Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning
Main Track

Inconsistency Handling in Prioritized Databases with Universal Constraints: Complexity Analysis and Links with Active Integrity Constraints

Meghyn Bienvenu[1] , **Camille Bourgaux**[2]
[1] CNRS, University of Bordeaux, Bordeaux INP, LaBRI, Talence, France
[2] DI ENS, ENS, CNRS, PSL University & Inria, Paris, France
meghyn.bienvenu@labri.fr, camille.bourgaux@ens.fr

Abstract

This paper revisits the problem of repairing and querying inconsistent databases equipped with universal constraints. We adopt symmetric difference repairs, in which both deletions and additions of facts can be used to restore consistency, and suppose that preferred repair actions are specified via a binary priority relation over (negated) facts. Our first contribution is to show how existing notions of optimal repairs, defined for simpler denial constraints and repairs solely based on fact deletion, can be suitably extended to our richer setting. We next study the computational properties of the resulting repair notions, in particular, the data complexity of repair checking and inconsistency-tolerant query answering. Finally, we clarify the relationship between optimal repairs of prioritized databases and repair notions introduced in the framework of active integrity constraints. In particular, we show that Pareto-optimal repairs in our setting correspond to founded, grounded and justified repairs w.r.t. the active integrity constraints obtained by translating the prioritized database. Our study also yields useful insights into the behavior of active integrity constraints.

1 Introduction

When a database is inconsistent w.r.t. the integrity constraints, it is possible to obtain meaningful query answers by adopting the *consistent query answering* (CQA) approach (Arenas, Bertossi, and Chomicki 1999). In a nutshell, the idea is to consider a set of *repairs*, which correspond to those databases that satisfy the constraints and are as close as possible to the original database. An answer is then considered true w.r.t. *CQA semantics* if it holds no matter which repair is chosen, thus embodying the cautious mode of reasoning employed in many KR contexts. The CQA approach was subsequently extended to the setting of ontology-mediated query answering, which led to the proposal of other natural repair-based semantics, such as the *brave semantics*, which considers as true those answers that hold in at least one repair (Bienvenu and Rosati 2013), and the *intersection (or IAR) semantics*, which evaluates queries w.r.t. the intersection of the repairs (Lembo et al. 2010). There is now an extensive literature on CQA and other forms of inconsistency-tolerant query answering, (Bertossi 2019) and (Bienvenu 2020) provide recent surveys for the database and ontology settings.

Several different notions of repair have been considered, depending on the considered class of constraints and the allowed repair actions. For denial constraints (such as functional dependencies, FDs) and constraints given by ontologies, consistency can be restored only by removing information, so subset repairs based upon fact deletions are the most common choice. For richer classes of constraints, however, it makes sense to consider *symmetric difference repairs* obtained using both fact additions and deletions. This is the case for the *universal constraints* considered in the present paper, which can be used to express data completeness assumptions and other kinds of domain knowledge. For example, in a hospital setting, a universal constraint can be used to capture expert knowledge that a patient cannot receive a certain treatment without a positive test for a given mutation, with violations indicating either an erroneous treatment record or missing test result. Universal constraints are one of the most expressive classes of first-order constraints for which CQA with symmetric difference repairs is decidable, albeit intractable: Π^p_2-complete w.r.t. data complexity (Staworko and Chomicki 2010; Arming, Pichler, and Sallinger 2016). Despite this high complexity, there have been some prototype implementations using logic programming (Eiter et al. 2008; Marileo and Bertossi 2010).

Repairs can be further refined by taking into account information about the relative reliability of the database facts. In the framework of *prioritized databases* (Staworko, Chomicki, and Marcinkowski 2012), a binary *priority relation* indicates preferences between pair of facts involved in some violation of a denial constraint. Three kinds of *optimal repair* (Pareto-, globally-, and completion-optimal) are then defined to select the most preferred subset repairs according to the priority relation. The complexity of reasoning with these three kinds of optimal repair has been investigated, primarily focusing on databases with FDs (Fagin, Kimelfeld, and Kolaitis 2015; Kimelfeld, Livshits, and Peterfreund 2017; Livshits and Kimelfeld 2017), but also in the context of description logic knowledge bases (Bienvenu and Bourgaux 2020). A recent system implements SAT-based algorithms for optimal repair-based semantics having (co)NP-complete data complexity (Bienvenu and Bourgaux 2022).

To the best of our knowledge, there has been no work addressing how to define fact-level preferences for databases with universal constraints and how to exploit such preferences to single out the optimal symmetric difference repairs. Our first contribution is thus an extension of the framework of prioritized databases to the case of universal constraints and symmetric difference repairs. By carefully defining a suitable notion of conflict (which may involve negative facts), we are able to faithfully lift existing notions of optimal repairs and optimal repair-based semantics, while retaining many properties of the original framework.

We next study the computational properties of optimal repairs of prioritized databases with universal constraints. We provide an almost-complete picture of the data complexity of repair checking and inconsistency-tolerant query answering for each of the three notions of optimal repair (Pareto, global, and completion) and three repair-based semantics (CQA, brave, and intersection). Our results show that adopting optimal repairs does not increase the complexity of inconsistency-tolerant query answering.

Our third contribution is to establish connections with active integrity constraints (AICs), 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 (Flesca, Greco, and Zumpano 2004; Caroprese et al. 2006; Caroprese, Greco, and Zumpano 2009). More precisely, we provide a natural translation from prioritized databases to AICs and observe that Pareto-optimal repairs coincide with three kinds of repairs (founded, grounded and justified) that have been defined for AICs. This leads us to explore more general conditions under which AIC repair notions coincide, which we subsequently exploit to exhibit a translation of certain 'well-behaved' sets of AICs into prioritized databases.

Proofs can be found in (Bienvenu and Bourgaux 2023).

2 Preliminaries

We assume familiarity with propositional and first-order logic (FOL) and provide here terminology and notation for databases, conjunctive queries, constraints, and repairs.

Relational databases Let $\mathbf{C}$ and $\mathbf{V}$ be two disjoint countably infinite sets of constants and variables respectively. A (relational) *schema* $\mathbf{S}$ is a finite set of relation names (or *predicates*), each with an associated arity $n > 0$. A *fact* over $\mathbf{S}$ is an expression of the form $P(c_1, \dots, c_n)$ where $P \in \mathbf{S}$ has arity n and $c_1, \dots, c_n \in \mathbf{C}$. A *database (instance)* over $\mathbf{S}$ is a finite set $\mathcal{D}$ of facts over $\mathbf{S}$. The *active domain* of $\mathcal{D}$, denoted $\mathsf{dom}(\mathcal{D})$, is the set of constants occurring in $\mathcal{D}$.

A database $\mathcal{D}$ can also be viewed as a finite relational structure whose domain is $\mathsf{dom}(\mathcal{D})$ and which interprets each predicate $P \in \mathbf{S}$ as the set $\{\mathbf{c} \mid P(\mathbf{c}) \in \mathcal{D}\}$. We shall use the standard notation $\mathcal{D} \models \Phi$ to indicate that a (set of) FOL sentence(s) Φ is satisfied in this structure.

Conjunctive queries A *conjunctive query* (CQ) is a conjunction of *relational atoms* $P(t_1, \dots, t_n)$ (with each $t_i \in \mathbf{V} \cup \mathbf{C}$), where some variables may be existentially quantified. A *Boolean* CQ (BCQ) has no free variables. Given a query $q(\mathbf{x})$, with free variables $\mathbf{x} = (x_1, \dots, x_k)$, and a tuple of constants $\mathbf{a} = (a_1, \dots, a_k)$, $q(\mathbf{a})$ denotes the BCQ obtained by replacing each variable in $\mathbf{x}$ by the corresponding constant in $\mathbf{a}$. An *answer* to $q(\mathbf{x})$ over a database $\mathcal{D}$ is a tuple of constants $\mathbf{a}$ from $\mathsf{dom}(\mathcal{D})$ such that $\mathcal{D} \models q(\mathbf{a})$.

Constraints A *universal constraint* over a schema $\mathbf{S}$ is a FOL sentence of the form $\forall \mathbf{x}(R_1(\mathbf{t_1}) \land \dots \land R_n(\mathbf{t_n}) \land \neg P_1(\mathbf{u_1}) \land \dots \land \neg P_m(\mathbf{u_m}) \land \varepsilon \rightarrow \bot)$, where each $R_i(\mathbf{t_i})$ (resp. $P_i(\mathbf{u_i})$) is a relational atom over $\mathbf{S}$, ε is a (possibly empty) conjunction of inequality atoms, and $\mathbf{u_1} \cup \dots \cup \mathbf{u_m} \subseteq \mathbf{t_1} \cup \dots \cup \mathbf{t_n}$ (safety condition). Universal constraints can also be written in the form $\forall \mathbf{x}(R_1(\mathbf{t_1}) \land \dots \land R_n(\mathbf{t_n}) \land \varepsilon \rightarrow P_1(\mathbf{u_1}) \lor \dots \lor P_m(\mathbf{u_m}))$. For simplicity, we shall often omit the universal quantification and will sometimes use the generic term *constraint* to mean universal constraint.

Denial constraints are universal constraints of the form $\forall \mathbf{x}(R_1(\mathbf{t_1}) \land \dots \land R_n(\mathbf{t_n}) \land \varepsilon \rightarrow \bot)$, which capture the well-known class of functional dependencies.

We say that a database $\mathcal{D}$ is *consistent* w.r.t. a set of constraints $\mathcal{C}$ if $\mathcal{D} \models \mathcal{C}$. Otherwise, $\mathcal{D}$ is *inconsistent* (w.r.t. $\mathcal{C}$).

A constraint is *ground* if it contains no variables. Given a constraint τ and database $\mathcal{D}$, we use $gr_{\mathcal{D}}(\tau)$ for the set of all ground constraints obtained by (i) replacing variables with constants from $\mathsf{dom}(\mathcal{D})$, (ii) removing all true $c \neq d$ atoms, and (iii) removing all constraints that contain an atom $c \neq c$. We let $gr_{\mathcal{D}}(\mathcal{C}) := \bigcup_{\tau \in \mathcal{C}} gr_{\mathcal{D}}(\tau)$, and note that $\mathcal{D} \models \tau$ iff $\mathcal{D} \models \tau_g$ for every $\tau_g \in gr_{\mathcal{D}}(\tau)$.

Repairs A *symmetric difference repair*, or Δ-repair, of $\mathcal{D}$ w.r.t. $\mathcal{C}$ is a database $\mathcal{R}$ such that (i) $\mathcal{R} \models \mathcal{C}$ and (ii) there is no $\mathcal{R}'$ such that $\mathcal{R}' \models \mathcal{C}$ and $\mathcal{R}' \Delta \mathcal{D} \subsetneq \mathcal{R} \Delta \mathcal{D}$, where Δ is the symmetric difference operator: $S_1 \Delta S_2 = (S_1 \setminus S_2) \cup (S_2 \setminus S_1)$. If only fact deletions are permitted, we obtain *subset repairs* ($\subseteq$-repairs), and if only fact additions are permitted, *superset repairs* ($\supseteq$-repairs). We denote the set of Δ-repairs of $\mathcal{D}$ w.r.t. $\mathcal{C}$ by $SRep(\mathcal{D}, \mathcal{C})$.

Because of the safety condition, an empty database satisfies any set of universal constraints, so every database has at least one $\subseteq$-repair (which is also a Δ-repair), while it may be the case that no $\supseteq$-repair exists. Moreover, for the subclass of denial constraints, Δ-repairs and $\subseteq$-repairs coincide since adding facts cannot resolve a violation of a denial constraint.

3 Optimal Repairs for Universal Constraints

In this section, we show how existing notions of optimal repairs, defined for $\subseteq$-repairs w.r.t. denial constraints, can be lifted to the broader setting of Δ-repairs w.r.t. universal constraints. We then use the resulting repair notions to define inconsistency-tolerant semantics for query answering.

3.1 Conflicts for Universal Constraints

In the setting of denial constraints, a conflict is a minimal subset of the database that is inconsistent w.r.t. the constraints. Conflicts and the associated notion of conflict (hyper)graph underpin many results and algorithms for consistent query answering, and in particular, they appear in the definition of prioritized databases (Staworko, Chomicki, and Marcinkowski 2012). Our first task will thus be to define a suitable notion of conflict for universal constraints.

An important observation is that the absence of a fact may contribute to the violation of a universal constraint. For this reason, conflicts will contain both facts and negated facts, where $\neg P(\mathbf{c})$ indicates that $P(\mathbf{c})$ is absent. We use $Facts^{\mathbf{S}}_{\mathcal{D}}$ for the set of facts over $\mathbf{S}$ with constants from $\mathsf{dom}(\mathcal{D})$, and let $Lits^{\mathbf{S}}_{\mathcal{D}} = \mathcal{D} \cup \{\neg\alpha \mid \alpha \in Facts^{\mathbf{S}}_{\mathcal{D}} \setminus \mathcal{D}\}$ be the set of *literals* of $\mathcal{D}$. Conflicts can then be defined as minimal sets of literals that necessarily lead to a constraint violation.

Definition 1. *Given a database $\mathcal{D}$ and set of (universal) constraints $\mathcal{C}$, the set $Conf(\mathcal{D},\mathcal{C})$ of* conflicts of $\mathcal{D}$ w.r.t. $\mathcal{C}$ *contains all $\subseteq$-minimal sets $\mathcal{E} \subseteq Lits^{\mathbf{S}}_{\mathcal{D}}$ such that for every database $\mathcal{I}$, if $\mathcal{I} \models \mathcal{E}$, then $\mathcal{I} \not\models \mathcal{C}$.*

Example 1. *Let $\mathcal{D} = \{A(a), B(a)\}$ and $\mathcal{C} = \{\tau_1, \tau_2, \tau_3\}$, where $\tau_1 := A(x) \to C(x)$, $\tau_2 := B(x) \to D(x)$, and $\tau_3 := C(x) \wedge D(x) \to \bot$. It can be verified that*

$$SRep(\mathcal{D},\mathcal{C}) = \{\emptyset, \{A(a), C(a)\}, \{B(a), D(a)\}\}$$

and that the set $Conf(\mathcal{D},\mathcal{C})$ is as follows:

$$\{\{A(a), \neg C(a)\}, \{B(a), \neg D(a)\}, \{A(a), B(a)\}\}$$

The first (resp. second) conflict directly violates τ_1 (resp. τ_2). To see why $\{A(a), B(a)\}$ is also a conflict, consider any database $\mathcal{I}$ such that $\{A(a), B(a)\} \subseteq \mathcal{I}$. Then either $C(a) \notin \mathcal{I}$ or $D(a) \notin \mathcal{I}$, in which case $\mathcal{I}$ violates τ_1 or τ_2, or $\mathcal{I}$ contains both $C(a)$ and $D(a)$, hence violates τ_3.

We also provide two alternative characterizations of conflicts, in terms of the hitting sets of literals removed from Δ-repairs and the prime implicants[1] of the propositional formula stating that there is a constraint violation (treating the elements of $Lits^{\mathbf{S}}_{\mathcal{D}}$ as propositional literals):

Proposition 1. *For every database $\mathcal{D}$ and constraint set $\mathcal{C}$:*

1. *$Conf(\mathcal{D},\mathcal{C}) = \{\mathcal{H} \cap \mathcal{D} \cup \{\neg\alpha \mid \alpha \in \mathcal{H} \setminus \mathcal{D}\} \mid \mathcal{H} \in MHS(\mathcal{D},\mathcal{C})\}$ where $MHS(\mathcal{D},\mathcal{C})$ is the set of all minimal hitting sets of $\{\mathcal{R}\Delta\mathcal{D} \mid \mathcal{R} \in SRep(\mathcal{D},\mathcal{C})\}$.*
2. *$Conf(\mathcal{D},\mathcal{C}) = \{\{\lambda_1, \dots, \lambda_k\} \subseteq Lits^{\mathbf{S}}_{\mathcal{D}} \mid \lambda_1 \wedge \dots \wedge \lambda_k$ is a prime implicant of $\bigvee_{\varphi \to \bot \in gr_{\mathcal{D}}(\mathcal{C})} \varphi\}$.*

We can show that our notion of conflicts enjoy similar properties to conflicts w.r.t. denial constraints, but to formulate them, we must first introduce some useful terminology and notation for moving between databases and sets of literals.

Given a database $\mathcal{D}$ over schema $\mathbf{S}$, a *candidate repair* for $\mathcal{D}$ is a database $\mathcal{B}$ with $\mathcal{B} \subseteq Facts^{\mathbf{S}}_{\mathcal{D}}$. For every candidate repair $\mathcal{B}$ for $\mathcal{D}$, we define its corresponding set of literals $Lits_{\mathcal{D}}(\mathcal{B}) = \mathcal{B} \cup \{\neg\alpha \mid \alpha \in Facts^{\mathbf{S}}_{\mathcal{D}} \setminus \mathcal{B}\}$ and the set of literals $Int_{\mathcal{D}}(\mathcal{B}) = Lits_{\mathcal{D}}(\mathcal{B}) \cap Lits^{\mathbf{S}}_{\mathcal{D}} = (\mathcal{B} \cap \mathcal{D}) \cup \{\neg\alpha \mid \alpha \in Facts^{\mathbf{S}}_{\mathcal{D}} \setminus (\mathcal{B} \cup \mathcal{D})\}$ upon which $\mathcal{B}$ and $\mathcal{D}$ agree. Furthermore, with every subset $\mathcal{B} \subseteq Lits^{\mathbf{S}}_{\mathcal{D}}$ we can associate a candidate repair $Dat_{\mathcal{D}}(\mathcal{B}) = \mathcal{B} \cap \mathcal{D} \cup \{\alpha \mid \neg\alpha \in Lits^{\mathbf{S}}_{\mathcal{D}} \setminus \mathcal{B}\}$. Note that if $\mathcal{B}$ is a candidate repair, $Dat_{\mathcal{D}}(Int_{\mathcal{D}}(\mathcal{B})) = \mathcal{B}$.

Proposition 2. *Let $\mathcal{D}$ be a database, $\mathcal{C}$ a set of universal constraints, and $\mathcal{R}$ a candidate repair for $\mathcal{D}$.*

1. *$\mathcal{R} \in SRep(\mathcal{D},\mathcal{C})$ iff $Int_{\mathcal{D}}(\mathcal{R})$ is a maximal subset of $Lits^{\mathbf{S}}_{\mathcal{D}}$ such that $Dat_{\mathcal{D}}(Int_{\mathcal{D}}(\mathcal{R})) \models \mathcal{C}$, i.e., $\mathcal{R} \models \mathcal{C}$.*
2. *$\mathcal{R} \in SRep(\mathcal{D},\mathcal{C})$ iff $Int_{\mathcal{D}}(\mathcal{R})$ is a maximal subset of $Lits^{\mathbf{S}}_{\mathcal{D}}$ such that $\mathcal{E} \not\subseteq Int_{\mathcal{D}}(\mathcal{R})$ for every $\mathcal{E} \in Conf(\mathcal{D},\mathcal{C})$.*
3. *$\mathcal{R} \in SRep(\mathcal{D},\mathcal{C})$ iff $Int_{\mathcal{D}}(\mathcal{R}) \cap (\bigcup_{\mathcal{E} \in Conf(\mathcal{D},\mathcal{C})} \mathcal{E})$ is a maximal independent set (MIS) of the* conflict hypergraph *$\mathcal{G}^{\mathcal{C}}_{\mathcal{D}}$, whose vertices are the literals from $\bigcup_{\mathcal{E} \in Conf(\mathcal{D},\mathcal{C})} \mathcal{E}$ and whose hyperedges are the conflicts of $\mathcal{D}$ w.r.t. $\mathcal{C}$.*

The first property states that Δ-repairs correspond to the consistent databases that preserve a maximal set of the original literals, while the second rephrases consistency in terms of conflicts. The third generalizes a well-known hypergraph-based characterization of $\subseteq$-repairs. As the next remark explains, an earlier attempt at defining conflicts for universal constraints failed to obtain such a property.

Remark 1. *Staworko and Chomicki (2010) define a conflict as a set of literals obtained by grounding a universal constraint, and the hyperedges of their extended conflict hypergraph $ECG(\mathcal{D},\mathcal{C})$ are either conflicts or 'relevant' pairs of literals $\{\alpha, \neg\alpha\}$. For instance, if we take $\mathcal{D}$ and $\mathcal{C}$ as in Example 1, then $ECG(\mathcal{D},\mathcal{C})$ has hyperedges $\{A(a), \neg C(a)\}, \{B(a), \neg D(a)\}, \{C(a), D(a)\}$, $\{C(a), \neg C(a)\}$ and $\{D(a), \neg D(a)\}$.*

Every repair gives rise to a MIS of $ECG(\mathcal{D},\mathcal{C})$, but a MIS need not correspond to any repair. Proposition 4 in (Staworko and Chomicki 2010) claims a weaker converse: for every MIS M of $ECG(\mathcal{D},\mathcal{C})$, either its positive projection $M^+ = M \cap Facts^{\mathbf{S}}_{\mathcal{D}}$ is a Δ-repair of $\mathcal{D}$ w.r.t. $\mathcal{C}$, or there exists a MIS N of $ECG(\mathcal{D},\mathcal{C})$ such that $N^+\Delta\mathcal{D} \subsetneq M^+\Delta\mathcal{D}$. However, our example disproves this claim, as $M = \{A(a), B(a), C(a)\}$ is a MIS of $ECG(\mathcal{D},\mathcal{C})$, but $M^+ = M$ is not a Δ-repair (it violates τ_2), and there is no MIS N with $N^+\Delta\mathcal{D} \subsetneq M^+\Delta\mathcal{D}$. Essentially, the problem is that their notion of conflicts does not take into account implicit constraints ($A(x) \wedge B(x) \to \bot$ in this example).

To clarify the relationship between the universal and denial constraint settings, we translate the former into the latter. Take a database $\mathcal{D}$ and set of universal constraints $\mathcal{C}$ over schema $\mathbf{S}$. To represent negative literals, we introduce an extended schema $\mathbf{S}' = \mathbf{S} \cup \{\widetilde{P} \mid P \in \mathbf{S}\}$ and a function *facts* that maps sets of literals over $\mathbf{S}$ into sets of facts over $\mathbf{S}'$ by replacing each negative literal $\neg P(\mathbf{c})$ by $\widetilde{P}(\mathbf{c})$. We then consider the database $\mathcal{D}_d = facts(Lits^{\mathbf{S}}_{\mathcal{D}}) = \mathcal{D} \cup \{\widetilde{P}(\mathbf{c}) \mid P(\mathbf{c}) \in Facts^{\mathbf{S}}_{\mathcal{D}} \setminus \mathcal{D}\}$, and the set of ground denial constraints $\mathcal{C}_{d,\mathcal{D}} = \{(\bigwedge_{\alpha \in facts(\mathcal{E})} \alpha) \to \bot \mid \mathcal{E} \in Conf(\mathcal{D},\mathcal{C})\}$.

Proposition 3. *For every database $\mathcal{D}$ and constraint set $\mathcal{C}$: $Conf(\mathcal{D}_d, \mathcal{C}_{d,\mathcal{D}}) = \{facts(\mathcal{E}) \mid \mathcal{E} \in Conf(\mathcal{D},\mathcal{C})\}$ and $SRep(\mathcal{D}_d, \mathcal{C}_{d,\mathcal{D}}) = \{facts(Int_{\mathcal{D}}(\mathcal{R})) \mid \mathcal{R} \in SRep(\mathcal{D},\mathcal{C})\}$.*

One may naturally wonder whether a set of denial constraints $\mathcal{C}_d$ which does not depend on $\mathcal{D}$ could be used in place of $\mathcal{C}_{d,\mathcal{D}}$ in Proposition 3. The answer is no: the existence of such a set $\mathcal{C}_d$ would imply a data-independent bound on the size of conflicts that may appear in any set $Conf(\mathcal{D}_d, \mathcal{C}_d)$, and hence in $Conf(\mathcal{D},\mathcal{C})$. However, as the next example illustrates, universal constraints differ from denial constraints in that the size of the conflicts cannot be bounded independently from the database.

Example 2. *Let $\mathcal{C}$ consist of $R(x,y) \wedge A(x) \to A(y)$ and $A(x) \wedge B(x) \to \bot$. Then for every $n \geq 1$, we can build*

[1]We recall that a prime implicant of a propositional formula ψ is a minimal conjunction of propositional literals κ that entails ψ.

a database $\{A(a_0), R(a_0, a_1), \ldots, R(a_{n-1}, a_n), B(a_n)\}$ *of size* $n+2$ *which is a conflict (of itself) w.r.t.* $\mathcal{C}$.

3.2 Prioritized Databases & Optimal Repairs

With the definition of conflicts in place, we can extend the notion of prioritized database (Staworko, Chomicki, and Marcinkowski 2012) to the setting of universal constraints.

Definition 2. *A* priority relation $\succ$ *for a database* $\mathcal{D}$ *w.r.t. a set of universal constraints* $\mathcal{C}$ *is an acyclic binary relation over the literals of* $Conf(\mathcal{D}, \mathcal{C})$ *such that if* $\lambda \succ \mu$, *then there exists* $\mathcal{E} \in Conf(\mathcal{D}, \mathcal{C})$ *such that* $\{\lambda, \mu\} \subseteq \mathcal{E}$. *We say that* $\succ$ *is* total *if for every pair* $\lambda \neq \mu$ *such that* $\{\lambda, \mu\} \subseteq \mathcal{E}$ *for some* $\mathcal{E} \in Conf(\mathcal{D}, \mathcal{C})$, *either* $\lambda \succ \mu$ *or* $\mu \succ \lambda$. *A* completion *of* $\succ$ *is a total priority relation* $\succ' \supseteq \succ$.

A priority relation $\succ$ *is* score-structured *if there is a scoring function* $s : \bigcup_{\mathcal{E} \in Conf(\mathcal{D},\mathcal{C})} \mathcal{E} \to \mathbb{N}$ *such that for every* $\{\lambda, \mu\} \subseteq \mathcal{E}$ *with* $\mathcal{E} \in Conf(\mathcal{D}, \mathcal{C})$, $\lambda \succ \mu$ *iff* $s(\lambda) > s(\mu)$.

Definition 3. *A* prioritized database $\mathcal{D}^{\mathcal{C}}_{\succ} = (\mathcal{D}, \mathcal{C}, \succ)$ *consists of a database* $\mathcal{D}$, *a set of universal constraints* $\mathcal{C}$, *and a priority relation* $\succ$ *for* $\mathcal{D}$ *w.r.t.* $\mathcal{C}$.

We now extend the definitions of optimal repairs to the case of universal constraints.

Definition 4. *Consider a prioritized database* $\mathcal{D}^{\mathcal{C}}_{\succ} = (\mathcal{D}, \mathcal{C}, \succ)$, *and let* $\mathcal{R} \in SRep(\mathcal{D}, \mathcal{C})$.

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

We say that $\mathcal{R}$ *is:*

- Pareto-optimal *if there is no Pareto improvement of* $\mathcal{R}$.
- globally-optimal *if there is no global improvement of* $\mathcal{R}$.
- completion-optimal *if* $\mathcal{R}$ *is a globally-optimal* Δ*-repair of* $\mathcal{D}^{\mathcal{C}}_{\succ'}$, *for some completion* $\succ'$ *of* $\succ$.

We denote by $GRep(\mathcal{D}^{\mathcal{C}}_{\succ})$, $PRep(\mathcal{D}^{\mathcal{C}}_{\succ})$ *and* $CRep(\mathcal{D}^{\mathcal{C}}_{\succ})$ *the sets of globally-, Pareto- and completion-optimal* Δ*-repairs.*

A Pareto improvement is also a global improvement, so $GRep(\mathcal{D}^{\mathcal{C}}_{\succ}) \subseteq PRep(\mathcal{D}^{\mathcal{C}}_{\succ})$, and a global improvement w.r.t. $\succ$ is a global improvement w.r.t. any completion $\succ'$ of $\succ$, so $CRep(\mathcal{D}^{\mathcal{C}}_{\succ}) \subseteq GRep(\mathcal{D}^{\mathcal{C}}_{\succ})$. Hence, as in the denial constraints case, $CRep(\mathcal{D}^{\mathcal{C}}_{\succ}) \subseteq GRep(\mathcal{D}^{\mathcal{C}}_{\succ}) \subseteq PRep(\mathcal{D}^{\mathcal{C}}_{\succ})$. Moreover, there always exists at least one completion- (hence Pareto- and globally-)optimal Δ-repair, which can be obtained from $\mathcal{G}^{\mathcal{C}}_{\mathcal{D}}$ by the following greedy procedure: while some literal from $Lits^{\mathbf{S}}_{\mathcal{D}}$ has not been considered, pick a literal that is maximal w.r.t. $\succ$ among those not yet considered, and add it to the current set if it does not introduce a conflict from $Conf(\mathcal{D}, \mathcal{C})$. If $\mathcal{B}$ is a subset of $Lits^{\mathbf{S}}_{\mathcal{D}}$ obtained by this procedure, we show that $Dat_{\mathcal{D}}(\mathcal{B}) \in CRep(\mathcal{D}^{\mathcal{C}}_{\succ})$. This procedure requires us to compute $Conf(\mathcal{D}, \mathcal{C})$, hence does not run in polynomial time (unlike the denial constraint case). However, as for denial constraints, we have:

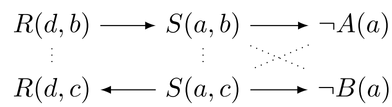
Proposition 4. *If* $\succ$ *is total, then* $|PRep(\mathcal{D}^{\mathcal{C}}_{\succ})| = 1$.

In particular, this means $GRep(\mathcal{D}^{\mathcal{C}}_{\succ}) = PRep(\mathcal{D}^{\mathcal{C}}_{\succ})$ when $\succ$ is total, so we may replace globally-optimal by Pareto-optimal in the definition of completion-optimal Δ-repairs.

Example 3. *Let* $\mathcal{D} = \{S(a, b), S(a, c), R(d, b), R(d, c)\}$, *where* $R(d, b) \succ S(a, b)$, $S(a, b) \succ \neg A(a)$, $S(a, c) \succ R(d, c)$, $S(a, c) \succ \neg B(a)$, *and* $\mathcal{C}$ *contains the constraints:*

$$S(x,y) \land S(x,z) \land y \neq z \to \bot \qquad S(x,y) \to A(x)$$
$$R(x,y) \land R(x,z) \land y \neq z \to \bot \qquad S(x,y) \to B(x)$$
$$R(y,x) \land S(z,x) \to \bot$$

The conflicts are all binary, so the conflict hypergraph is a graph, pictured below. We use an arrow $\lambda \to \mu$ *when* $\lambda \succ \mu$ *and dotted lines for conflicting literals with no priority.*



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

$$\begin{aligned} CRep(\mathcal{D}^{\mathcal{C}}_{\succ}) &= \{\{R(d,b), S(a,c), A(a), B(a)\}\} \\ GRep(\mathcal{D}^{\mathcal{C}}_{\succ}) &= CRep(\mathcal{D}^{\mathcal{C}}_{\succ}) \cup \{\{R(d,b)\}\} \\ PRep(\mathcal{D}^{\mathcal{C}}_{\succ}) &= GRep(\mathcal{D}^{\mathcal{C}}_{\succ}) \cup \{\{R(d,c)\}, \\ &\quad \{R(d,c), S(a,b), A(a), B(a)\}\} \end{aligned}$$

and that $SRep(\mathcal{D}, \mathcal{C}) = PRep(\mathcal{D}^{\mathcal{C}}_{\succ})$.

When $\succ$ is score-structured with scoring function s, we define the *prioritization* of $\bigcup_{\mathcal{E} \in Conf(\mathcal{D},\mathcal{C})} \mathcal{E}$ as the partition $\mathcal{S}_1, \ldots, \mathcal{S}_n$ such that for every $1 \leq i \leq n$, there exists $m \in \mathbb{N}$ such that $\mathcal{S}_i = \{\lambda \mid s(\lambda) = m\}$, and for every $\{\lambda_i, \lambda_j\} \subseteq \mathcal{E} \in Conf(\mathcal{D}, \mathcal{C})$, $\lambda_i \succ \lambda_j$ iff $\lambda_i \in \mathcal{S}_i$, $\lambda_j \in \mathcal{S}_j$ and $i < j$. Intuitively, the more reliable a literal λ the smaller the index of $\mathcal{S}_i$ that contains λ. Bienvenu, Bourgaux, and Goasdoué (2014) introduced a notion of $\subseteq_P$-repair based upon such prioritizations, which we adapt below to Δ-repairs.

Definition 5. *Let* $\mathcal{D}^{\mathcal{C}}_{\succ}$ *be a prioritized database such that* $\succ$ *is score-structured and* $\mathcal{S}_1, \ldots, \mathcal{S}_n$ *is the prioritization of* $\bigcup_{\mathcal{E} \in Conf(\mathcal{D},\mathcal{C})} \mathcal{E}$. *A* Δ_P-repair *of* $\mathcal{D}^{\mathcal{C}}_{\succ}$ *is a candidate repair* $\mathcal{R}$ *such that (i)* $\mathcal{R} \models \mathcal{C}$ *and (ii) there is no* $\mathcal{R}' \models \mathcal{C}$ *such that there is some* $1 \leq i \leq n$ *such that*

- $Int_{\mathcal{D}}(\mathcal{R}) \cap \mathcal{S}_i \subsetneq Int_{\mathcal{D}}(\mathcal{R}') \cap \mathcal{S}_i$ *and*
- *for all* $1 \leq j < i$, $Int_{\mathcal{D}}(\mathcal{R}) \cap \mathcal{S}_j = Int_{\mathcal{D}}(\mathcal{R}') \cap \mathcal{S}_j$.

We denote by $LRep(\mathcal{D}^{\mathcal{C}}_{\succ})$ *the set of* Δ_P*-repairs of* $\mathcal{D}^{\mathcal{C}}_{\succ}$.

As in the case of denial constraints, all four notions of optimal Δ-repairs coincide when $\succ$ is score-structured.

Proposition 5. *If* $\succ$ *is score-structured, then* $CRep(\mathcal{D}^{\mathcal{C}}_{\succ}) = GRep(\mathcal{D}^{\mathcal{C}}_{\succ}) = PRep(\mathcal{D}^{\mathcal{C}}_{\succ}) = LRep(\mathcal{D}^{\mathcal{C}}_{\succ})$.

We can now define variants of existing inconsistency-tolerant semantics based upon our optimal repairs.

Definition 6. *Fix* $X \in \{S, P, G, C\}$ *and consider a prioritized database* $\mathcal{D}^{\mathcal{C}}_{\succ}$, *query* $q(\mathbf{x})$, *and tuple of constants* $\mathbf{a}$ *with* $|\mathbf{x}| = |\mathbf{a}|$. *Then* $\mathbf{a}$ *is an answer to* $q(\mathbf{x})$ *over* $\mathcal{D}^{\mathcal{C}}_{\succ}$

- *under* X-brave semantics*, denoted* $\mathcal{D}^{\mathcal{C}}_{\succ} \models^{X}_{brave} q(\mathbf{a})$*, if* $\mathcal{R} \models q(\mathbf{a})$ *for some* $\mathcal{R} \in XRep(\mathcal{D}^{\mathcal{C}}_{\succ})$*;*
- *under* X-CQA semantics*, denoted* $\mathcal{D}^{\mathcal{C}}_{\succ} \models^{X}_{CQA} q(\mathbf{a})$*, if* $\mathcal{R} \models q(\mathbf{a})$ *for every* $\mathcal{R} \in XRep(\mathcal{D}^{\mathcal{C}}_{\succ})$*;*
- *under* X-intersection semantics*, denoted* $\mathcal{D}^{\mathcal{C}}_{\succ} \models^{X}_{\cap} q(\mathbf{a})$*, if* $\mathcal{B} \models q(\mathbf{a})$ *where* $\mathcal{B} = \bigcap_{\mathcal{R} \in XRep(\mathcal{D}^{\mathcal{C}}_{\succ})} \mathcal{R}$*.*

Just as in the case of denial constraints, these semantics are related as follows:

$$\mathcal{D}^{\mathcal{C}}_{\succ} \models^{X}_{\cap} q(\mathbf{a}) \Rightarrow \mathcal{D}^{\mathcal{C}}_{\succ} \models^{X}_{\text{CQA}} q(\mathbf{a}) \Rightarrow \mathcal{D}^{\mathcal{C}}_{\succ} \models^{X}_{\text{brave}} q(\mathbf{a})$$

Unlike the denial constraint case, the intersection of the optimal Δ-repairs may be inconsistent w.r.t. $\mathcal{C}$. For example, if $\mathcal{D} = \{A(a)\}, \mathcal{C} = \{A(x) \rightarrow B(x) \vee C(x)\}$, $A(a) \succ \neg B(a)$ and $A(a) \succ \neg C(a)$, then $\bigcap_{\mathcal{R} \in PRep(\mathcal{D}^{\mathcal{C}}_{\succ})} \mathcal{R} = \{A(a)\}$ violates the constraint. This is not a problem since we consider conjunctive queries, which are monotone, meaning that if the intersection of the optimal Δ-repairs yields a query answer, then the tuple is an answer in every optimal Δ-repair.

Example 4 (Example 3 cont'd)**.** *Considering the different semantics based upon Pareto-optimal repairs:*

- $\mathcal{D}^{\mathcal{C}}_{\succ} \models^{P}_{brave} A(a)$ *but* $\mathcal{D}^{\mathcal{C}}_{\succ} \not\models^{P}_{CQA} A(a)$*;*
- $\mathcal{D}^{\mathcal{C}}_{\succ} \models^{P}_{CQA} \exists y R(d, y)$ *but* $\mathcal{D}^{\mathcal{C}}_{\succ} \not\models^{P}_{\cap} \exists y R(d, y)$*.*

If we consider now CQA semantics for the different kinds of optimal repairs, we find that:

- $\mathcal{D}^{\mathcal{C}}_{\succ} \models^{C}_{CQA} A(a)$ *but* $\mathcal{D}^{\mathcal{C}}_{\succ} \not\models^{G}_{CQA} A(a)$*;*
- $\mathcal{D}^{\mathcal{C}}_{\succ} \models^{G}_{CQA} R(d, b)$ *but* $\mathcal{D}^{\mathcal{C}}_{\succ} \not\models^{P}_{CQA} R(d, b)$*.*

4 Complexity Analysis

In this section, we analyze the data complexity of the central computational tasks related to optimal repairs. We consider the following decision problems:

- X-repair checking: given a prioritized database $\mathcal{D}^{\mathcal{C}}_{\succ}$ and a candidate repair $\mathcal{R}$, decide whether $\mathcal{R} \in XRep(\mathcal{D}^{\mathcal{C}}_{\succ})$;
- Query answering under X-Sem semantics: given a prioritized database $\mathcal{D}^{\mathcal{C}}_{\succ}$, a query q, and a candidate answer $\mathbf{a}$, decide whether $\mathcal{D}^{\mathcal{C}}_{\succ} \models^{\text{X}}_{\text{Sem}} q(\mathbf{a})$;

where X $\in \{S, P, G, C\}$ and Sem $\in$ {brave, CQA, $\cap$}. We focus on data complexity, which is measured in terms of the size of the database $\mathcal{D}$, treating the constraints $\mathcal{C}$ and query q as fixed and of constant size (under the latter assumption, $\mathcal{R}$ and $\mathbf{a}$ are of polynomial size w.r.t. $\mathcal{D}$). Table 1 summarizes our new results for optimal repairs w.r.t. universal constraints alongside existing results for denial constraints.

Staworko and Chomicki (2010) showed that S-repair checking is coNP-complete in data complexity. We show that the same holds for Pareto- and globally-optimal repairs:

Theorem 1. *X-repair checking is co*NP*-complete in data complexity for X* $\in \{P, G\}$*.*

Proof Sketch. The lower bound is inherited from Δ-repairs. For the upper bounds, we sketch NP procedures for checking whether $\mathcal{R} \notin XRep(\mathcal{D}^{\mathcal{C}}_{\succ})$ for a given candidate repair $\mathcal{R}$. In a nutshell, we guess either (i) 'inconsistent', (ii) 'not maximal' together with another candidate repair $\mathcal{R}'$, or (iii) 'improvement' together with a candidate (Pareto or global) improvement $\mathcal{B}$. In case (i), it suffices to verify in P that $\mathcal{R} \not\models \mathcal{C}$, returning yes if so. In case (ii), we test in P whether $\mathcal{R}' \Delta \mathcal{D} \subsetneq \mathcal{R} \Delta \mathcal{D}$ and $\mathcal{R}' \models \mathcal{C}$, returning yes if both conditions hold. In case (iii), we check in P whether $\mathcal{B}$ is indeed a (Pareto / global) improvement of $\mathcal{R}$, returning yes if so. ☐

		S	P	C	G
Univ.	RC	coNP	coNP	coNP-h, in Σ^p_2	coNP
	Brave	Σ^p_2	Σ^p_2	Σ^p_2	Σ^p_2
	CQA, Int	Π^p_2	Π^p_2	Π^p_2	Π^p_2
Denial	RC	in P	in P	in P	coNP
	Brave	NP	NP	NP	Σ^p_2
	CQA, Int	coNP	coNP	coNP	Π^p_2

Table 1: Data complexity of X-repair checking (RC) and query answering under X-brave (Brave), X-CQA, and X-intersection (Int) semantics (X $\in \{S, P, G, C\}$) w.r.t. universal or denial constraints. Completeness results except where indicated otherwise.

Interestingly, we observe that P-repair checking is hard even if we already know the input is a Δ-repair:

Lemma 1. *Deciding whether a given Δ-repair is Pareto-optimal is co*NP*-complete in data complexity.*

We next turn to C-repair checking. A first idea would be to guess a completion $\succ'$ and check (using an NP oracle) that the input database is Pareto-optimal w.r.t. $\succ'$. However, determining whether the guessed binary relation is a completion is not straightforward, as we must make sure that we relate all and only those literals that appear together in some conflict. As the following result shows, even identifying conflicts is a challenging task for universal contraints:

Lemma 2. *Deciding whether a set of literals belongs to* $Conf(\mathcal{D}, \mathcal{C})$ *is* BH_2*-complete w.r.t. data complexity.*

With a more careful approach, we can show that C-repair checking does belong to Σ^p_2. The exact complexity is open.

Theorem 2. *C-repair checking is co*NP*-hard and in* Σ^p_2 *w.r.t. data complexity.*

Proof Sketch. We use a non-deterministic version of the greedy procedure sketched in Section 3.2: to decide if $\mathcal{R} \in CRep(\mathcal{D}^{\mathcal{C}}_{\succ})$, we guess the order in which literals of $Lits^{\mathbf{S}}_{\mathcal{D}}$ will be considered, and for each $\lambda \in Lits^{\mathbf{S}}_{\mathcal{D}} \setminus Int_{\mathcal{D}}(\mathcal{R})$, we guess a set of literals $L \subseteq Int_{\mathcal{D}}(\mathcal{R})$ that precede λ in the order and such that $L \cup \{\lambda\}$ forms a conflict. ☐

Leveraging our results for repair checking, we can establish the precise data complexity of query answering for all combinations of semantics and optimality notions:

Theorem 3. *Query answering under X-brave (resp. X-CQA and X-intersection) semantics is* Σ^p_2*-complete (resp.* Π^p_2*-complete) in data complexity, for X* $\in \{P, G, C\}$*.*

The lower bounds that are higher for universal constraints than denial constraints involve databases whose conflicts are difficult to compute. This is no coincidence, as we show that if the set of conflicts are available, the complexity drops:

Theorem 4. *If $Conf(\mathcal{D}, \mathcal{C})$ is given and considered as part of the input, then all complexity results for denial constraints listed in Table 1 hold also for universal constraints.*

The lower complexities apply in particular to sets of constraints whose conflicts have bounded size, such as universal constraints with at most two relational atoms. Unfortunately, we show that it is impossible in general to determine whether a given set of constraints has bounded conflicts:

Theorem 5. *Given a set of universal constraints $\mathcal{C}$, it is undecidable to determine whether there exists $k \in \mathbb{N}$ such that for every database $\mathcal{D}$, $\max_{\mathcal{E} \in Conf(\mathcal{D},\mathcal{C})}(|\mathcal{E}|) \leq k$.*

5 Links with Active Integrity Constraints

Active integrity constraints define which update operations are allowed to solve a constraint violation (Flesca, Greco, and Zumpano 2004; Caroprese et al. 2006; Caroprese, Greco, and Zumpano 2009), in the same spririt that prioritized databases express preferred ways of solving conflicts. This section investigates how these two frameworks relate.

5.1 Preliminaries on Active Integrity Constraints

We briefly recall the basics of active integrity constraints, directing readers to (Bogaerts and Cruz-Filipe 2018) for a good overview of the area.

Update actions An *update atom* is of the form $+P(\mathbf{x})$ or $-P(\mathbf{x})$ where $P(\mathbf{x})$ is a relational atom. We use *fix* to map relational literals to the corresponding update atoms: $\mathit{fix}(P(\mathbf{x})) = -P(\mathbf{x})$ and $\mathit{fix}(\neg P(\mathbf{x})) = +P(\mathbf{x})$. 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 database $\mathcal{D}$ is $\mathcal{D} \circ \mathcal{U} := \mathcal{D} \setminus \{\alpha \mid -\alpha \in \mathcal{U}\} \cup \{\alpha \mid +\alpha \in \mathcal{U}\}$.

Active integrity constraints An *active integrity constraint* (AIC) takes the form $r = \ell_1 \wedge \cdots \wedge \ell_n \rightarrow \{A_1, \ldots, A_k\}$, where $\mathsf{body}(r) = \ell_1 \wedge \cdots \wedge \ell_n$ is such that $\tau_r := \mathsf{body}(r) \rightarrow \bot$ is a universal constraint, $\mathsf{upd}(r) = \{A_1, \ldots, A_k\}$ is non-empty, and every A_j is equal to $\mathit{fix}(\ell_i)$ for some ℓ_i. We use $\mathsf{lits}(r)$ for the set of literals appearing in $\mathsf{body}(r)$, and say that $\ell \in \mathsf{lits}(r)$ is *non-updatable* if $\mathit{fix}(\ell) \notin \mathsf{upd}(r)$. A database $\mathcal{D}$ *satisfies* r, denoted $\mathcal{D} \models r$, if it satisfies τ_r. It satisfies a set of AICs η, denoted $\mathcal{D} \models \eta$, if $\mathcal{D} \models r$ for every $r \in \eta$. A set of AICs is *consistent* if there exists a database $\mathcal{D}$ such that $\mathcal{D} \models \eta$.

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 from $\mathsf{dom}(\mathcal{D})$, (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 $|\mathsf{upd}(r)| = 1$. The *normalization* of an AIC r is the set of AICs $N(r) = \{\mathsf{body}(r) \rightarrow \{A\} \mid A \in \mathsf{upd}(r)\}$. The normalization of a set of AICs η is $N(\eta) = \bigcup_{r \in \eta} N(r)$. Note that $gr_{\mathcal{D}}(N(\eta)) = N(gr_{\mathcal{D}}(\eta))$.

Repair updates A *repair update (r-update)*[2] of a database $\mathcal{D}$ w.r.t. a set of AICs η is a consistent subset-minimal set of update actions $\mathcal{U}$ such that $\mathcal{D} \circ \mathcal{U} \models \eta$. We denote the set of r-updates of $\mathcal{D}$ w.r.t. η by $Up(\mathcal{D}, \eta)$. It is easy to check that $\{\mathcal{D} \circ \mathcal{U} \mid \mathcal{U} \in Up(\mathcal{D}, \eta)\} = SRep(\mathcal{D}, \mathcal{C}_\eta)$ where $\mathcal{C}_\eta$ is the set of universal constraints that correspond to AICs in η.

To take into account the restrictions on the possible update actions expressed by the AICs, several classes of r-updates have been defined. The first one, *founded* r-updates (Caroprese et al. 2006), was criticized for exhibiting circularity of support, leading to the introduction of more restrictive *justified* (Caroprese and Truszczynski 2011), *well-founded* (Cruz-Filipe et al. 2013), and *grounded* r-updates (Cruz-Filipe 2016). The latter were motivated by arguably unexpected behaviors of justified and well-founded r-updates. In particular, justified r-updates are criticized for being too complicated and for excluding some r-updates that seem reasonable. Moreover, they are sensitive to normalization, unlike founded, well-founded and grounded r-updates.

Definition 7. *An r-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 $A \in \mathsf{upd}(r)$ and $\mathcal{D} \circ \mathcal{U} \setminus \{A\} \not\models r$.*
- well-founded *if there exists a sequence of actions $A_1, \ldots, A_n$ such that $\mathcal{U} = \{A_1, \ldots, A_n\}$, and for every $1 \leq i \leq n$, there exists $r_i \in gr_{\mathcal{D}}(\eta)$ such that $A_i \in \mathsf{upd}(r_i)$ and $\mathcal{D} \circ \{A_1, \ldots, A_{i-1}\} \not\models r_i$.*
- grounded *if for every $\mathcal{V} \subsetneq \mathcal{U}$, there exists $r \in gr_{\mathcal{D}}(N(\eta))$ such that $\mathcal{D} \circ \mathcal{V} \not\models r$ and $\mathsf{upd}(r) \subseteq \mathcal{U} \setminus \mathcal{V}$.*
- 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 $ne(\mathcal{D}, \mathcal{D} \circ \mathcal{U})$ where*
  - *$ne(\mathcal{D}, \mathcal{D} \circ \mathcal{U}) = \{+\alpha \mid \alpha \in \mathcal{D} \cap (\mathcal{D} \circ \mathcal{U})\} \cup \{-\alpha \mid \alpha \notin \mathcal{D} \cup (\mathcal{D} \circ \mathcal{U}), \alpha \in Facts^{\mathbf{S}}_{\mathcal{D}}\}$ (set of* no-effect actions*)*
  - *$\mathcal{U}$ is* closed under *η if for every $r \in gr_{\mathcal{D}}(\eta)$, if $\mathcal{U}$ satisfies all the non-updatable literals of r, then $\mathcal{U}$ contains an update action from r.*

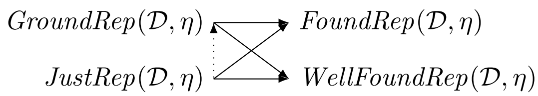
We denote by $FoundUp(\mathcal{D}, \eta)$, $WellFoundUp(\mathcal{D}, \eta)$, $GroundUp(\mathcal{D}, \eta)$ and $JustUp(\mathcal{D}, \eta)$ respectively the sets of founded, well-founded, grounded and justified r-updates of $\mathcal{D}$ w.r.t. η and let $XRep(\mathcal{D}, \eta) = \{\mathcal{D} \circ \mathcal{U} \mid \mathcal{U} \in XUp(\mathcal{D}, \eta)\}$ be the set of corresponding repairs.

Calautti et al. (2021) recently redefined founded r-updates. In fact, we show that their definition coincides with grounded r-updates, yielding the following characterization.

Proposition 6. *For every $\mathcal{U} \in Up(\mathcal{D}, \eta)$, $\mathcal{U}$ is grounded iff $\mathcal{U} \in Up(\mathcal{D}, \eta[\mathcal{U}])$, where $\eta[\mathcal{U}]$ is the set of AICs obtained from $gr_{\mathcal{D}}(\eta)$ by deleting update actions not occurring in $\mathcal{U}$ and AICs whose update actions have all been deleted.*

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 (Caroprese and Truszczynski 2011; Cruz-Filipe et al. 2013; Cruz-Filipe 2016).

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



5.2 From Prioritized Databases to AICs

Given a prioritized database $\mathcal{D}^{\mathcal{C}}_{\succ}$ we define the following set of ground AICs: $\eta^{\mathcal{C}}_{\succ} = \{r_{\mathcal{E}} \mid \mathcal{E} \in \mathit{Conf}(\mathcal{D}, \mathcal{C})\}$ where

$$r_{\mathcal{E}} := \bigwedge_{\lambda \in \mathcal{E}} \lambda \rightarrow \{\mathit{fix}(\lambda) \mid \lambda \in \mathcal{E}, \forall \mu \in \mathcal{E}, \lambda \not\succ \mu\}.$$

Intuitively, $\eta^{\mathcal{C}}_{\succ}$ expresses that conflicts of $\mathcal{D}^{\mathcal{C}}_{\succ}$ should be fixed by modifying the least preferred literals according to $\succ$.

We can prove that Pareto-optimal repairs of $\mathcal{D}^{\mathcal{C}}_{\succ}$ coincide with several kinds of repairs of $\mathcal{D}$ w.r.t. $\eta^{\mathcal{C}}_{\succ}$.

Proposition 7. *For every prioritized database $\mathcal{D}^{\mathcal{C}}_{\succ}$, $PRep(\mathcal{D}^{\mathcal{C}}_{\succ}) = JustRep(\mathcal{D}, \eta^{\mathcal{C}}_{\succ}) = GroundRep(\mathcal{D}, \eta^{\mathcal{C}}_{\succ}) = FoundRep(\mathcal{D}, \eta^{\mathcal{C}}_{\succ}) \subseteq WellFoundRep(\mathcal{D}, \eta^{\mathcal{C}}_{\succ})$.*

This result is interesting not only because it provides additional evidence for the naturalness of Pareto-optimal repairs, but also because it identifies a class of AICs for which justified, grounded, and founded r-updates coincide. The proof in fact shows that these three notions coincide for every set of ground AICs η that is *monotone*, i.e. does not contain both a fact α and the complementary literal $\neg\alpha$.

We remark that the final inclusion in Proposition 7 may be strict. This is demonstrated on the next example, which suggests that well-founded repairs may be too permissive:

Example 5. *It is possible to construct a prioritized database $\mathcal{D}^{\mathcal{C}}_{\succ}$ where $\mathcal{D} = \{\alpha, \beta, \gamma, \delta\}$ and $\eta^{\mathcal{C}}_{\succ} = \{\alpha \wedge \beta \rightarrow \{-\beta\}, \alpha \wedge \gamma \rightarrow \{-\alpha\}, \gamma \wedge \delta \rightarrow \{-\gamma\}\}$. For the AICs $\eta^{\mathcal{C}}_{\succ}$, the r-update $\{-\alpha, -\gamma\}$ is well-founded, but not founded, as the only founded r-update is $\{-\beta, -\gamma\}$. We argue that $\{-\beta, -\gamma\}$ should indeed be preferred to $\{-\alpha, -\gamma\}$, since the first AIC expresses that it is better to remove β than α.*

The reduction used to show Proposition 7 is data-dependent and requires us to create potentially exponentially many ground AICs, one for every conflict. In the case of *denial constraints*, however, we can give an alternative data-independent reduction, provided that the priority relation $\succ$ is specified in the database. We thus assume for the next result that $P_{\succ}$ is a predicate in $\mathbf{S}$, that the first attribute of each relation in $\mathbf{S} \setminus \{P_{\succ}\}$ stores a unique fact identifier, and $P_{\succ}$ stores pairs of such identifiers. Then given a set of denial constraints $\mathcal{C}$ over $\mathbf{S} \setminus \{P_{\succ}\}$, we build a set $min(\mathcal{C})$ that is equivalent to $\mathcal{C}$ but has the property that the conflicts of $\mathcal{D}$ w.r.t. $\mathcal{C}$ are precisely the images of constraint bodies of $min(\mathcal{C})$ on $\mathcal{D}$. This can be achieved by replacing each $\varphi \rightarrow \bot \in \mathcal{C}$ with all refinements obtaining by (dis)equating variables in φ with each other, or with constants mentioned in $\mathcal{C}$, then removing any subsumed constraints. For example, if $\mathcal{C} = \{R(x, x) \rightarrow \bot, R(x, y) \wedge S(y) \rightarrow \bot\}$, then $min(\mathcal{C})$ contains $R(x, x) \rightarrow \bot$ and $R(x, y) \wedge S(y) \wedge x \neq y \rightarrow \bot$, so $\{R(a, a), S(a)\}$ is no longer an image of a constraint body. We then define $\eta^{\mathcal{C}}$ as the set of all AICs

$$\big(\ell_1 \wedge \ldots \wedge \ell_n \wedge \varepsilon \wedge \bigwedge_{\ell_j \neq \ell_i} \neg P_{\succ}(id_i, id_j)\big) \rightarrow \{-\ell_i\}.$$

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

Proposition 8. *For every set of denial constraints $\mathcal{C}$, database $\mathcal{D}$ and priority relation $\succ$ of $\mathcal{D}$ w.r.t. $\mathcal{C}$, $PRep(\mathcal{D}^{\mathcal{C}}_{\succ}) = JustRep(\mathcal{D}, \eta^{\mathcal{C}}) = GroundRep(\mathcal{D}, \eta^{\mathcal{C}}) = FoundRep(\mathcal{D}, \eta^{\mathcal{C}}) \subseteq WellFoundRep(\mathcal{D}, \eta^{\mathcal{C}})$.*

This reduction could be used for example to transfer data complexity lower bounds for prioritized databases with denial constraints to the setting of AICs.

5.3 Towards Well-Behaved AICs

When translating a prioritized database into AICs, we obtained monotone sets of AICs, for which most of the different kinds of r-update coincide. Can we generalize this idea to obtain larger classes of 'well-behaved' sets of AICs which share this desirable behavior? This subsection explores this question and provides some first insights.

We start by defining the following condition, which serves to ensure that all constraints are made explicit:

Definition 8. *We say that a set η of ground AICs is* closed under resolution *if it is consistent, and for every pair of AICs $r_1, r_2 \in \eta$, if there exists $\alpha \in \mathsf{lits}(r_1)$ such that $\neg\alpha \in \mathsf{lits}(r_2)$, and $\mathsf{lits}(r_1) \cup \mathsf{lits}(r_2) \setminus \{\alpha, \neg\alpha\}$ is a consistent set of literals, then there exists $r_3 \in \eta$ with $\mathsf{lits}(r_3) = \mathsf{lits}(r_1) \cup \mathsf{lits}(r_2) \setminus \{\alpha, \neg\alpha\}$. A set of AICs η is closed under resolution if so is $gr_{\mathcal{D}}(\eta)$ for every database $\mathcal{D}$.*

The name 'closure under resolution' comes from considering the clauses that correspond to the negation of the rule bodies: if we have AICs whose clauses are $\neg\alpha \vee \varphi$ and $\alpha \vee \psi$, then we should also have an AIC for their resolvent $\varphi \vee \psi'$, corresponding to the implied constraint $\neg\varphi \wedge \neg\psi \rightarrow \bot$. This property ensures that η captures all potential conflicts: for every $\mathcal{D}$, if $\mathcal{C}_\eta = \{\tau_r \mid r \in \eta\}$, then $\mathit{Conf}(\mathcal{D}, \mathcal{C}_\eta) = \{\mathsf{lits}(r) \mid r \in gr_{\mathcal{D}}(\eta), \mathcal{D} \not\models r$, and there is no $r' \in gr_{\mathcal{D}}(\eta)$ with $\mathsf{lits}(r') \subsetneq \mathsf{lits}(r)\}$.

The following example, given by Bogaerts and Cruz-Filipe (2018) to show that grounded r-updates do not coincide with the intersection of founded and well-founded r-updates, illustrates that sets of AICs not closed under resolution may exhibit undesirable behaviors.

Example 6. *Consider $\mathcal{D} = \emptyset$ and η that contains the AICs:*

$$r_1 : \neg\alpha \wedge \neg\beta \rightarrow \{+\alpha\} \qquad r_4 : \alpha \wedge \beta \wedge \neg\gamma \rightarrow \{+\gamma\}$$
$$r_2 : \alpha \wedge \neg\beta \rightarrow \{+\beta\} \qquad r_5 : \alpha \wedge \neg\beta \wedge \gamma \rightarrow \{+\beta\}$$
$$r_3 : \neg\alpha \wedge \beta \rightarrow \{-\beta\} \qquad r_6 : \neg\alpha \wedge \beta \wedge \gamma \rightarrow \{+\alpha\}$$

$\mathcal{U} = \{+\alpha, +\beta, +\gamma\}$ is founded and well-founded but is not grounded: taking $\mathcal{V} = \{+\beta\}$, we have $\mathcal{V} \subsetneq \mathcal{U}$ but there is no $r \in \eta$ such that $\{\beta\} \not\models r$ and $\mathsf{upd}(r) \cap \{+\alpha, +\gamma\} \neq \emptyset$.

However, it can be verified that $\mathcal{U}$ is in fact the only r-update of $\mathcal{D}$ w.r.t. η. Indeed, the conflicts of $\mathcal{D}$ w.r.t. the constraints expressed by η are $\{\neg\alpha\}$, $\{\neg\beta\}$ and $\{\neg\gamma\}$.

If η were closed under resolution, it would contain $\neg\alpha \rightarrow \{+\alpha\}$, $\neg\beta \rightarrow \{+\beta\}$, and $\neg\gamma \rightarrow \{+\gamma\}$, in which case $\mathcal{U}$ would be grounded, as expected for the unique r-update.

It is always possible to transform a set of ground AICs into one that is closed under resolution by adding the required AICs. However this may result in an exponential blowup. Moreover, we need to choose the update actions of the added AICs. We advocate for this to be done by propagating the relevant update actions of the rules on which the resolution is done. A set of ground AICs obtained in this way will be closed under resolution and will *preserve actions under resolution* according to the following definition.

Definition 9. *We say that a set η of ground AICs* preserves actions under resolution *if for every triple of AICs r_1, r_2, $r_3 \in \eta$, if there exists α such that $\alpha \in \mathsf{lits}(r_1)$, $\neg\alpha \in \mathsf{lits}(r_2)$, and $\mathsf{lits}(r_3) = \mathsf{lits}(r_1) \cup \mathsf{lits}(r_2) \setminus \{\alpha, \neg\alpha\}$, then $\mathsf{upd}(r_1) \cup \mathsf{upd}(r_2) \setminus \{+\alpha, -\alpha\} \subseteq \mathsf{upd}(r_3)$. A set of AICs η preserves actions under resolution if so does $gr_{\mathcal{D}}(\eta)$ for every database $\mathcal{D}$.*

The next example shows that a set of AICs which does not preserve actions under resolution may be ambiguous.

Example 7. *Let $\mathcal{D} = \{\alpha, \beta, \gamma\}$, and η that contains:*

$$r_1 : \alpha \wedge \beta \rightarrow \{-\alpha\} \qquad r_3 : \alpha \wedge \neg\delta \rightarrow \{+\delta\}$$
$$r_2 : \beta \wedge \gamma \rightarrow \{-\gamma\} \qquad r_4 : \beta \wedge \delta \rightarrow \{-\beta\}$$

This set of AICs is closed under resolution but does not preserve actions under resolution: due to r_3 and r_4, $-\beta$ should be an update action of r_1. Indeed, r_3 and r_4 together indicate that if α and β are present, β should be removed (since if δ is absent, it should be added, due to r_3, and β should be removed when δ is present, by r_4).

To make η preserve actions under resolution, there are three possibilities: (1) change r_1 to $\alpha \wedge \beta \rightarrow \{-\beta\}$ (if α is preferred to β), or (2) change r_4 to $\beta \wedge \delta \rightarrow \{-\delta\}$ (if β is preferred to α), or (3) change r_1 to $\alpha \wedge \beta \rightarrow \{-\alpha, -\beta\}$ (if neither α nor β is preferred to the other).

Sets of AICs that are closed under resolution and preserve actions under resolution are well behaved in the sense that they make most of the r-update notions coincide. The monotone sets of AICs mentioned in relation to Proposition 7 trivially satisfy these two conditions.

Proposition 9. *If η is closed under resolution and preserves actions under resolution, then for every database $\mathcal{D}$, $JustRep(\mathcal{D}, \eta) = GroundRep(\mathcal{D}, \eta) = FoundRep(\mathcal{D}, \eta) \subseteq WellFoundRep(\mathcal{D}, \eta)$.*

The next example shows that both conditions are necessary for obtaining Proposition 9.

Example 8. *Consider $\mathcal{D} = \{\alpha, \beta, \gamma\}$ and the two sets*

$$\eta_1 = \{\alpha \wedge \neg\beta \rightarrow \{-\alpha\}, \quad \neg\alpha \wedge \beta \rightarrow \{-\beta\}, \quad \alpha \wedge \beta \wedge \gamma \rightarrow \{-\gamma\}\}$$
$$\eta_2 = \eta_1 \cup \{\alpha \wedge \gamma \rightarrow \{-\gamma\}, \quad \beta \wedge \gamma \rightarrow \{-\gamma\}\}.$$

η_1 is not closed under resolution but (trivially) preserves actions under resolution, while η_2 is closed under resolution but does not preserve actions under resolution.

In both cases, there are two founded r-updates: $\{-\gamma\}$ and $\{-\alpha, -\beta\}$. However, $\{-\alpha, -\beta\}$ is not well-founded, hence not grounded nor justified. Indeed, $\mathcal{D}$ violates only AICs whose only update action is $-\gamma$.

Even if a set of AICs is such that justified, grounded and founded repairs are guaranteed to exist and coincide, its behavior may still be puzzling, as illustrated next.

Example 9. *Let $\mathcal{D} = \{\alpha, \beta, \gamma, \delta\}$ and η be the monotone set of AICs comprising the following AICs:*

$$r_1 : \alpha \wedge \delta \rightarrow \{-\delta\} \qquad r_3 : \alpha \wedge \beta \wedge \gamma \wedge \delta \rightarrow \{-\beta\}$$
$$r_2 : \alpha \wedge \beta \wedge \delta \rightarrow \{-\alpha\} \qquad r_4 : \beta \wedge \gamma \rightarrow \{-\gamma\}$$

There are four r-updates:

$\mathcal{U}_1 = \{-\alpha, -\gamma\}$ and $\mathcal{U}_2 = \{-\delta, -\gamma\}$ are founded
$\mathcal{U}_3 = \{-\delta, -\beta\}$ is not founded but is well-founded
$\mathcal{U}_4 = \{-\alpha, -\beta\}$ is not founded nor well-founded

There are two conflicts: $\{\alpha, \delta\}$ and $\{\beta, \gamma\}$. It is natural to prefer removing γ rather than β to resolve the latter conflict (due to r_4), which would justify to preferring $\mathcal{U}_1$ and $\mathcal{U}_2$ over $\mathcal{U}_4$ and $\mathcal{U}_3$ respectively. However, the exact same argument applied to r_1 should lead us to prefer removing δ to solve the first conflict, thus to prefer $\mathcal{U}_2$ over $\mathcal{U}_1$. It is therefore not clear why both $\mathcal{U}_1$ and $\mathcal{U}_2$ should be the preferred r-updates. The intention of a user specifying the preceding AICs is probably quite far from their actual behavior.

We thus believe that a reasonable property for sets of AICs is to respect the principle that adding atoms to a rule body can only restrict the possible update actions. We call the *anti-normalization* of a set η of AICs the set $AN(\eta)$ of AICs that replace all the AICs $r_1, \ldots, r_n \in \eta$ that share the same body by a single AIC whose update actions are the union of the update actions of $r_1, \ldots, r_n$.

Definition 10. *We say that a set η of ground AICs* preserves actions under strengthening *if for every pair of AICs r_1, r_2 in $AN(\eta)$, if $\mathsf{lits}(r_1) \subseteq \mathsf{lits}(r_2)$, then $\mathsf{upd}(r_2) \subseteq \mathsf{upd}(r_1)$. A set of AICs η preserves actions under strengthening if so does $gr_{\mathcal{D}}(\eta)$ for every database $\mathcal{D}$.*

The following proposition shows that if η preserves actions under strengthening, then constraints that have non-minimal bodies have no influence on the r-updates.

Proposition 10. *Let η be a set of ground AICs and $min(\eta)$ be the set of AICs from $AN(\eta)$ that have (subset-)minimal bodies. If η preserves actions under strengthening, then for every $\mathcal{D}$, for $X \in \{Found, WellFound, Ground\}$ $XUp(\mathcal{D}, \eta) = XUp(\mathcal{D}, AN(\eta)) = XUp(\mathcal{D}, min(\eta))$, and $JustUp(\mathcal{D}, AN(\eta)) = JustUp(\mathcal{D}, min(\eta))$.*

5.4 From AICs to Prioritized Databases

We next study the possibility of reducing well-behaved sets of AICs to prioritized databases and discuss the differences between the two settings.

Binary conflicts case We first consider the case where the size of the conflicts is at most two (this covers, for example, AIC bodies corresponding to functional dependencies or class disjointness). In this case, given a set η of AICs *closed under resolution that preserves actions under resolution and under strengthening* and a database $\mathcal{D}$, we build a set of constraints $\mathcal{C}_\eta$ and a binary relation $\succ_\eta$ such that if $\succ_\eta$ is acyclic, the Pareto-optimal repairs of $\mathcal{D}_{\succ_\eta}^{\mathcal{C}_\eta}$ coincide with the founded, grounded and justified repairs of $\mathcal{D}$ w.r.t. η. We take $\mathcal{C}_\eta = \{\tau_r \mid r \in \eta\}$ and define $\succ_\eta$ so that $\lambda \succ_\eta \mu$ iff

- there exists $r \in min_g(\eta)$ such that $\mathcal{D} \not\models r$, $\{\lambda, \mu\} \subseteq \mathsf{lits}(r)$, and $\mathit{fix}(\mu) \in \mathsf{upd}(r)$; and
- for every $r \in min_g(\eta)$ such that $\mathcal{D} \not\models r$ and $\{\lambda, \mu\} \subseteq \mathsf{lits}(r)$, $\mathit{fix}(\lambda) \notin \mathsf{upd}(r)$,

where $min_g(\eta) = \{r \in gr_{\mathcal{D}}(\eta) \mid$ there is no $r' \in gr_{\mathcal{D}}(\eta)$ with $\mathsf{lits}(r') \subsetneq \mathsf{lits}(r)\}$. As η is closed under resolution, $Conf(\mathcal{D}, \mathcal{C}_\eta) = \{\mathsf{lits}(r) \mid r \in min_g(\eta), \mathcal{D} \not\models r\}$.

Proposition 11. *If η is closed under resolution, preserves actions under resolution and under strengthening, the size of the conflicts of $\mathcal{D}$ w.r.t. η is bounded by 2, and $\succ_\eta$ is acyclic, then $PRep(\mathcal{D}_{\succ_\eta}^{\mathcal{C}_\eta}) = JustRep(\mathcal{D}, \eta) = GroundRep(\mathcal{D}, \eta) = FoundRep(\mathcal{D}, \eta) \subseteq WellFoundRep(\mathcal{D}, \eta)$.*

The following examples show that the three first conditions are necessary.

Example 10. *Let $\mathcal{D} = \{\alpha, \beta, \gamma\}$ and $\eta = \{\alpha \wedge \beta \rightarrow \{-\beta\}, \neg\beta \wedge \gamma \rightarrow \{-\gamma\}\}$, which preserves actions under resolution and strengthening but is not closed under resolution. We have $Conf(\mathcal{D}, \mathcal{C}_\eta) = \{\{\alpha, \beta\}, \{\alpha, \gamma\}\}$ and $\alpha \succ_\eta \beta$. Both $\{\alpha\}$ and $\{\beta, \gamma\}$ are Pareto-optimal, but the only founded r-update (which is also grounded and justified) is $\{-\beta, -\gamma\}$.*

Example 11 (Example 7 cont'd)**.** *In Example 7, η is closed under resolution and preserves actions under strengthening but not under resolution. We have $Conf(\mathcal{D}, \mathcal{C}_\eta) = \{\{\alpha, \beta\}, \{\beta, \gamma\}, \{\alpha, \neg\delta\}\}$ and $\beta \succ_\eta \alpha$, $\beta \succ_\eta \gamma$, $\alpha \succ_\eta \neg\delta$. The only Pareto-optimal repair is $\{\beta\}$, but $\{-\beta, +\delta\}$ is a founded, grounded and justified r-update.*

Example 12 (Example 9 cont'd)**.** *In Example 9, η is closed under resolution and preserves actions under resolution but not under strengthening. We have $Conf(\mathcal{D}, \mathcal{C}_\eta) = \{\{\alpha, \delta\}, \{\beta, \gamma\}\}$ and $\alpha \succ_\eta \delta$, $\beta \succ_\eta \gamma$. The only Pareto-optimal repair is $\{\alpha, \beta\}$, but $\{-\alpha, -\gamma\}$ is a founded, grounded and justified r-update.*

Note that $\succ_\eta$ may be cyclic: if $\eta = \{A(x) \wedge B(x) \rightarrow \{-A(x)\}, B(x) \wedge C(x) \rightarrow \{-B(x)\}, C(x) \wedge A(x) \rightarrow \{-C(x)\}\}$ and $\mathcal{D} = \{A(a), B(a), C(a)\}$, we obtain $A(a) \succ_\eta C(a) \succ_\eta B(a) \succ_\eta A(a)$.

General case Let us now consider the case where the size of the conflicts is not bounded. If we apply the same reduction, we can only show the following inclusions between repairs of $\mathcal{D}$ w.r.t. η and Pareto-optimal repairs of $\mathcal{D}_{\succ_\eta}^{\mathcal{C}_\eta}$:

Proposition 12. *If η is closed under resolution, preserves actions under resolution and under strengthening, and $\succ_\eta$ is acyclic, then $JustRep(\mathcal{D}, \eta) = GroundRep(\mathcal{D}, \eta) = FoundRep(\mathcal{D}, \eta) \subseteq PRep(\mathcal{D}_{\succ_\eta}^{\mathcal{C}_\eta})$.*

The next example shows that the inclusion may be strict.

Example 13. *Let $\mathcal{D} = \{\alpha, \beta, \gamma, \delta, \epsilon\}$ and η consist of:*

$$r_1 : \alpha \wedge \beta \wedge \gamma \rightarrow \{-\beta\} \qquad r_3 : \delta \wedge \epsilon \rightarrow \{-\delta\}$$
$$r_2 : \alpha \wedge \beta \wedge \delta \rightarrow \{-\alpha, -\beta\}$$

We obtain $Conf(\mathcal{D}, \mathcal{C}_\eta) = \{\{\alpha, \beta, \gamma\}, \{\alpha, \beta, \delta\}, \{\delta, \epsilon\}\}$ and $\gamma \succ_\eta \beta$, $\delta \succ_\eta \alpha$, $\delta \succ_\eta \beta$ and $\epsilon \succ_\eta \delta$ (note that $\alpha \not\succ_\eta \beta$ because $-\alpha$ is an update action of r_2). The repair $\{\beta, \gamma, \epsilon\}$ is Pareto-optimal, but the corresponding r-update $\{-\alpha, -\delta\}$ is not founded, as $-\alpha$ appears only in r_2 and $\mathcal{D} \circ \{-\delta\} \models r_2$.

One might try to modify the definition of $\succ_\eta$ by dropping the second condition and adding $\mathit{fix}(\lambda) \notin \mathsf{upd}(r)$ to the first. In this case, $\{\beta, \gamma, \epsilon\}$ is no longer Pareto-optimal. However, now if we take $\eta' = \eta \setminus \{r_3\}$, then $\{-\alpha\}$ would be founded, but the corresponding repair $\{\beta, \gamma, \delta, \epsilon\}$ would not be Pareto-optimal, violating the inclusion of Proposition 12.

This example shows that even for AICs corresponding to denial constraints, there is no clear way to define a priority relation that captures the preferences expressed by the AICs.

6 Conclusion and Future Work

We studied how to incorporate preferences into repair-based query answering for an expressive setting in which databases are equipped with universal constraints, and both fact additions and deletions are used to restore consistency. We showed that the existing framework of prioritized databases could be faithfully adapted to this richer setting, although the proofs are more involved and crucially rely upon finding the right definition of what constitutes a conflict. While these results focus on databases, we expect that they will also prove useful for exploring symmetric difference repairs in related KR settings, e.g. ontologies with closed predicates.

Our complexity analysis showed that adopting optimal repairs in place of symmetric difference repairs does not increase the complexity of repair-based query answering. A major difference between denial and universal constraints is that the latter may lead to conflicts of unbounded size. We showed that it is intractable to recognize a conflict and that several problems drop in complexity if we assume that the conflicts are available. This suggests the interest of developing structural conditions on constraint sets that ensure easy-to-compute conflicts, as well as practical algorithms for computing and updating the set of conflicts, which could enable an integration with existing SAT-based approaches.

Intrigued by the high-level similarities between prioritized databases and active integrity constraints, we explored how the two formalisms relate. We exhibited a natural translation of prioritized databases into AICs whereby Pareto-optimal repairs coincide with founded, grounded and justified repairs w.r.t. the generated set of AICs. We take this as further evidence that Pareto-optimal repairs are an especially natural notion (we previously showed that Pareto-optimal (subset) repairs correspond to stable extensions in argumentation (Bienvenu and Bourgaux 2020)). It would be of interest to extend our comparison to other more recent notions of repair updates for AICs (Feuillade, Herzig, and Rantsoudis 2019, Bogaerts and Cruz-Filipe 2018; 2021).

Our work also provided new insights into AICs. Existing examples used to distinguish different notions of r-update often seem unnatural in some respect. This led us to devise a set of criteria for 'well-behaved' AICs, which provide sufficient conditions for founded, grounded and justified repairs to coincide (Example 5 suggests that well-founded repairs are too permissive). Even for such restricted AICs, it is not always clear what user intentions are being captured. We thus believe that there is still work to be done to develop user-friendly formalisms for expressing constraints and preferences on how to handle constraint violations.

Acknowledgements

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

References

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

Arming, S.; Pichler, R.; and Sallinger, E. 2016. Complexity of repair checking and consistent query answering. In *Proceedings of the 19th International Conference on Database Theory (ICDT)*.

Bertossi, L. E. 2019. 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)*, 48–58.

Bienvenu, M., and Bourgaux, C. 2020. 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)*.

Bienvenu, M., and Bourgaux, C. 2022. 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)*.

Bienvenu, M., and Bourgaux, C. 2023. Inconsistency handling in prioritized databases with universal constraints: Complexity analysis and links with active integrity constraints. arxiv.org/abs/2306.03523 [cs.DB].

Bienvenu, M., and Rosati, R. 2013. Tractable approximations of consistent query answering for robust ontology-based data access. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*.

Bienvenu, M.; Bourgaux, C.; and Goasdoué, F. 2014. Querying inconsistent description logic knowledge bases under preferred repair semantics. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*.

Bienvenu, M. 2020. A short survey on inconsistency handling in ontology-mediated query answering. *Künstliche Intelligenz* 34(4):443–451.

Bogaerts, B., and Cruz-Filipe, L. 2018. Fixpoint semantics for active integrity constraints. *Artif. Intell.* 255:43–70.

Bogaerts, B., and Cruz-Filipe, L. 2021. Stratification in approximation fixpoint theory and its application to active integrity constraints. *ACM Trans. Comput. Log.* 22(1):6:1–6:19.

Calautti, M.; Caroprese, L.; Greco, S.; Molinaro, C.; Trubitsyna, I.; and Zumpano, E. 2021. Existential active integrity constraints. *Expert Syst. Appl.* 168:114297.

Caroprese, L., and Truszczynski, M. 2011. Active integrity constraints and revision programming. *Theory Pract. Log. Program.* 11(6):905–952.

Caroprese, L.; Greco, S.; Sirangelo, C.; and Zumpano, E. 2006. Declarative semantics of production rules for integrity maintenance. In *Logic Programming, 22nd International Conference, ICLP 2006, Seattle, WA, USA, August 17-20, 2006, Proceedings*.

Caroprese, L.; Greco, S.; and Zumpano, E. 2009. Active integrity constraints for database consistency maintenance. *IEEE Trans. Knowl. Data Eng.* 21(7):1042–1058.

Cruz-Filipe, L.; Gaspar, G.; Engrácia, P.; and Nunes, I. 2013. Computing repairs from active integrity constraints. In *Seventh International Symposium on Theoretical Aspects of Software Engineering, TASE 2013, 1-3 July 2013, Birmingham, UK*.

Cruz-Filipe, L. 2016. Grounded fixpoints and active integrity constraints. In *Technical Communications of the 32nd International Conference on Logic Programming, ICLP 2016 TCs, October 16-21, 2016, New York City, USA*.

Eiter, T.; Fink, M.; Greco, G.; and Lembo, D. 2008. Repair localization for query answering from inconsistent databases. *ACM Trans. Database Syst.* 33(2):10:1–10:51.

Fagin, R.; Kimelfeld, B.; and Kolaitis, P. G. 2015. Dichotomies in the complexity of preferred repairs. In *Proceedings of the 34th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS)*.

Feuillade, G.; Herzig, A.; and Rantsoudis, C. 2019. A dynamic logic account of active integrity constraints. *Fundam. Informaticae* 169(3):179–210.

Flesca, S.; Greco, S.; and Zumpano, E. 2004. Active integrity constraints. In *Proceedings of the 6th International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming, 24-26 August 2004, Verona, Italy*.

Kimelfeld, B.; Livshits, E.; and Peterfreund, L. 2017. Detecting ambiguity in prioritized database repairing. In *Proceedings of the 20th International Conference on Database Theory (ICDT)*.

Lembo, D.; Lenzerini, M.; Rosati, R.; Ruzzi, M.; and Savo, D. F. 2010. Inconsistency-tolerant semantics for description logics. In *Proceedings of the 4th International Conference on Web Reasoning and Rule Systems (RR)*.

Livshits, E., and Kimelfeld, B. 2017. Counting and enumerating (preferred) database repairs. In *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS)*.

Marileo, M. C., and Bertossi, L. E. 2010. The consistency extractor system: Answer set programs for consistent query answering in databases. *Data Knowl. Eng.* 69(6):545–572.

Staworko, S., and Chomicki, J. 2010. Consistent query answers in the presence of universal constraints. *Inf. Syst.* 35(1):1–22.

Staworko, S.; Chomicki, J.; and Marcinkowski, J. 2012. Prioritized repairing and consistent query answering in relational databases. *Annals of Mathematics and Artificial Intelligence (AMAI)* 64(2-3):209–246.