a,b) \mid x_i \in c_j, 1 \leq j \leq m\} \cup \{P(c_j,x_i,0,a,b) \mid \neg x_i \in c_j, 1 \leq j \leq m\}\\ \cup\ &\{P(d_j,e,f,a,c) \mid 1 \leq j \leq m-1\}\end{aligned}$$

  We can show that $\mathcal{R} = \{P(d_j,e,f,a,c) \mid 1 \leq j \leq m-1\}$ is not a $\leq$-optimal repair of $\mathcal{K}_{\mathsf{FD}}$ iff φ is satisfiable. Indeed, $\mathcal{R}$ is $\mathcal{T}_{\mathsf{FD}}$-consistent and $|\mathcal{R}| = m-1$, so $\mathcal{R} \notin \leq$-*Rep*$(\mathcal{K}_{\mathsf{FD}})$ iff there exists a $\mathcal{T}_{\mathsf{FD}}$-consistent $\mathcal{R}' \subseteq \mathcal{D}_{\mathsf{FD}}$ of size at least m. Such $\mathcal{R}'$ necessarily contains only facts of the form $P(c_j, x_i, \_, a, b)$ because of the last FD, so contains exactly m facts because of the first two FDs. It is easy to check that the valuation ν of the x_i defined by $\nu(x_i) = 1$ iff $\mathcal{R}'$ contains some $P(c_j, x_i, 1, a, b)$ satisfies every clause. In the other direction, every valuation that satisfies φ can be used to build a repair of size m.

- For the DL-Lite$_{\mathsf{core}}$ (and NC) case, define $\mathcal{K}_{\mathsf{DL}} = \langle \mathcal{D}_{\mathsf{DL}}, \mathcal{T}_{\mathsf{DL}} \rangle$ with

$$\begin{aligned}\mathcal{T}_{\mathsf{DL}} = \{&\exists P_\ell^- \sqsubseteq \neg\exists N_k^- \mid 1 \leq \ell, k \leq 3\} \cup \{\exists U^- \sqsubseteq \neg\exists P_\ell,\ \exists U^- \sqsubseteq \neg\exists N_\ell \mid 1 \leq \ell \leq 3\} \cup\\ &\{S \sqsubseteq \neg\exists U\} \cup \{\exists P_\ell \sqsubseteq \neg\exists P_k,\ \exists N_\ell \sqsubseteq \neg\exists N_k,\ \exists P_\ell \sqsubseteq \neg\exists N_k \mid 1 \leq \ell \neq k \leq 3\}\\ \mathcal{D}_{\mathsf{DL}} = \{&S(\varphi)\} \cup \{U(\varphi, c_j) \mid 1 \leq j \leq m\} \cup\\ &\{P_\ell(c_j, x_i) \mid x_i \text{ is the } \ell^{th} \text{ literal of } c_j\} \cup \{N_\ell(c_j, x_i) \mid \neg x_i \text{ is the } \ell^{th} \text{ literal of } c_j\}\end{aligned}$$

  We can show that $\mathcal{R} = \{U(\varphi, c_j) \mid 1 \leq j \leq m\}$ is not a $\leq$-optimal repair of $\mathcal{K}_{\mathsf{DL}}$ iff φ is satisfiable. Indeed, $\mathcal{R}$ is $\mathcal{T}_{\mathsf{DL}}$-consistent and $|\mathcal{R}| = m$, so $\mathcal{R} \notin \leq$-*Rep*$(\mathcal{K}_{\mathsf{DL}})$ iff there exists a $\mathcal{T}_{\mathsf{DL}}$-consistent $\mathcal{R}' \subseteq \mathcal{D}_{\mathsf{DL}}$ of size at least $m+1$. Since every repair of $\mathcal{K}_{\mathsf{DL}}$ contains at most one fact of the form $P_\ell(c_j, x_i)$ or $N_k(c_j, x_i)$ per c_j, and such fact prevents the corresponding $U(\varphi, c_j)$ to belong to the repair, the only way to obtain a repair of size $m+1$ is to select exactly one $P_\ell(c_j, x_i)$ or $N_k(c_j, x_i)$ per c_j, remove all $U(\varphi, c_j)$, and keep $S(\varphi)$. Since each x_i cannot occur both in P_ℓ- and N_k-facts, such repair $\mathcal{R}'$ corresponds to a valuation ν of the x_i such that $\nu(x_i) = 1$ iff $\mathcal{R}'$ contains some $P_\ell(c_j, x_i)$ and it is easy to check that ν satisfies φ. In the other direction, every satisfying valuation can be used to build a repair of size $m+1$.

6.2.2 BCQ entailment under CQA, intersection, and brave semantics

This section presents the proofs for the results on BCQ entailment under the different semantics given in Table 6.

▶ **Proposition 89.** *For every ontology language or anti-monotone IC language for which consistency checking and BCQ entailment are in P w.r.t. data complexity, the data complexity of BCQ entailment under CQA or intersection (resp. brave) with X-optimal repairs is in*

- *coNP (resp. NP) for $X \in \{\subseteq_P, \text{Pareto}, \text{completion}\}$;*
- *Π_2^P (resp. Σ_2^P) for $X = $ globally;*
- *Δ_2^P for $X \in \{\leq_w, \leq_P\}$;*
- *Θ_2^P for $X = \leq$ and for $X \in \{\leq_w, \leq_P\}$ if there is a data-independent bound on the weights or number of priority levels.*

Matching lower bounds hold for every ontology language or IC language that extends DL-Lite$_{\mathsf{core}}$, NC or FD.

Upper bounds. For semantics based on X-optimal repairs with X $\in$ $\{\subseteq_P, \text{Pareto}, \text{completion}\}$, since X-optimal repair checking is in P, the coNP and NP upper bounds for BCQ entailment under X-CQA, X-brave and X-intersection semantics follow from the non-deterministic versions of Algorithms 1, 2 and 3 for $X \in \{\subseteq_P, P, C\}$. Similarly, for semantics based on globally-optimal repairs, since globally-optimal repair checking is in coNP, the Π_2^{P} and Σ_2^{P} upper bounds for BCQ entailment under G-CQA, G-brave and G-intersection follow from these algorithms. For the other kinds of repairs, we use the reduction of $\leq$- and $\leq_P$-optimal repairs to $\leq_w$-optimal ones and Algorithms 11, 12 and 13 for semantics based on $\leq_w$-optimal repairs. These algorithms rely on an NP oracle to check the existence of $\mathcal{R} \subseteq \mathcal{D}$ as required both in the function OptimalWeightByBinarySearch (Algorithm 10) and the final condition.

Algorithm 10 OptimalWeightByBinarySearch.

Input: $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$, a weight function $w : \mathcal{D} \mapsto \mathbb{N}$
Output: the weight u_{opt} of $\leq_w$-optimal repairs

```
u_min ← 0, u_max ← Σ_{α∈D} w(α)
while u_min < u_max, let u = u_min + ⌈(u_max − u_min)/2⌉ and do
  if there exists a T-consistent R ⊆ D such that Σ_{α∈R} w(α) ≥ u then
    u_min ← u
  else
    u_max ← u − 1
return u_max
```

Algorithm 11 $\leq_w$-CQA.

Input: $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$, a weight function $w : \mathcal{D} \mapsto \mathbb{N}$, and a BCQ q
Output: true iff $\mathcal{K}$ entails q under CQA semantics based on $\leq_w$-$Rep(\mathcal{K})$

```
u_opt ← OptimalWeightByBinarySearch(K, w)
if there exists a T-consistent R ⊆ D such that Σ_{α∈R} w(α) = u_opt and ⟨R, T⟩ ⊭ q then
  return false
return true
```

Algorithm 12 $\leq_w$-brave.

Input: $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$, a weight function $w : \mathcal{D} \mapsto \mathbb{N}$, and a BCQ q
Output: true iff $\mathcal{K}$ entails q under brave semantics based on $\leq_w$-$Rep(\mathcal{K})$

```
u_opt ← OptimalWeightByBinarySearch(K, w)
if there exists a T-consistent R ⊆ D such that Σ_{α∈R} w(α) = u_opt and ⟨R, T⟩ ⊨ q then
  return true
return false
```

The number of calls to the NP oracle is logarithmically bounded in $\Sigma_{\alpha \in \mathcal{D}} w(\alpha)$ so is polynomial w.r.t. the size of $\mathcal{D}$ if the weights (considered part of $\mathcal{D}$) are encoded in binary, and logarithmic if the weights are encoded in unary or if there exists a constant k independent of $\mathcal{D}$ such that $w(\alpha) \leq k$ for every α, since in this case $\Sigma_{\alpha \in \mathcal{D}} w(\alpha) \leq k|\mathcal{D}|$. In particular, this is the case for $\leq$-optimal repairs (case $w(\alpha) = 1$). For $\leq_P$-optimal repairs, we use the reduction to $\leq_w$-optimal repairs presented in Section 5.1: given $P = \langle \mathcal{P}_1, \ldots, \mathcal{P}_n \rangle$, let $u = (\max_{i=1}^{n} |\mathcal{P}_i|) + 1$, and $w(\alpha) = u^{n-i}$ for $\alpha \in \mathcal{P}_i$, then $\leq_P$-$Rep(\mathcal{D}, \mathcal{T}) =\leq_w$-$Rep(\mathcal{D}, \mathcal{T})$. Again, if n is bounded by a constant k independent from $\mathcal{D}$, the resulting weights are bounded by $(|\mathcal{D}| + 1)^{k-1}$ so $\Sigma_{\alpha \in \mathcal{D}} w(\alpha) \leq (|\mathcal{D}| + 1)^k$ and the number of oracle calls is logarithmic.

Algorithm 13 $\leq_w$-intersection.

Input: $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$, a weight function $w : \mathcal{D} \mapsto \mathbb{N}$, and a BCQ q
Output: true iff $\mathcal{K}$ entails q under intersection semantics based on $\leq_w$-$Rep(\mathcal{K})$

$u_{opt} \leftarrow$ OptimalWeightByBinarySearch$(\mathcal{K}, w)$
if there exist $\mathcal{B} \subseteq \mathcal{D}$ with $\mathcal{B} = \{\alpha_1, \dots, \alpha_n\}$ and $\mathcal{T}$-consistent $\mathcal{R}_1 \subseteq \mathcal{D}, \dots, \mathcal{R}_n \subseteq \mathcal{D}$ such that $\alpha_i \notin \mathcal{R}_i$, $\Sigma_{\alpha \in \mathcal{R}_i} w(\alpha) = u_{opt}$ for $1 \leq i \leq n$, and $\langle \mathcal{D} \setminus \mathcal{B}, \mathcal{T} \rangle \not\models q$ **then**
  return false
return true

Lower bounds. For semantics based on X-optimal repairs with X $\in$ $\{\subseteq_P$, Pareto, completion$\}$, the coNP and NP lower bounds have been shown in the case of $\subseteq_P$-optimal repair-based semantics for DL-Lite$_{\mathsf{core}}$ [44, Prop. 6.2.3 and 6.2.8] and the ontologies used in the reduction are equivalent to sets of NCs. Since in the case where a priority relation $\succ$ is induced by a prioritization $P = \langle \mathcal{P}_1, \dots, \mathcal{P}_n \rangle$, Pareto- and completion-optimal repairs coincide with $\subseteq_P$-optimal repairs, these lower bounds transfer to the cases of Pareto- and completion-optimal repairs. For FD, the coNP and NP lower bounds have been shown for X-CQA [129, Theorem 3] and for X-intersection and X-brave [31, Theorem 3], either directly for $X = \subseteq_P$ or using a priority relation that can be induced by a prioritization.

For semantics based on globally-optimal repairs, the Π_2^{P} and Σ_2^{P} lower bounds have been shown for DL-Lite$_{\mathsf{core}}$ using ontologies equivalent to sets of NCs [30, Theorem 10], and the Π_2^{P}-hardness proof for G-CQA with FDs [129, Theorem 2] immediately yields the lower bound for G-intersection and G-brave semantics. Indeed, it uses as query a fact $p_\exists$ from the dataset, and every repair that does not contain $p_\exists$ contains $p_\forall$. Hence $p_\exists$ is entailed under G-CQA iff it is entailed under G-intersection iff $p_\forall$ is not entailed G-brave semantics.

For semantics based on X-optimal repairs with X $\in \{\leq, \leq_w, \leq_P\}$, a Θ_2^{P} lower bound has been shown for $\leq$-CQA, $\leq$-intersection and $\leq$-brave semantics (hence transfers to $\leq_w$- and $\leq_P$-optimal repair-based semantics, even in the case where the weights are bounded by a constant and there is a single priority level) [44, Prop. 6.2.4 and 6.2.9] and the reductions can be easily adapted to use only NCs. The lower bound for $\leq$-CQA has also been shown for a logical theory that contains a single NC [103, Theorem 4]. For FD, we use a reduction from the following Θ_2^{P}-hard problem [103, Lemma 6]: given a graph $\mathcal{G} = (V, E)$ and a vertex $v_d \in V$, decide if v_d belongs to all (cardinality) maximum independent sets (MISs) of $\mathcal{G}$. Assume w.l.o.g. that every $v \in V$ belongs to some $e \in E$ and define $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ as follows, where for $v \in V$, $ne(v) = |\{e \mid v \in e, e \in E\}|$ is the number of edges of $\mathcal{G}$ in which v occurs.

$$\begin{aligned}\mathcal{T} = \{&R(x_1, x_2, x_3, x_4) \wedge R(y_1, y_2, y_3, y_4) \wedge x_1 = y_1 \rightarrow x_2 = y_2,\\ &R(x_1, x_2, x_3, x_4) \wedge R(y_1, y_2, y_3, y_4) \wedge x_2 = y_2 \rightarrow x_3 = y_3\}\\ \mathcal{D} = \{&R(e, v, 1, j) \mid v \in e, e \in E, 1 \leq j \leq 2\} \cup \{R(v, v, 0, i) \mid v \in V, 1 \leq i \leq 2ne(v) - 1\}\end{aligned}$$

We show that v_d belongs to all MISs of $\mathcal{G}$ iff $\exists x R(x, v_d, 1, 1)$ is entailed under $\leq$-CQA (resp. $\leq$-intersection) semantics, iff $R(v_d, v_d, 0, 1)$ is *not* entailed under $\leq$-brave semantics. We first show a correspondence between $\leq$-optimal repairs of $\mathcal{K}$ and MISs of $\mathcal{G}$:

- Given an independent set $S \subseteq V$ of $\mathcal{G}$, define $\mathcal{R}_S = \{R(e, v, 1, j) \mid R(e, v, 1, j) \in \mathcal{D}, v \in S\} \cup \{R(v, v, 0, i) \mid R(v, v, 0, i) \in \mathcal{D}, v \notin S\}$. $\mathcal{R}_S$ satisfies the second FD by construction and the first one because S is an independent set. Hence $\mathcal{R}_S$ is $\mathcal{T}$-consistent. Moreover, $|\mathcal{R}_S| = \Sigma_{v \in S} 2ne(v) + \Sigma_{v \notin S}(2ne(v) - 1) = \Sigma_{v \in V} 2ne(v) - |V \setminus S|$.

- Given a $\mathcal{T}$-consistent subset $\mathcal{R} \subseteq \mathcal{D}$, define $S_\mathcal{R} = \{v \mid R(e, v, 1, j) \in \mathcal{R}\}$. For each $v \in S_\mathcal{R}$, $\mathcal{R}$ does not contain any fact of the form $R(v, v, 0, i)$ (or it would violate the second FD) and for each $v \notin S_\mathcal{R}$, by definition of $S_\mathcal{R}$, $\mathcal{R}$ does not contain any fact of the form $R(e, v, 1, j)$. Hence $|\mathcal{R}| \leq \Sigma_{v \in S_\mathcal{R}} 2ne(v) + \Sigma_{v \notin S_\mathcal{R}}(2ne(v) - 1) = \Sigma_{v \in V} 2ne(v) - |V \setminus S_\mathcal{R}|$.
- Let $\mathcal{R} \in \leq\text{-}Rep(\mathcal{K})$. We show that $S_\mathcal{R}$ is an independent set of $\mathcal{G}$. Assume for a contradiction that there is an edge $e_0 = (v_1, v_2)$ such that $v_1, v_2 \in S_\mathcal{R}$. For $k \in \{1, 2\}$, let $\mathcal{R}_k = \mathcal{R} \setminus \{R(e', v_k, 1, j) \mid v_k \in e', e' \in E, 1 \leq j \leq 2\} \cup \{R(v_k, v_k, 0, i) \mid 1 \leq i \leq 2ne(v_k) - 1\}$. Since $\mathcal{R}$ is $\mathcal{T}$-consistent, it is easy to check that the $\mathcal{R}_k$ are also $\mathcal{T}$-consistent. Since $v_1, v_2 \in S_\mathcal{R}$, $\mathcal{R}$ must contain some facts of the form $R(e_1, v_1, 1, j)$ and $R(e_2, v_2, 1, j)$ (by definition of $S_\mathcal{R}$). Hence $\mathcal{R}$ does not contain any fact of the form $R(v_1, v_1, 0, i)$ nor $R(v_2, v_2, 0, i)$. Moreover, by the first FD, there is $k \in \{1, 2\}$ such that $\mathcal{R}$ does not contain $R(e_0, v_k, 1, j)$ for $1 \leq j \leq 2$. Hence there is $k \in \{1, 2\}$ such that $\mathcal{R}_k$ is obtained from $\mathcal{R}$ by adding exactly $2ne(v_k) - 1$ facts of the form $R(v_k, v_k, 0, i)$ and removing at most $2ne(v_k) - 2$ facts of the form $R(e', v_k, 1, j)$ so $|\mathcal{R}_k| > |\mathcal{R}|$. This contradicts the fact that $\mathcal{R} \in \leq\text{-}Rep(\mathcal{K})$. Hence $S_\mathcal{R}$ is an independent set of $\mathcal{G}$.
- Let $\mathcal{R} \in \leq\text{-}Rep(\mathcal{K})$. We have shown that $S_\mathcal{R}$ is an independent set of $\mathcal{G}$ and $|\mathcal{R}| \leq \Sigma_{v \in V} 2ne(v) - |V \setminus S_\mathcal{R}|$. Assume for a contradiction that there exists an independent set S of $\mathcal{G}$ such that $|S| > |S_\mathcal{R}|$. We have shown that $|\mathcal{R}_S|$ is $\mathcal{T}$-consistent and $|\mathcal{R}_S| = \Sigma_{v \in V} 2ne(v) - |V \setminus S|$. It follows that $|\mathcal{R}_S| = \Sigma_{v \in V} 2ne(v) - |V \setminus S| > \Sigma_{v \in V} 2ne(v) - |V \setminus S_\mathcal{R}| \geq |\mathcal{R}|$, which contradicts $\mathcal{R} \in \leq\text{-}Rep(\mathcal{K})$. We conclude that $S_\mathcal{R}$ is a MIS of $\mathcal{G}$.
- Let $S \subseteq V$ be a MIS of $\mathcal{G}$. Since S is an independent set, $\mathcal{R}_S$ is $\mathcal{T}$-consistent and $|\mathcal{R}_S| = \Sigma_{v \in V} 2ne(v) - |V \setminus S|$. Assume for a contradiction that $\mathcal{R}_S \notin \leq\text{-}Rep(\mathcal{K})$ and let $\mathcal{R} \in \leq\text{-}Rep(\mathcal{K})$. We have shown that $S_\mathcal{R}$ is an independent set of $\mathcal{G}$ and $|\mathcal{R}| \leq \Sigma_{v \in V} 2ne(v) - |V \setminus S_\mathcal{R}|$. Since we assumed that $|\mathcal{R}| > |\mathcal{R}_S|$, it follows that $|S_\mathcal{R}| > |S|$, which contradicts that S is a MIS of $\mathcal{G}$. We conclude that $\mathcal{R}_S \in \leq\text{-}Rep(\mathcal{K})$.

Assume that v_d belongs to every MIS of $\mathcal{G}$ and let $\mathcal{R} \in \leq\text{-}Rep(\mathcal{K})$. Since $S_\mathcal{R}$ is a MIS of $\mathcal{G}$, $v_d \in S_\mathcal{R}$ so there exists $R(e_d, v_d, 1, j) \in \mathcal{R}$ and $\mathcal{R}$ does not contain any fact of form $R(v_d, v_d, 0, j)$ and contains $R(e_d, v_d, 1, 1)$ (by maximality). Hence $\exists x R(x, v_d, 1, 1)$ is entailed under $\leq$-intersection (hence $\leq$-CQA) semantics and $R(v_d, v_d, 0, 1)$ is not entailed under $\leq$-brave semantics. In the other direction, assume that there exists a MIS S of $\mathcal{G}$ such that $v_d \notin S$. Then $\mathcal{R}_S \in \leq\text{-}Rep(\mathcal{K})$ is such that there is no fact of the form $R(e, v_d, 1, j)$ in $\mathcal{R}_S$ and $R(v_d, v_d, 0, 1) \in \mathcal{R}_S$ so $R(v_d, v_d, 0, 1)$ is entailed under $\leq$-brave semantics and $\exists x R(x, v_d, 1, 1)$ is not entailed under $\leq$-CQA (hence nor under $\leq$-intersection) semantics.

For semantics based on X-optimal repairs with X $\in \{\leq_w, \leq_P\}$ (in the case where the weights and priority levels are not bounded independently from the data), a Δ_2^P lower bound has been shown for $\leq_P$-CQA, $\leq_P$-intersection and $\leq_P$-brave semantics (hence transfers to $\leq_w$-optimal repair-based semantics) in DL-Lite$_\mathsf{core}$ [44, Prop. 6.2.4 and 6.2.9] and the ontology used in the reductions uses only NCs. For FD, we use a reduction from the following Δ_2^P-hard problem [96]: given a satisfiable CNF formula $\varphi = c_1 \wedge \cdots \wedge c_m$ over variables $x_1, \ldots, x_n$, decide whether the lexicographically maximum truth assignment ν_max satisfying φ with respect to $(x_1, \ldots, x_n)$ is such that $\nu_\mathsf{max}(x_n) = 1$. Define $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$ and a prioritization $P = \langle \mathcal{P}_1, \ldots, \mathcal{P}_{n+1} \rangle$ as follows:

$$
\begin{aligned}
\mathcal{T} = \{&R(x_1, x_2, x_3) \wedge R(y_1, y_2, y_3) \wedge x_1 = y_1 \rightarrow x_2 = y_2,\\
&R(x_1, x_2, x_3) \wedge R(y_1, y_2, y_3) \wedge x_2 = y_2 \rightarrow x_3 = y_3\}\\
\mathcal{D} = \ &\mathcal{P}_1 \cup \cdots \cup \mathcal{P}_{n+1} \text{ with}\\
\mathcal{P}_1 = \ &\{R(c_j, x_i, 1) \mid x_i \in c_j, 1 \leq j \leq m\} \cup \{R(c_j, x_i, 0) \mid \neg x_i \in c_j, 1 \leq j \leq m\}\\
\mathcal{P}_{i+1} = \ &\{R(x_i, x_i, 1)\} \text{ for } 1 \leq i \leq n
\end{aligned}
$$

We can show that $R(x_n, x_n, 1)$ is entailed under $\leq_P$-CQA (resp. $\leq_P$-intersection, $\leq_P$-brave) semantics iff $\nu_{\mathsf{max}}(x_n) = 1$. Indeed, since φ is satisfiable, for each satisfying valuation ν, we can obtain a $\mathcal{T}$-consistent subset of m facts from $\mathcal{P}_1$ by selecting exactly one fact of the form $R(c_j, x_i, \nu(x_i))$ per c_j and complete it into a (standard) repair by adding all $R(x_i, x_i, 1)$ such that no fact of the form $R(c_j, x_i, 0)$ has been selected. Since it is not possible to keep more than m facts from $\mathcal{P}_1$ because of the first FD, and each other $\mathcal{P}_i$ contains a single fact, the $\leq_P$-repairs all contain the same set of facts from $\bigcup_{i=1}^{n} \mathcal{P}_{i+1}$ and these facts correspond to the x_i set to 1 by ν_{max}.

6.2.3 Comments and additional results

When looking at Tables 6 and 7, we can distinguish three kinds of preferred repairs depending on their impact on the data complexity:

- Those for which the complexity of repair checking and BCQ entailment under CQA is the same as in the case of standard repairs ($\subseteq_P$-, Pareto- and completion-optimal). Intuitively, these preferred repairs correspond to some "local" preference, so that given a repair, we can decide whether it is a preferred repair without guessing or computing another preferred repair. Note that the complexity of intersection and brave semantics nevertheless increases for the cases where it was in AC^0 for standard repairs.
- Those for which the complexity of repair checking and BCQ entailment under CQA increases to coNP-complete and Δ_2^{P}- or Θ_2^{P}-complete ($\leq$-, $\leq_w$-, $\leq_P$-optimal and $\Pi_{\leq}$-preferred). Intuitively, for these kinds of preferred repairs, we can precompute the value of a global parameter (such as the weight of the $\leq_w$-optimal repairs or the number of preference rules satisfied by $\Pi_{\leq}$-preferred repairs) that allows us to check efficiently that a $\mathcal{T}$-consistent subset of $\mathcal{D}$ is a preferred repair.
- Those for which the complexity of repair checking and BCQ entailment under CQA jumps to coNP-complete and Π_2^{P}-complete (globally-optimal and $\Pi_{\subseteq}$-preferred). Intuitively, for these kinds of preferred repairs, there is no better option than relying on the naïve guess-and-check algorithm to determine whether a repair is a preferred repair.

We conclude this section by mentioning various other complexity results for reasoning under CQA, intersection and brave semantics using preferred repairs.

Combined complexity. The combined complexity of BCQ entailment under CQA and intersection semantics based on $\preceq$-optimal repairs over Datalog$^{\pm}$ KBs have been studied for $\preceq \in \{\leq, \leq_w, \subseteq_P, \leq_P\}$ and shown to coincide with the standard repair case for all fragments of Datalog$^{\pm}$ we consider (cf. [108, Tables 2 and 3] for $\leq$ and [107, Tables 4, 5, 6, 7] for $\leq_w$, $\leq_P$ and $\subseteq_P$). Intuitively, this is because the combined complexity of BCQ entailment under these semantics is already very high when using standard repairs. In contrast, for DL-Lite$_{\mathcal{R}}$, the combined complexity of CQA does not change (Π_2^{P}-complete) but the one of intersection semantics increases from NP-complete to Θ_2^{P} or Δ_2^{P} when $\leq$-, $\subseteq_P$-, $\leq_w$- or $\leq_P$-optimal repairs are used [33, Figure 1], and the one of brave semantics increases similarly when using $\leq$-, $\leq_w$- or $\leq_P$-optimal repairs (but remains NP-complete in the case of $\subseteq_P$-optimal repairs) [44, Table 6.1]. Note that this difference between CQA and intersection or brave in combined complexity contrasts with the data complexity where CQA and intersection have the same complexity and brave is complete for the complement class.

Preferred repairs for more expressive DLs. It has been shown that fact entailment over $\mathcal{SHIQ}$ KBs under $\leq_w$-CQA semantics is Θ_2^{P}-complete when the input size is measured by the number of facts in $\mathcal{D}$ (hence assuming a data-independent bound on the weights) [73].

We can extend this result and obtain Δ_2^{P}-complete (resp. Θ_2^{P}-complete when the weights are bounded) data complexity for BCQ entailment under $\leq_w$-CQA, as well as under $\leq_w$-intersection (extended as usual to $\leq_P$ and $\leq$). The lower bounds are inherited from DL-Lite$_{\mathsf{core}}$ and the upper bounds follow from Algorithms 11 and 13. Indeed, since consistency checking in $\mathcal{SHIQ}$ is in NP, an NP-oracle can decide whether there exists a $\mathcal{T}$-consistent subset $\mathcal{R}$ of $\mathcal{D}$ of at least a given weight by guessing the certificate of the $\mathcal{T}$-consistency of $\mathcal{R}$ together with $\mathcal{R}$, and similarly, since BCQ entailment is in coNP, the oracle can guess the certificate that $\langle \mathcal{R}, \mathcal{T} \rangle \not\models q$ (resp. $\langle \mathcal{D} \setminus \mathcal{B}, \mathcal{T} \rangle \not\models q$) together with $\mathcal{R}$ (resp. $\mathcal{B}$ and the $\mathcal{R}_1, \ldots, \mathcal{R}_n$) and the $\mathcal{T}$-consistency certificate(s).

Preferred Δ-repairs of databases with universal constraints. The data complexity of reasoning with optimal repairs based on a priority relation has been studied in the case of databases with UCs: in this case, using $\subseteq_P$-optimal, Pareto-optimal, completion-optimal or globally-optimal repairs instead of standard Δ-repairs does not increase the data complexity of BCQ entailment under CQA, intersection and brave semantics, which remains $\Pi_2^{\mathrm{P}}/\Sigma_2^{\mathrm{P}}$-complete (cf. Table 5) [32, Theorem 3]. For $\leq$-optimal repairs, the data complexity actually decreases: it is Θ_2^{P}-complete [74, Theorem 4]. The lower bound comes from the FD case and the upper bound follows from the fact that we can pre-compute by binary search the minimal size of the symmetric difference between a Δ-repair and $\mathcal{D}$, then use it to check in P whether a subset of $Facts_{\mathcal{D}}^{\mathcal{T}}$ is a $\leq$-optimal Δ-repair.

Optimal repairs based on a transitive priority relation. In the case of repairs based on a priority relation $\succ$, it has been shown (for DL-Lite$_{\mathsf{core}}$ and NCs) that all lower bounds given in Table 6 hold even in the case where $\succ$ is assumed to be transitive [30, Theorem 14] (while transitivity decreases the complexity of deciding whether there is a unique globally-optimal repair [92, Corollary 7.7]).

7 Implementations of (preferred) repair-based semantics

Table 8 provides a (partial) overview of the systems that have been implemented for answering queries under (preferred) repair-based semantics. They are divided in four groups, based on the semantics they support:

- the systems of the first group target CQA, intersection and brave semantics with both standard and preferred repairs based on priority ($\subseteq_P$-, Pareto- and completion-optimal);
- the second group consists of implementations of $\leq_w$- and/or $\leq$-CQA for expressive DLs or database constraints;
- the systems of the third group implement (plain) intersection semantics for DL KBs;
- the last group consists of systems that implement (plain) CQA for databases.

Most of the systems in this last group further fall into two categories: approaches that handle general CQs and potentially expressive ICs and rely on solvers (e.g., SAT, Binary Integer Programming (BIP), Answer Set Programming (ASP)), and those that impose strong restrictions on the setting (key constraints and restriction to some classes of CQs) to ensure that CQA is FO rewritable and apply rewriting techniques. Indeed, the database community conducted an extensive study of the complexity of BCQ entailment under CQA with key constraints for different kinds of CQ, which is out of the scope of these lecture notes.

In the remainder of the section, we illustrate the prominent approach of CQA relying on solvers by presenting some SAT encodings for CQA based on standard or Pareto-optimal repairs, then give a few details on the approaches to tractable semantics and approximations.

Table 8 Overview of implementations of (preferred) repair-based semantics. X-Sem stands for semantics Sem based on repairs from $X\text{-}Rep(\mathcal{K})$. Recall that a system that supports $\leq_w$-Sem also supports $\leq$-Sem, and that a system that supports P- or C-Sem also supports $\subseteq_P$-Sem. We focus on (preferred) repair-based semantics (cf. Section 4.1) and omit here the alternative semantics implemented in some systems.

	semantics	input supported	main techniques
[37] (2025)	X-CQA/X-inter./X-brave for $X \in \{S, P, C\}$	$\mathcal{D}$, metadata, pref. rules, DCs, UCQ	ASP
ORBITS [31] (2022)	X-CQA/X-inter./X-brave for $X \in \{S, P, C\}$	binary $Conf(\mathcal{K})$, $\succ$, pot. answers + causes	SAT + poly. approx.
CQAPri [33, 35] (2014)	X-CQA/X-inter./X-brave for $X \in \{S, \subseteq_P\}$	$\mathcal{D}$ (possibly with P), DL-Lite$_\mathcal{R}$ onto., CQ	SAT + poly. approx.
[73] (2013)	$\leq_w$-CQA	$\mathcal{D}$, w, $\mathcal{SHIQ}$ onto., CQ without $\exists$	SAT
[74] (2010)	$\leq$-CQA	$\mathcal{D}$, UCs, CQ	SAT
[133] (2018)	S-inter.	$\mathcal{D}$, $\mathcal{ELH}^{dr}_{\bot}$ onto. (with restr.), CQ	(disj.) Datalog$^\neg$ rew.
SaQAI [134] (2016)	S-inter.	$\mathcal{D}$, DL-Lite$_\mathcal{R}$ onto., CQ	comp. $\bigcap_{\mathcal{R} \in S\text{-}Rep(\mathcal{K})} \mathcal{R}$
QuID [127, 99] (2012)	S-inter.	$\mathcal{D}$, DL-Lite$_\mathcal{A}$ onto., CQ	FO rewriting annot. with $Conf(\mathcal{K})$ comp. $\bigcap_{\mathcal{R} \in S\text{-}Rep(\mathcal{K})} \mathcal{R}$
CAvSAT [71, 72] (2019)	S-CQA	$\mathcal{D}$, DCs, CQ	SAT + poly. approx. SQL rewriting
EQUIP [95] (2013)	S-CQA	$\mathcal{D}$, keys, CQ	BIP + poly. approx.
[86, 87, 77, 117] [114, 115](2001-15)	S-CQA	$\mathcal{D}$, keys to UCs, CQ to UCQ$^\neg$	logic programing
Hippo [64, 65] (2004)	S-CQA	$\mathcal{D}$, DCs, UCQ without $\exists$	indep. set of $Conf(\mathcal{K})$
LinCQA [79] (2023)	S-CQA	$\mathcal{D}$, keys, FO rew. CQ	SQL rewriting non-rec. Datalog$^\neg$ rew.
Conquesto [91] (2020)	S-CQA	$\mathcal{D}$, keys, FO rew. CQ	non-rec. Datalog$^\neg$ rew.
Conquer [82, 83] (2005)	S-CQA	$\mathcal{D}$, keys, FO rew. CQ	SQL rewriting annot. with $Conf(\mathcal{K})$

7.1 Examples of encodings for CQA

When encoding BCQ entailment under some intractable semantics into SAT, BIP, or ASP, it is crucial to produce encodings that are small enough to be handled efficiently. Localization techniques have been considered since the early implementations based on logic programing (see e.g., [77]). In this section, we present some examples of SAT encodings for CQA semantics using standard or Pareto-optimal repairs (applicable to $\subseteq_P$-optimal repairs) that have been proposed for cases where the conflicts and minimal $\mathcal{T}$-consistent subsets of the dataset that support the query can be efficiently computed (in polynomial time w.r.t. data complexity), such as DL-Lite, $\mathsf{L}_\bot$, $\mathsf{A}_\bot$, or $\mathsf{S}_\bot$ KBs, or databases with DCs. Indeed, most of the recent implementations of CQA that handle unrestricted CQs and relatively expressive constraints use encodings based on these sets [95, 33, 71, 31, 37]. Several variants of such encodings have been proposed. In particular, we can treat each candidate answer separately, reducing each entailment to an instance of (here) SAT, or use the same encoding to handle all potential answers, and rely on some optimization variant (e.g., weighted MaxSAT), or incremental solving. We refer the interested reader to [31] for a comparison of different SAT encodings and different reasoning modes of SAT solvers. The encodings we present next are used by CQAPri [33] and implemented among others by ORBITS [31] (restricted to the case of binary conflicts, which simplifies the encodings). An ASP version of these encodings (for the general case where conflicts may not be binary) has also recently been implemented [37].

Given a KB or database $\mathcal{K} = \langle \mathcal{D}, \mathcal{T} \rangle$, a CQ $q(\vec{x})$, and a tuple $\vec{a}$, a *cause* for $q(\vec{a})$ in $\mathcal{K}$ is a $\subseteq$-minimal $\mathcal{T}$-consistent subset $\mathcal{C}$ of $\mathcal{D}$ such that $\langle \mathcal{C}, \mathcal{T} \rangle \models q(\vec{a})$. We denote by $Causes(q(\vec{a}), \mathcal{K})$ the set of all causes for $q(\vec{a})$ in $\mathcal{K}$. Recall from Sections 3.3 and 5.2.2.1 that given a prioritized KB or database $\mathcal{K}_\succ = (\mathcal{K}, \succ)$ we have the following notation (attack relations of the SETAF corresponding to $\mathcal{K}$ and of the PSETAF corresponding to $\mathcal{K}_\succ$, respectively):

- $\mathcal{B} \rightsquigarrow_\mathcal{K} \alpha$ iff $\mathcal{B} \cup \{\alpha\} \in Conf(\mathcal{K})$ and $\alpha \notin \mathcal{B}$;
- $\mathcal{B} \rightsquigarrow_\succ \alpha$ iff $\mathcal{B} \rightsquigarrow_\mathcal{K} \alpha$ and $\alpha \not\succ \beta$ for every $\beta \in \mathcal{B}$.

The propositional formula $\Phi_{\mathsf{CQA}}(q(\vec{a}))$ is satisfiable iff $\vec{a}$ is *not* an answer to $q(\vec{x})$ over $\mathcal{K}$ under the CQA semantics.

$$\Phi_{\mathsf{CQA}}(q(\vec{a})) = \Big(\bigwedge_{\mathcal{C} \in Causes(q(\vec{a}),\mathcal{K})} \varphi_{\neg\mathcal{C}}\Big) \wedge \varphi_{\mathsf{cons}}(F) \quad \text{where}$$

$$\varphi_{\neg\mathcal{C}} = \Big(\bigvee_{\substack{\alpha \in \mathcal{C},\\ \mathcal{B} \rightsquigarrow_\mathcal{K} \alpha}} x_\mathcal{B}\Big) \wedge \bigwedge_{\substack{\alpha \in \mathcal{C}, \mathcal{B} \rightsquigarrow_\mathcal{K} \alpha,\\ \beta \in \mathcal{B}}} \neg x_\mathcal{B} \vee x_\beta \qquad \text{and} \qquad \varphi_{\mathsf{cons}}(F) = \bigwedge_{\substack{\alpha \in F,\\ \mathcal{B} \rightsquigarrow_\mathcal{K} \alpha}} \Big(\neg x_\alpha \vee \bigvee_{\beta \in \mathcal{B}} \neg x_\beta\Big)$$

and $F = \{\beta \mid x_\beta \text{ occurs in } \bigwedge_{\mathcal{C} \in Causes(q(\vec{a}),\mathcal{K})} \varphi_{\neg\mathcal{C}}\}$. Intuitively, the set of facts β such that x_β is assigned to true by a satisfying valuation for $\Phi_{\mathsf{CQA}}(q(\vec{a}))$ is $\mathcal{T}$-consistent and contradicts every cause for $q(\vec{a})$ in $\mathcal{K}$, hence can be extended into a repair of $\mathcal{K}$ that does not entail $q(\vec{a})$. Indeed, for each cause $\mathcal{C}$ of $q(\vec{a})$ in $\mathcal{K}$, $\varphi_{\neg\mathcal{C}}$ ensures that we choose some $\alpha \in \mathcal{C}$ and some $\mathcal{B}$ such that $\mathcal{B} \cup \{\alpha\} \in Conf(\mathcal{K})$, and select all facts from $\mathcal{B}$, and $\varphi_{\mathsf{cons}}(F)$ ensures that the set of selected facts is $\mathcal{T}$-consistent.

To use Pareto-optimal repairs instead of standard ones, it suffices to slightly modify $\varphi_{\neg\mathcal{C}}$ to take into account the priority relation and to add an additional component $\varphi_{\mathsf{P\text{-}max}}(F)$ that ensures that the selected facts can be extended into a Pareto-optimal repair by stating that a relevant fact α can only be omitted if we include a contradicting set of facts that are not dominated by α. The propositional formula $\Phi_{\mathsf{P\text{-}CQA}}(q(\vec{a}))$ is satisfiable iff $\vec{a}$ is *not* an answer to $q(\vec{x})$ over $\mathcal{K}_\succ$ under the CQA semantics based on Pareto-optimal repairs.

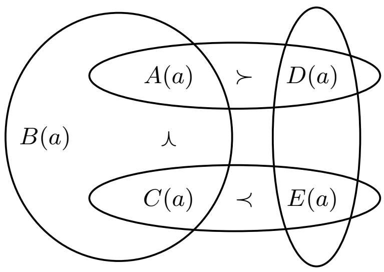
$$\Phi_{\mathsf{P\text{-}CQA}}(q(\vec{a})) = (\bigwedge_{\mathcal{C}\in Causes(q(\vec{a}),\mathcal{K})} \varphi^{\succ}_{\neg\mathcal{C}}) \wedge \varphi_{\mathsf{P\text{-}max}}(F) \wedge \varphi_{\mathsf{cons}}(F') \quad \text{where}$$

$$\varphi^{\succ}_{\neg\mathcal{C}} = (\bigvee_{\substack{\alpha\in\mathcal{C},\\ \mathcal{B}\rightsquigarrow_{\succ}\alpha}} x_{\mathcal{B}}) \wedge \bigwedge_{\substack{\alpha\in\mathcal{C},\mathcal{B}\rightsquigarrow_{\succ}\alpha,\\ \beta\in\mathcal{B}}} \neg x_{\mathcal{B}} \vee x_{\beta} \quad \text{and}$$

$$\varphi_{\mathsf{P\text{-}max}}(F) = \bigwedge_{\alpha\in\mathsf{R}(F)} (x_{\alpha} \vee \bigvee_{\mathcal{B}\rightsquigarrow_{\succ}\alpha} x_{\mathcal{B}}) \wedge \bigwedge_{\substack{\alpha\in\mathsf{R}(F),\\ \mathcal{B}\rightsquigarrow_{\succ}\alpha,\beta\in\mathcal{B}}} \neg x_{\mathcal{B}} \vee x_{\beta}$$

$F = \{\beta \mid x_\beta \text{ occurs in } \bigwedge_{\mathcal{C}\in Causes(q(\vec{a}),\mathcal{K})} \varphi^{\succ}_{\neg\mathcal{C}}\}$, $F' = \{\beta \mid x_\beta \text{ occurs in } \varphi_{\mathsf{P\text{-}max}}(F)\}$, and $\mathsf{R}(F)$ is the set of facts reachable from F in the following sense: β is reachable from F if $\beta \in F$ or if there exists γ reachable from F and $\mathcal{B}$ such that $\beta \in \mathcal{B}$ and $\mathcal{B} \rightsquigarrow_{\succ} \gamma$.

▶ **Example 90.** Recall the KB $\mathcal{K}$ from Examples 15, 51 and 59, whose conflict hypergraph, priority relation and Pareto-optimal repairs are recalled below, and let $q(x) = B(x)$.



$S\text{-}Rep(\mathcal{K}_\succ) = \{\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3\}$ and
$P\text{-}Rep(\mathcal{K}_\succ) = \{\mathcal{R}_1, \mathcal{R}_3\}$ with:
$\mathcal{R}_1 = \{A(a), B(a), E(a)\}$
$\mathcal{R}_2 = \{A(a), C(a)\}$
$\mathcal{R}_3 = \{B(a), C(a), D(a)\}$

The only cause for $q(a)$ is $\mathcal{C} = \{B(a)\}$ and the SAT encodings to decide whether $q(a)$ holds under CQA based on standard and Pareto-optimal repairs, respectively, are as follows (for readability we write A, B, C, D, E instead of $A(a), B(a), C(a), D(a), E(a)$ in indexes).

$$\begin{aligned}
\Phi_{\mathsf{CQA}}(q(a)) =& x_{\{A,C\}} \wedge (\neg x_{\{A,C\}} \vee x_A) \wedge (\neg x_{\{A,C\}} \vee x_C) \\
& \wedge (\neg x_A \vee \neg x_B \vee \neg x_C) \\
\Phi_{\mathsf{P\text{-}CQA}}(q(a)) =& x_{\{A,C\}} \wedge (\neg x_{\{A,C\}} \vee x_A) \wedge (\neg x_{\{A,C\}} \vee x_C) \\
& \wedge (x_A \vee x_{\{B,C\}}) \wedge (\neg x_{\{B,C\}} \vee x_B) \wedge (\neg x_{\{B,C\}} \vee x_C) \\
& \wedge (x_B \vee x_{\{A,C\}}) \wedge (\neg x_{\{A,C\}} \vee x_A) \wedge (\neg x_{\{A,C\}} \vee x_C) \\
& \wedge (x_C \vee x_{\{E\}}) \wedge (\neg x_{\{E\}} \vee x_E) \\
& \wedge (x_D \vee x_{\{A\}} \vee x_{\{E\}}) \wedge (\neg x_{\{A\}} \vee x_A) \wedge (\neg x_{\{E\}} \vee x_E) \\
& \wedge (x_E \vee x_{\{D\}}) \wedge (\neg x_{\{D\}} \vee x_D) \\
& \wedge (\neg x_A \vee \neg x_B \vee \neg x_C) \wedge (\neg x_A \vee \neg x_D) \wedge (\neg x_C \vee \neg x_E) \wedge (\neg x_D \vee \neg x_E)
\end{aligned}$$

$\Phi_{\mathsf{CQA}}(q(a))$ is satisfiable (assign $x_{\{A,C\}}$, x_A and x_C to true and x_B to false), and indeed $q(a)$ is not entailed under CQA based on standard repairs because of $\mathcal{R}_2$. On the other hand, $q(a)$ holds under P-CQA and $\Phi_{\mathsf{P\text{-}CQA}}(q(a))$ is indeed unsatisfiable: to satisfy $\varphi^{\succ}_{\neg\mathcal{C}}$ (first line), we need to assign x_A and x_C to true, which implies that x_D and x_E are assigned to false to satisfy $\varphi_{\mathsf{cons}}(F')$ (last line), but assigning x_E to false implies that x_D must be assigned to true to satisfy $(x_E \vee x_{\{D\}}) \wedge (\neg x_{\{D\}} \vee x_D)$. Note that $\Phi_{\mathsf{P\text{-}CQA}}(q(a))$ is much bigger than $\Phi_{\mathsf{CQA}}(q(a))$: this is because we need to take into account all facts reachable from $A(a)$ and $C(a)$ to ensure Pareto-optimality. However, in practice, the size of both kinds of encodings remains typically much smaller than the dataset size.

7.2 Tractable semantics and approximations

Several methods have been proposed for tractable semantics, in particular for the intersection semantics and languages for which CQ answering is UCQ-rewritable. Indeed, UCQ-rewritability ensures that the size of the conflicts and causes can be bounded independently from the dataset, hence that CQ answering under intersection semantics is FO-rewritable (cf. Proposition 83 and Corollary 84). The methods for intersection semantics in this tractable case fall into two categories:

- Data-independent rewritings: compute $\varphi^{\cap}_{\mathcal{T},q}$ such that for every $\mathcal{D}$, q is entailed by $\langle \mathcal{D}, \mathcal{T} \rangle$ under intersection iff $\mathcal{I}_{\mathcal{D}} \models \varphi^{\cap}_{\mathcal{T},q}$. For example, if we extend the ontology of Example 15 with $B \sqsubseteq Q$ and $E \sqsubseteq Q$ and consider the query $q(x) = Q(x)$, an FO-rewriting of $q(x)$ w.r.t. $\mathcal{T}$ is $\varphi^{\cap}_{\mathcal{T},q}(x) = Q(x) \vee (B(x) \wedge \neg A(x)) \vee (B(x) \wedge \neg C(x)) \vee (E(x) \wedge \neg C(x) \wedge \neg D(x))$.
- Methods based on the computation of $\mathit{Conf}(\mathcal{K})$, which can be computed in polynomial time w.r.t. data complexity and used in different manners:
  - evaluate the queries on $\langle \mathcal{R}_{\cap}, \mathcal{T} \rangle$ where $\mathcal{R}_{\cap} = \bigcap_{\mathcal{R} \in S\text{-}Rep(\mathcal{K})} \mathcal{R} = \mathcal{D} \setminus \bigcup_{\mathcal{C} \in \mathit{Conf}(\mathcal{K})} \mathcal{C}$;
  - annotate the facts in $\mathcal{D}$ that belong to some conflicts and rewrite the queries to ignore the annotated facts;
  - compute the query causes in polynomial time w.r.t. data complexity and check whether some cause is such that none of its fact belongs to some conflict.

Many systems that implement intractable semantics use tractable approximations to reserve the use of solvers for the complex cases. For example, when computing answers to a query under CQA semantics (based on standard repairs) EQUIP, CQAPri, CAvSAT and ORBITS all use the intersection semantics to get a subset of the answers that hold under CQA in polynomial time, before checking whether the other potential answers hold via translation to BIP or SAT. Even in the case of preferred repairs, for which all semantics are intractable, CQAPri and ORBITS use as a tractable under-approximation of Pareto- or completion-optimal repair-based intersection (hence also of CQA and brave) semantics the set of answers that have some cause $\mathcal{C}$ such that for every $\alpha \in \mathcal{C}$, there is no $\mathcal{B} \rightsquigarrow_{\succ} \alpha$.

8 Summary and outlook

We have seen that many inconsistency-tolerant semantics have been proposed to handle inconsistent databases or knowledge bases, most of them being based on repairs, and surveyed the different kinds of preferred repairs that can be used to obtain more accurate answers when some preference information is available. We presented the properties of these preferred repairs, their relationships and how they relate to other formalisms for inconsistency handling (such as abstract argumentation). We also summarized some results on the computational complexity of reasoning with (preferred) repairs and gave a brief overview of existing implementations. We conclude these lecture notes by mentioning some recent related work on repair-based semantics and research directions.

Extensions of repair-based semantics to new settings. Repairs, CQA, brave and intersection semantics have been defined for RDF graphs that do not validate a set of SHACL (Shapes Constraint Language) constraints [5, 6]. SHACL is related to DLs but makes the closed-world assumption. In this case, a repair is obtained by deleting some facts and/or adding some facts from a given set of hypotheses. Both subset-minimal and cardinality-minimal repairs, corresponding to Δ- and $\leq$-optimal Δ-repairs, have been considered. Repairs and CQA have also been studied in the related setting of graph databases [21, 3], and preferred repairs (based on weights or on a partial order over the set of edge labels and data values) have started to be considered in this context [2]. Another relevant setting is that of temporal databases or KBs [66, 45, 38], which includes a large number of frameworks, since the dataset facts may be annotated with time points or time intervals, the constraints may be atemporal or temporal (including rigid predicates whose interpretation is required not to change over time), and several temporal query languages have been designed. Several definitions of repair have been proposed in these contexts. When facts are annotated with time points and constraints are atemporal, we can repair each snapshot individually, possibly taking into account predicate rigidity [45], or use a more global approach to favor fact persistence (so that, e.g., two identical consecutive snapshots are repaired in the same way) [66]. When time intervals and temporal constraints are used, three notions of repair have been defined, based upon deleting whole facts, seing intervals as sets of time points and deleting punctual facts, or minimally shrinking the time intervals of facts [38]. Finally, CQA has recently been extended to databases whose facts are annotated with values from a naturally ordered positive semiring [94]. In this setting, the answer to a Boolean query is a semiring value, and the consistent answer of such query is defined as the minimum of the semiring values that the query takes over all repairs.

Reasoning tasks beyond query answering. The question of explaining the result of a query attracted a lot of attention, both in the database and KB settings, and is all the more crucial when dealing with repair-based semantics, since an answer can be true under a given semantics and not under another one. Some work in this direction has been conducted for DL and Datalog$^{\pm}$ KBs, with explanations defined as (sets of) subsets of the dataset [34, 35, 105, 106]. Preferences among these explanations (smallest explanations...) have also been considered. Research effort is still needed to make such data-centric explanations more understandable, e.g., by relating them to the ontology and presenting them in a concise way. Moreover, while the existing definitions of explanations could be straightforwardly extended to preferred repair-based semantics, they would probably not be useful to understand the query result by themselves. For example, a cause is an explanation for query entailment under the brave semantics, but while in the case of standard repairs, every cause can be extended to a repair, we would need to explain why a cause belongs to some preferred repair to justify entailment under preferred repair-based brave semantics. Another related reasoning task that attracted some attention in the context of inconsistent KBs is that of abduction, i.e., the task of minimally extending the dataset so that some query is entailed under intersection semantics [75].

References

1 Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

2 Sergio Abriola, Santiago Cifuentes, Nina Pardal, and Edwin Pin Baque. Data-graph repairs: the preferred approach. In *Proceedings of the 15th Alberto Mendelzon International Workshop on Foundations of Data Management (AMW)*, 2023.

3 Sergio Abriola, María Vanina Martínez, Nina Pardal, Santiago Cifuentes, and Edwin Pin Baque. On the complexity of finding set repairs for data-graphs. *Journal of Artificial Intelligence Research (JAIR)*, 76:721–759, 2023. doi:10.1613/JAIR.1.13994.

4 Foto N. Afrati and Phokion G. Kolaitis. Repair checking in inconsistent databases: algorithms and complexity. In *Proceedings of the 12th International Conference on Database Theory (ICDT)*, 2009.

5 Shqiponja Ahmetaj, Robert David, Magdalena Ortiz, Axel Polleres, Bojken Shehu, and Mantas Simkus. Reasoning about explanations for non-validation in SHACL. In *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 2021.

6 Shqiponja Ahmetaj, Timo Camillo Merkl, and Reinhard Pichler. Consistent query answering over SHACL constraints. In *Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 2024.

7 Leila Amgoud and Claudette Cayrol. Inferring from inconsistency in preference-based argumentation frameworks. *Journal of Automated Reasoning (JAR)*, 29(2):125–169, 2002. doi:10.1023/A:1021603608656.

8 Marcelo Arenas, Leopoldo E. Bertossi, and Jan Chomicki. Consistent query answers in inconsistent databases. In *Proceedings of the 18th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS)*, 1999.

9 Marcelo Arenas, Leopoldo E. Bertossi, and Jan Chomicki. Answer sets for consistent query answering in inconsistent databases. *Theory and Practice of Logic Programming (TPLP)*, 3(4-5):393–424, 2003. doi:10.1017/S1471068403001832.

10 Abdallah Arioua, Madalina Croitoru, and Srdjan Vesic. Logic-based argumentation with existential rules. *International Journal of Approximate Reasoning*, 90:76–106, 2017. doi:10.1016/J.IJAR.2017.07.004.

11 Sebastian Arming, Reinhard Pichler, and Emanuel Sallinger. Complexity of repair checking and consistent query answering. In *Proceedings of the 19th International Conference on Database Theory (ICDT)*, 2016.

12 Alessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyaschev. The DL-Lite family and relations. *Journal of Artificial Intelligence Research (JAIR)*, 36:1–69, 2009. doi:10.1613/JAIR.2820.

13 Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, 2005.

14 Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

15 Franz Baader, Patrick Koopmann, and Francesco Kriegel. Optimal repairs in the description logic $\mathcal{EL}$ revisited. In *Proceedings of the 18th European Conference on Logics in Artificial Intelligence (JELIA)*, 2023.

16 Franz Baader, Patrick Koopmann, Francesco Kriegel, and Adrian Nuradiansyah. Computing optimal repairs of quantified ABoxes w.r.t. static $\mathcal{EL}$ TBoxes. In *Proceedings of the 28th International Conference on Automated Deduction (CADE)*, 2021.

17 Franz Baader and Francesco Kriegel. Pushing optimal ABox repair from $\mathcal{EL}$ towards more expressive Horn-DLs. In *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 2022.

18 Franz Baader, Francesco Kriegel, and Adrian Nuradiansyah. Inconsistency- and error-tolerant reasoning w.r.t. optimal repairs of $\mathcal{EL}^{\perp}$ ontologies. In *Proceedings of the 13th International Symposium on Foundations of Information and Knowledge Systems (FoIKS)*, 2024.

19 Jean-François Baget, Salem Benferhat, Zied Bouraoui, Madalina Croitoru, Marie-Laure Mugnier, Odile Papini, Swan Rocher, and Karim Tabia. A general modifier-based framework for inconsistency-tolerant query answering. In *Proceedings of the 15th International Conference Principles of Knowledge Representation and Reasoning (KR)*, 2016.

20 Jean-François Baget, Michel Leclère, Marie-Laure Mugnier, and Eric Salvat. On rules with existential variables: Walking the decidability line. *Artificial Intelligence (AIJ)*, 175(9-10):1620–1654, 2011. doi:10.1016/J.ARTINT.2011.03.002.

21 Pablo Barceló, Gaëlle Fontaine, Sylvain Salvati, and Sophie Tison. Corrections to "On the data complexity of consistent query answering over graph databases [Journal of Computer and

System Sciences 88 (2017) 164-194]". *Journal of Computer and System Sciences*, 155:103694, 2026. doi:10.1016/J.JCSS.2025.103694.

22 Sihem Belabbes, Salem Benferhat, and Jan Chomicki. Handling inconsistency in partially preordered ontologies: the Elect method. *Journal of Logic and Computation*, 31(5):1356–1388, 2021. doi:10.1093/LOGCOM/EXAB024.

23 Salem Benferhat, Zied Bouraoui, Madalina Croitoru, Odile Papini, and Karim Tabia. Non-objection inference for inconsistency-tolerant query answering. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.

24 Salem Benferhat, Zied Bouraoui, and Karim Tabia. How to select one preferred assertional-based repair from inconsistent and prioritized DL-Lite knowledge bases? In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

25 Leopoldo E. Bertossi. *Database Repairing and Consistent Query Answering.* Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2011. doi:10.2200/S00379ED1V01Y201108DTM020.

26 Leopoldo E. Bertossi. Database repairs and consistent query answering: Origins and further developments. In *Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS)*, 2019.

27 Meghyn Bienvenu. On the complexity of consistent query answering in the presence of simple ontologies. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*, 2012.

28 Meghyn Bienvenu. A short survey on inconsistency handling in ontology-mediated query answering. *Künstliche Intelligenz*, 34(4):443–451, 2020. doi:10.1007/S13218-020-00680-9.

29 Meghyn Bienvenu and Camille Bourgaux. Inconsistency-tolerant querying of description logic knowledge bases. In *Tutorial Lectures of the 12th International Reasoning Web Summer School (RW)*, 2016.

30 Meghyn Bienvenu and Camille Bourgaux. Querying and repairing inconsistent prioritized knowledge bases: Complexity analysis and links with abstract argumentation. In *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 2020.

31 Meghyn Bienvenu and Camille Bourgaux. Querying inconsistent prioritized data with ORBITS: Algorithms, implementation, and experiments. In *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 2022.

32 Meghyn Bienvenu and Camille Bourgaux. Inconsistency handling in prioritized databases with universal constraints: Complexity analysis and links with active integrity constraints. In *Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 2023.

33 Meghyn Bienvenu, Camille Bourgaux, and François Goasdoué. Querying inconsistent description logic knowledge bases under preferred repair semantics. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*, 2014.

34 Meghyn Bienvenu, Camille Bourgaux, and François Goasdoué. Explaining inconsistency-tolerant query answering over description logic knowledge bases. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)*, 2016.

35 Meghyn Bienvenu, Camille Bourgaux, and François Goasdoué. Computing and explaining query answers over inconsistent DL-Lite knowledge bases. *Journal of Artificial Intelligence Research (JAIR)*, 64:563–644, 2019. doi:10.1613/JAIR.1.11395.

36 Meghyn Bienvenu, Camille Bourgaux, and Robin Jean. Cost-based semantics for querying inconsistent weighted knowledge bases. In *Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 2024.

37 Meghyn Bienvenu, Camille Bourgaux, and Robin Jean. A rule-based approach to specifying preferences over conflicting facts and querying inconsistent knowledge bases. In *Proceedings of the 22nd International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 2025.

38 Meghyn Bienvenu, Camille Bourgaux, and Atefe Khodadaditaghanaki. Inconsistency handling in DatalogMTL. In *Proceedings of the 34th International Joint Conference on Artificial Intelligence (IJCAI)*, 2025.

39 Meghyn Bienvenu and Pierre Bourhis. Mixed-world reasoning with existential rules under active-domain semantics. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, 2019.

40 Meghyn Bienvenu and Magdalena Ortiz. Ontology-mediated query answering with data-tractable description logics. In *Tutorial Lectures of the 11th Reasoning Web International Summer School (RW)*, 2015.

41 Meghyn Bienvenu and Riccardo Rosati. 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.

42 Bart Bogaerts and Luís Cruz-Filipe. Fixpoint semantics for active integrity constraints. *Artificial Intelligence (AIJ)*, 255:43–70, 2018. doi:10.1016/J.ARTINT.2017.11.003.

43 Philip Bohannon, Michael Flaster, Wenfei Fan, and Rajeev Rastogi. A cost-based model and effective heuristic for repairing constraints by value modification. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 2005.

44 Camille Bourgaux. *Inconsistency handling in ontology-mediated query answering. (Gestion des incohérences pour l'accès aux données en présence d'ontologies)*. PhD thesis, University of Paris-Saclay, France, 2016. URL: https://tel.archives-ouvertes.fr/tel-01378723.

45 Camille Bourgaux, Patrick Koopmann, and Anni-Yasmin Turhan. Ontology-mediated query answering over temporal and inconsistent data. *Semantic Web*, 10(3):475–521, 2019. doi:10.3233/SW-180337.

46 Gerhard Brewka, Ilkka Niemelä, and Miroslaw Truszczynski. Answer set optimization. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, 2003.

47 Samuel R. Buss and Louise Hay. On truth-table reducibility to SAT. *Information and Computation*, 91(1):86–102, 1991. doi:10.1016/0890-5401(91)90075-D.

48 Marco Calautti, Marco Console, and Andreas Pieris. Counting database repairs under primary keys revisited. In *Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS)*, 2019.

49 Marco Calautti, Marco Console, and Andreas Pieris. Benchmarking approximate consistent query answering. In *Proceedings of the 40th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS)*, 2021.

50 Marco Calautti, Sergio Greco, Cristian Molinaro, and Irina Trubitsyna. Preference-based inconsistency-tolerant query answering under existential rules. *Artificial Intelligence (AIJ)*, 312:103772, 2022. doi:10.1016/J.ARTINT.2022.103772.

51 Marco Calautti, Leonid Libkin, and Andreas Pieris. An operational approach to consistent query answering. In *Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS)*, 2018.

52 Marco Calautti, Ester Livshits, Andreas Pieris, and Markus Schneider. Counting database repairs entailing a query: The case of functional dependencies. In *Proceedings of the 41st ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS)*, 2022.

53 Marco Calautti, Ester Livshits, Andreas Pieris, and Markus Schneider. Combined approximations for uniform operational consistent query answering. *Proceedings of the ACM on Management of Data (PACMMOD)*, 2(2):99, 2024. doi:10.1145/3651600.

54 Andrea Calì, Georg Gottlob, and Michael Kifer. Taming the infinite chase: Query answering under expressive relational constraints. *Journal of Artificial Intelligence Research (JAIR)*, 48:115–174, 2013. doi:10.1613/JAIR.3873.

55 Andrea Calì, Georg Gottlob, and Thomas Lukasiewicz. Datalog extensions for tractable query answering over ontologies. In *Semantic Web Information Management - A Model-Based Perspective*, pages 249–279. 2009. doi:10.1007/978-3-642-04329-1_12.

56 Andrea Calì, Georg Gottlob, and Thomas Lukasiewicz. A general Datalog-based framework for tractable query answering over ontologies. *Journal of Web Semantics*, 14:57–83, 2012. doi:10.1016/J.WEBSEM.2012.03.001.

57 Andrea Calì, Georg Gottlob, and Andreas Pieris. Towards more expressive ontology languages: The query answering problem. *Artificial Intelligence (AIJ)*, 193:87–128, 2012. doi:10.1016/J.ARTINT.2012.08.002.

58 Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *Journal of Automated Reasoning (JAR)*, 2007.

59 Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Data complexity of query answering in description logics. In *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 2006.

60 Nofar Carmeli, Martin Grohe, Benny Kimelfeld, Ester Livshits, and Muhammad Tibi. Database repairing with soft functional dependencies. *ACM Transactions on Database Systems*, 49(2):8:1–8:34, 2024. doi:10.1145/3651156.

61 Luciano Caroprese, Sergio Greco, and Ester Zumpano. Active integrity constraints for database consistency maintenance. *IEEE Transactions on Knowledge and Data Engineering*, 21(7):1042–1058, 2009. doi:10.1109/TKDE.2008.226.

62 Luciano Caroprese and Miroslaw Truszczynski. Active integrity constraints and revision programming. *Theory and Practice of Logic Programming (TPLP)*, 11(6):905–952, 2011. doi:10.1017/S1471068410000475.

63 Ashok K. Chandra and Philip M. Merlin. Optimal implementation of conjunctive queries in relational data bases. In *Proceedings of the 9th Annual ACM Symposium on Theory of Computing (STOC)*, 1977.

64 Jan Chomicki, Jerzy Marcinkowski, and Slawomir Staworko. Computing consistent query answers using conflict hypergraphs. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*, 2004.

65 Jan Chomicki, Jerzy Marcinkowski, and Slawomir Staworko. 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)*, 2004.

66 Jan Chomicki and Jef Wijsen. Consistent query answering for atemporal constraints over temporal databases. In *Proceedings of the 23rd International Symposium on Temporal Representation and Reasoning (TIME)*, 2016.

67 Madalina Croitoru and Srdjan Vesic. What can argumentation do for inconsistent ontology query answering? In *Proceedings of the 7th International Conference on Scalable Uncertainty Management (SUM)*, 2013.

68 Luís Cruz-Filipe. Grounded fixpoints and active integrity constraints. In *Technical Communications of the 32nd International Conference on Logic Programming (ICLP)*, 2016.

69 Luís Cruz-Filipe, Graça Gaspar, Patrícia Engrácia, and Isabel Nunes. Computing repairs from active integrity constraints. In *Proceedings of the 7th International Symposium on Theoretical Aspects of Software Engineering (TASE)*, 2013.

70 Evgeny Dantsin, Thomas Eiter, Georg Gottlob, and Andrei Voronkov. Complexity and expressive power of logic programming. *ACM Computing Surveys*, 33(3):374–425, 2001. doi:10.1145/502807.502810.

71 Akhil A. Dixit and Phokion G. Kolaitis. A SAT-based system for consistent query answering. In *Proceedings of the 22nd International Conference on Theory and Applications of Satisfiability Testing (SAT)*, 2019.

72 Akhil A. Dixit and Phokion G. Kolaitis. CAvSAT: Answering aggregation queries over inconsistent databases via SAT solving. In *Proceedings of the International Conference on Management of Data (SIGMOD)*, 2021.

73 Jianfeng Du, Guilin Qi, and Yi-Dong Shen. Weight-based consistent query answering over inconsistent $\mathcal{SHIQ}$ knowledge bases. *Knowledge and Information Systems*, 34(2):335–371, 2013.

74 Jianfeng Du, Yi-Dong Shen, and Guilin Qi. Decomposition-based consistent query answering in databases with universal constraints under cardinality-based semantics. In *Proceedings of the International Conference on Computational Intelligence and Software Engineering (CISE)*, 2010.

75 Jianfeng Du, Kewen Wang, and Yi-Dong Shen. Towards tractable and practical ABox abduction over inconsistent description logic ontologies. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

76 Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence (AIJ)*, 77(2):321–358, 1995. `doi:10.1016/0004-3702(94)00041-X`.

77 Thomas Eiter, Michael Fink, Gianluigi Greco, and Domenico Lembo. Repair localization for query answering from inconsistent databases. *ACM Transactions on Database Systems*, 33(2):10:1–10:51, 2008. `doi:10.1145/1366102.1366107`.

78 Thomas Eiter, Thomas Lukasiewicz, and Livia Predoiu. Generalized consistent query answering under existential rules. In *Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 2016.

79 Zhiwei Fan, Paraschos Koutris, Xiating Ouyang, and Jef Wijsen. LinCQA: Faster consistent query answering with linear time guarantees. *Proceedings of the ACM on Management of Data (PACMMOD)*, 1(1):38:1–38:25, 2023. `doi:10.1145/3588718`.

80 Sergio Flesca, Sergio Greco, and Ester Zumpano. Active integrity constraints. In *Proceedings of the 6th International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming (PPDP)*, 2004.

81 Giorgos Flouris and Antonis Bikakis. A comprehensive study of argumentation frameworks with sets of attacking arguments. *International Journal of Approximate Reasoning*, 109:55–86, 2019. `doi:10.1016/J.IJAR.2019.03.006`.

82 Ariel Fuxman, Elham Fazli, and Renée J. Miller. Conquer: Efficient management of inconsistent databases. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, 2005.

83 Ariel Fuxman and Renée J. Miller. First-order query rewriting for inconsistent databases. In *Proceedings of the 10th International Conference on Database Theory (ICDT)*, 2005.

84 Birte Glimm, Carsten Lutz, Ian Horrocks, and Ulrike Sattler. Conjunctive query answering for the description logic $\mathcal{SHIQ}$. *Journal of Artificial Intelligence Research (JAIR)*, 31:157–204, 2008. `doi:10.1613/JAIR.2372`.

85 Nikos Gorogiannis and Anthony Hunter. Instantiating abstract argumentation with classical logic arguments: Postulates and properties. *Artificial Intelligence (AIJ)*, 175(9-10):1479–1497, 2011. `doi:10.1016/J.ARTINT.2010.12.003`.

86 Gianluigi Greco, Sergio Greco, and Ester Zumpano. A logic programming approach to the integration, repairing and querying of inconsistent databases. In *Proceedings of the 17th International Conference on Logic Programming (ICLP)*, 2001.

87 Gianluigi Greco, Sergio Greco, and Ester Zumpano. A logical framework for querying and repairing inconsistent databases. *IEEE Transactions on Knowledge and Data Engineering*, 15(6):1389–1408, 2003. `doi:10.1109/TKDE.2003.1245280`.

88 Sergio Greco, Cristian Molinaro, and Irina Trubitsyna. Computing approximate query answers over inconsistent knowledge bases. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

89 Lane A. Hemachandra. The strong exponential hierarchy collapses. *Journal of Computer and System Sciences*, 39(3):299–322, 1989. `doi:10.1016/0022-0000(89)90025-1`.

90 Ullrich Hustadt, Boris Motik, and Ulrike Sattler. Data complexity of reasoning in very expressive description logics. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, 2005.

91 Aziz Amezian El Khalfioui, Jonathan Joertz, Dorian Labeeuw, Gaëtan Staquet, and Jef Wijsen. Optimization of answer set programs for consistent query answering by means of first-order rewriting. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM)*, 2020.

92 Benny Kimelfeld, Ester Livshits, and Liat Peterfreund. Detecting ambiguity in prioritized database repairing. In *Proceedings of the 20th International Conference on Database Theory (ICDT)*, 2017.

93 Benny Kimelfeld, Ester Livshits, and Liat Peterfreund. Counting and enumerating preferred database repairs. *Theoretical Computer Science*, 837:115–157, 2020. doi:10.1016/J.TCS.2020.05.016.

94 Phokion G. Kolaitis, Nina Pardal, Jonni Virtema, and Jef Wijsen. Rewriting consistent answers on annotated data. *Proceedings of the ACM on Management of Data (PACMMOD)*, 3(2):110:1–110:26, 2025. doi:10.1145/3725247.

95 Phokion G. Kolaitis, Enela Pema, and Wang-Chiew Tan. Efficient querying of inconsistent databases with binary integer programming. *Proceedings of the VLDB Endowment*, 6(6):397–408, 2013. doi:10.14778/2536336.2536341.

96 Mark W. Krentel. The complexity of optimization problems. *Journal of Computer and System Sciences*, 36(3):490–509, 1988. doi:10.1016/0022-0000(88)90039-6.

97 Ahmed Laouar, Sihem Belabbes, and Salem Benferhat. Tractable closure-based possibilistic repair for partially ordered DL-Lite ontologies. In *Proceedings of the 18th European Conference Logics in Artificial Intelligence (JELIA)*, 2023.

98 Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. Inconsistency-tolerant semantics for description logics. In *Proceedings of the 4th International Conference on Web Reasoning and Rule Systems (RR)*, 2010.

99 Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. Inconsistency-tolerant query answering in ontology-based data access. *Journal of Web Semantics*, 33:3–29, 2015. doi:10.1016/J.WEBSEM.2015.04.002.

100 Ester Livshits, Benny Kimelfeld, and Sudeepa Roy. Computing optimal repairs for functional dependencies. *ACM Transactions on Database Systems*, 45(1):4:1–4:46, 2020. doi:10.1145/3360904.

101 Ester Livshits, Benny Kimelfeld, and Jef Wijsen. Counting subset repairs with functional dependencies. *Journal of Computer and System Sciences*, 117:154–164, 2021. doi:10.1016/J.JCSS.2020.10.001.

102 Andrei Lopatenko and Leopoldo E. Bertossi. Complexity of consistent query answering in databases under cardinality-based and incremental repair semantics. In *Proceedings of the 11th International Conference on Database Theory (ICDT)*, 2007.

103 Andrei Lopatenko and Leopoldo E. Bertossi. Complexity of consistent query answering in databases under cardinality-based and incremental repair semantics (extended version). *CoRR*, abs/1605.07159, 2016. arXiv:1605.07159.

104 Thomas Lukasiewicz, Enrico Malizia, Maria Vanina Martinez, Cristian Molinaro, Andreas Pieris, and Gerardo I. Simari. Inconsistency-tolerant query answering for existential rules. *Artificial Intelligence (AIJ)*, 307:103685, 2022. doi:10.1016/J.ARTINT.2022.103685.

105 Thomas Lukasiewicz, Enrico Malizia, and Cristian Molinaro. Explanations for inconsistency-tolerant query answering under existential rules. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

106 Thomas Lukasiewicz, Enrico Malizia, and Cristian Molinaro. Explanations for negative query answers under inconsistency-tolerant semantics. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)*, 2022.

107 Thomas Lukasiewicz, Enrico Malizia, and Cristian Molinaro. Complexity of inconsistency-tolerant query answering in Datalog+/- under preferred repairs. In *Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 2023.

108 Thomas Lukasiewicz, Enrico Malizia, and Andrius Vaicenavicius. Complexity of inconsistency-tolerant query answering in Datalog+/- under cardinality-based repairs. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence, (AAAI)*, 2019.

109 Thomas Lukasiewicz, Maria Vanina Martinez, and Gerardo I. Simari. Inconsistency handling in Datalog+/- ontologies. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI)*, 2012.

110 Carsten Lutz. The complexity of conjunctive query answering in expressive description logics. In *Proceedings of the 4th International Joint Conference on Automated Reasoning, (IJCAR)*, 2008.

111 Carsten Lutz, Inanç Seylan, and Frank Wolter. Ontology-mediated queries with closed predicates. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

112 Yasir Mahmood, Markus Hecher, and Axel-Cyrille Ngonga Ngomo. Dung's argumentation framework: Unveiling the expressive power with inconsistent databases. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence (AAAI)*, 2025.

113 Yasir Mahmood, Jonni Virtema, Timon Barlag, and Axel-Cyrille Ngonga Ngomo. Computing repairs under functional and inclusion dependencies via argumentation. In *Proceedings of the 13th International Symposium on Foundations of Information and Knowledge Systems (FoIKS)*, 2024.

114 Marco Manna, Francesco Ricca, and Giorgio Terracina. Consistent query answering via ASP from different perspectives: Theory and practice. *Theory and Practice of Logic Programming (TPLP)*, 13(2):227–252, 2013. `doi:10.1017/S1471068411000640`.

115 Marco Manna, Francesco Ricca, and Giorgio Terracina. Taming primary key violations to query large inconsistent data via ASP. *Theory and Practice of Logic Programming (TPLP)*, 15(4-5):696–710, 2015. `doi:10.1017/S1471068415000320`.

116 Lorenzo Marconi and Riccardo Rosati. Consistent query answering over existential rules with open and closed predicates. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence (AAAI)*, 2025.

117 Mónica Caniupán Marileo and Leopoldo E. Bertossi. The consistency extractor system: Answer set programs for consistent query answering in databases. *Data & Knowledge Engineering*, 69(6):545–572, 2010. `doi:10.1016/J.DATAK.2010.01.005`.

118 Dany Maslowski and Jef Wijsen. A dichotomy in the complexity of counting database repairs. *Journal of Computer and System Sciences*, 79(6):958–983, 2013. `doi:10.1016/J.JCSS.2013.01.011`.

119 Nhung Ngo, Magdalena Ortiz, and Mantas Simkus. Closed predicates in description logics: Results on combined complexity. In *Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 2016.

120 Magdalena Ortiz, Diego Calvanese, and Thomas Eiter. Data complexity of query answering in expressive description logics via tableaux. *Journal of Automated Reasoning (JAR)*, 41(1):61–98, 2008. `doi:10.1007/S10817-008-9102-9`.

121 W3C OWL Working Group. OWL 2 Web Ontology Language: Document Overview, 2009. Available at `http://www.w3.org/TR/owl2-overview/`.

122 Horacio Tellez Perez and Jef Wijsen. Generalized weighted repairs. In *Proceedings of the 14th International Conference on Flexible Query Answering Systems (FQAS)*, 2021.

123 Reinhard Pichler and Sebastian Skritek. The complexity of evaluating tuple generating dependencies. In *Proceedings of the 14th International Conference on Database Theory (ICDT)*, 2011.

124 Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. Linking data to ontologies. *Journal of Data Semantics*, 10:133–173, 2008. doi:10.1007/978-3-540-77688-8_5.

125 Riccardo Rosati. On conjunctive query answering in $\mathcal{EL}$. In *Proceedings of the International Workshop on Description Logics (DL)*, 2007.

126 Riccardo Rosati. On the complexity of dealing with inconsistency in description logic ontologies. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, 2011.

127 Riccardo Rosati, Marco Ruzzi, Mirko Graziosi, and Giulia Masotti. Evaluation of techniques for inconsistency handling in OWL 2 QL ontologies. In *Proceedings of the 11th International Semantic Web Conference (ISWC)*, 2012.

128 Andrea Schaerf. On the complexity of the instance checking problem in concept languages with existential quantification. *Journal of Intelligent Information Systems*, 2(3):265–278, 1993. doi:10.1007/BF00962071.

129 Slawek Staworko, Jan Chomicki, and Jerzy Marcinkowski. Prioritized repairing and consistent query answering in relational databases. *Annals of Mathematics and Artificial Intelligence (AMAI)*, 64(2-3):209–246, 2012. doi:10.1007/S10472-012-9288-8.

130 Balder ten Cate, Gaëlle Fontaine, and Phokion G. Kolaitis. On the data complexity of consistent query answering. *Theory of Computing Systems (TOCS)*, 57(4):843–891, 2015. doi:10.1007/S00224-014-9586-0.

131 Michaël Thomazo. *Conjunctive query answering under existential rules - Decidability, complexity, and algorithms*. PhD thesis, Montpellier 2 University, France, 2013. URL: https://tel.archives-ouvertes.fr/tel-00925722.

132 Stephan Tobies. *Complexity results and practical algorithms for logics in knowledge representation*. PhD thesis, RWTH Aachen University, Germany, 2001. URL: http://sylvester.bth.rwth-aachen.de/dissertationen/2001/082/01_082.pdf.

133 Despoina Trivela, Giorgos Stoilos, and Vasilis Vassalos. A framework and positive results for IAR-answering. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

134 Eleni Tsalapati, Giorgos Stoilos, Giorgos B. Stamou, and George Koletsos. Efficient query answering over expressive inconsistent description logics. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.

135 Moshe Y. Vardi. On the complexity of bounded-variable queries. In *Proceedings of the 14th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, 1995.

136 Guohui Xiao, Diego Calvanese, Roman Kontchakov, Domenico Lembo, Antonella Poggi, Riccardo Rosati, and Michael Zakharyaschev. Ontology-based data access: A survey. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.