Using ASP(Q) to Handle Inconsistent Prioritized Data (Extended Abstract)

Meghyn Bienvenu[1], Camille Bourgaux[2], Robin Jean[1] and Giuseppe Mazzotta[3]

[1] Université de Bordeaux, CNRS, Bordeaux INP, LaBRI, Talence, France
[2] DI ENS, ENS, CNRS, PSL University & Inria, Paris, France
[3] University of Calabria, Rende, Italy

Abstract

This extended abstract summarizes our KR'26 paper [1], where we explore the use of answer set programming (ASP) and its extension with quantifiers, ASP(Q), for inconsistency-tolerant querying of prioritized data. We consider the variants of three well-known semantics (AR, brave and IAR) that use three notions of optimal repairs (Pareto-, globally- and completion-optimal), and for which query answering is in the first or second level of the polynomial hierarchy for a large class of logical theories. Notably, this paper presents the first implementation of globally-optimal repair-based semantics, as well as the first implementation of the grounded semantics, which is a tractable under-approximation of all these optimal repair-based semantics. Our experimental evaluation sheds light on the feasibility of computing answers under globally-optimal repair semantics and the impact of adopting different semantics, approximations, and encodings.

Keywords

Inconsistency-tolerant query answering, Optimal repairs, Prioritized data, Answer set programming, ASP(Q)

1. Introduction

Repair-based semantics are a prominent means of obtaining meaningful answers to queries posed over some data which is inconsistent w.r.t. some logical theory, both in the relational database and ontology-mediated query answering setting (cf. [2, 3] for brief overviews). In this context, a repair is a subset-maximal subset of the data consistent with the logical theory. The most well-known repair-based semantics, called AR in the KR community, requires that the query holds in every repair, while the less cautious brave semantics requires that it holds in some repair, and the more cautious IAR semantics that it holds in the intersection of all repairs [4, 5, 6]. Several notions of preferred repairs have been proposed to take into account some preference information and consider only a subset of all the possible repairs to evaluate the queries (cf. [7] for a survey). In particular, Staworko, Chomicki, and Marcinkowski [8] introduced three kinds of optimal repairs based on a priority relation between conflicting facts, which have attracted a lot of interest in the last decade with extensive complexity analyses [9, 10, 11, 12], two implementations [13, 14], and a framework for specifying and computing priority relations [14].

However, only two of these three kinds of repairs have actually been implemented in the existing systems, namely the Pareto- and completion-optimal repairs. This is due to the higher complexity of reasoning with the third kind of repairs, called globally-optimal. Indeed, for a large class of database constraint and ontology languages, the data complexity of query answering under the variants of AR, IAR and brave that use Pareto- or completion-optimal repairs is in the first level of the polynomial hierarchy, while it is in the second level for the globally-optimal repair-based variants.

We address this gap by developing ASP(Q)-based encodings for all three types of optimal repair semantics, including the first implementation of globally-optimal repairs. Indeed, answer set programming (ASP) with disjunctive rules can be used to model problems up to the second level of the polynomial hierarchy, meaning that semantics based on globally-optimal repairs could theoretically be implemented in ASP. However, this requires advanced techniques, such as saturation [15], which

DL 2026: 39th International Workshop on Description Logics, July 17–19, 2026, Lisbon, Portugal
meghyn.bienvenu@cnrs.fr (M. Bienvenu); camille.bourgaux@ens.fr (C. Bourgaux); robin.jean@u-bordeaux.fr (R. Jean); giuseppe.mazzotta@unical.it (G. Mazzotta)

complicate modeling *"beyond the capabilities of the common ASP laymen"* [16]. Moreover, efficiency-wise it has been shown that in many cases of practical interest alternative formalisms are preferred to disjunctive ASP [17]. Among alternative formalisms, answer set programming with quantifiers (ASP(Q)) uses quantifiers over answer sets of ASP programs and provides a natural and compact way of modeling problems in the entire polynomial hierarchy [18]. Compared to the previous ASP-based implementation [14], we also improve the system performance by using tractable approximations in a similar way as [13]. Moreover, we explore the use of another tractable approximation of the optimal repair-based semantics, called the grounded semantics [11], which has never been implemented so far. Our experimental evaluation shows that using globally-optimal repairs is considerably more challenging than for Pareto- or completion-optimal repairs, while at the same time demonstrating the utility of different optimizations, in particular, the surprising effectiveness of the grounded semantics.

2. Background

Prioritized KBs and optimal repairs. A knowledge base (KB) $\mathcal{K} = (\mathcal{D}, \mathcal{T})$ consists of a dataset $\mathcal{D}$ and a logical theory $\mathcal{T}$ (an ontology or a set of constraints). A *conflict* of $\mathcal{K}$ is a minimal subset $\mathcal{C} \subseteq \mathcal{D}$ such that $(\mathcal{C}, \mathcal{T})$ is inconsistent, and a *repair* is a maximal subset $\mathcal{R} \subseteq \mathcal{D}$ such that $(\mathcal{R}, \mathcal{T})$ is consistent. A *priority relation* $\succ$ for $\mathcal{K}$ is an acyclic binary relation over $\mathcal{D}$ such that $\alpha \succ \beta$ implies $\{\alpha, \beta\}$ is included in some conflict. It is *total* if for every pair $\alpha \neq \beta$ such that $\{\alpha, \beta\}$ is included in a conflict, either $\alpha \succ \beta$ or $\beta \succ \alpha$. A *completion* of $\succ$ is a total priority relation $\succ' \supseteq \succ$. A *prioritized KB* $\mathcal{K}_\succ$ is a KB $\mathcal{K}$ with a priority relation $\succ$ for $\mathcal{K}$. A repair $\mathcal{R}$ admits a *Pareto improvement* $\mathcal{R}'$ if $(\mathcal{R}', \mathcal{T})$ is consistent and there exists $\beta \in \mathcal{R}' \setminus \mathcal{R}$ such that $\beta \succ \alpha$ for every $\alpha \in \mathcal{R} \setminus \mathcal{R}'$. It admits a *global improvement* $\mathcal{R}'$ if $(\mathcal{R}', \mathcal{T})$ is consistent, $\mathcal{R}' \neq \mathcal{R}$, and for every $\alpha \in \mathcal{R} \setminus \mathcal{R}'$, there exists $\beta \in \mathcal{R}' \setminus \mathcal{R}$ with $\beta \succ \alpha$. A repair $\mathcal{R}$ is *Pareto-optimal* (resp. *globally-optimal*) if it does not admit a Pareto (resp. global) improvement. It is *completion-optimal* if it is a globally-optimal repair of $\mathcal{K}_{\succ'}$, for some completion $\succ'$ of $\succ$. We denote by $\mathit{SRep}(\mathcal{K})$, $\mathit{PRep}(\mathcal{K}_\succ)$, $\mathit{GRep}(\mathcal{K}_\succ)$ and $\mathit{CRep}(\mathcal{K}_\succ)$ the sets of (standard) repairs and Pareto-, globally- and completion-optimal repairs, respectively. It is known that $\mathit{CRep}(\mathcal{K}_\succ) \subseteq \mathit{GRep}(\mathcal{K}_\succ) \subseteq \mathit{PRep}(\mathcal{K}_\succ) \subseteq \mathit{SRep}(\mathcal{K})$.

X-brave, X-AR and X-IAR semantics. Fix $X \in \{S, P, G, C\}$ and consider a prioritized KB $\mathcal{K}_\succ$ with $\mathcal{K} = (\mathcal{D}, \mathcal{T})$, query $q(\vec{x})$, and tuple of constants $\vec{a}$. Then $\vec{a}$ is an answer to $q(\vec{x})$ over $\mathcal{K}_\succ$ under *X-brave semantics* if $(\mathcal{R}, \mathcal{T}) \models q(\vec{a})$ for some $\mathcal{R} \in \mathit{XRep}(\mathcal{K}_\succ)$; under *X-AR semantics*, if $(\mathcal{R}, \mathcal{T}) \models q(\vec{a})$ for every $\mathcal{R} \in \mathit{XRep}(\mathcal{K}_\succ)$; under *X-IAR semantics*, if $(\mathcal{B}, \mathcal{T}) \models q(\vec{a})$ where $\mathcal{B} = \bigcap_{\mathcal{R} \in \mathit{XRep}(\mathcal{K}_\succ)} \mathcal{R}$. In FOL fragments for which KB consistency checking and query entailment are in PTime w.r.t. data complexity (e.g., DL-Lite or denial constraints), the data complexity of query entailment under these semantics is in NP (X-brave) and coNP (X-AR/X-IAR) for $X \in \{P, C\}$, but in Σ_2^p (G-brave) and Π_2^p (G-AR/G-IAR) when globally-optimal repairs are used [8, 11].

Grounded semantics. The grounded semantics for prioritized KBs comes from the area of abstract argumentation. For prioritized KB $\mathcal{K}_\succ$ with $\mathcal{K} = (\mathcal{D}, \mathcal{T})$, the *attack relation* $\rightsquigarrow \subseteq (2^{\mathcal{D}} \setminus \{\emptyset\}) \times \mathcal{D}$ is defined by: $\mathcal{B} \rightsquigarrow \alpha$ iff $\mathcal{B} \cup \{\alpha\}$ is a conflict and for every $\beta \in \mathcal{B}$, $\alpha \not\succ \beta$. The characteristic function $\Gamma : 2^{\mathcal{D}} \mapsto 2^{\mathcal{D}}$ is defined by $\Gamma(\mathcal{B}) = \{\alpha \mid \mathcal{E} \rightsquigarrow \alpha \Rightarrow \exists \mathcal{F} \subseteq \mathcal{B}, \beta \in \mathcal{E} \text{ s.t. } \mathcal{F} \rightsquigarrow \beta\}$. The *grounded repair* of $\mathcal{K}_\succ$ is the inclusion-minimal $\mathcal{G} \subseteq \mathcal{D}$ such that $(\mathcal{G}, \mathcal{T})$ is consistent and $\mathcal{G} = \Gamma(\mathcal{G})$, or equivalently, the least fixpoint of Γ. A tuple $\vec{a}$ is an answer to $q(\vec{x})$ over $\mathcal{K}_\succ$ under *grounded semantics* if $(\mathcal{G}, \mathcal{T}) \models q(\vec{a})$. It is known that answers that hold under the grounded semantics hold under X-IAR for $X \in \{P, G, C\}$ and that in FOL fragments for which the size of conflicts is bounded independently from the data, and for which KB consistency and query entailment are in PTime w.r.t. data complexity, grounded query entailment is in PTime w.r.t. data complexity [11].

ASP(Q). We assume basic familiarity with answer set programming (ASP), cf. [19, 20, 21]. An ASP(Q) program has the form $\Box_1 P_1 \cdots \Box_n P_n : C$, where C is a stratified ASP program with constraints, each P_i

Table 1
Main ASP building blocks.

Program	Role
Π_{Reach}	Computes the set $\mathcal{F}$ of relevant (reachable) facts (depends on kind of repair considered)
Π_{SubRep}	Guesses a subset $\mathcal{R} \subseteq \mathcal{F}$ that does not contain any conflict
Π_{Rep}	Guesses a maximal subset $\mathcal{R} \subseteq \mathcal{F}$ that does not contain any conflict
$\Pi_{SomeCause}$	Enforces that $\mathcal{R}$ contains some cause
$\Pi_{NoCause}$	Enforces that $\mathcal{R}$ does not contain any cause
Π_{GImp}	Guesses a global improvement of $\mathcal{R}$
Π_{POpt}	Enforces that $\mathcal{R}$ is a Pareto-optimal repair
Π_{COpt}	Enforces that $\mathcal{R}$ is a completion-optimal repair

is an ASP program, and each $\Box_i \in \{\exists^{st}, \forall^{st}\}$ is a quantifier, which intuitively quantifies over the answer sets of P_i. ASP(Q) captures the full polynomial hierarchy [18]. In particular, programs of the form $\exists^{st} P_1 \, \forall^{st} P_2 : C$ and $\forall^{st} P_1 \, \exists^{st} P_2 : C$ allow one to model Σ_2^p- and Π_2^p-complete problems, respectively. In a nutshell, $\exists^{st} P_1 \, \forall^{st} P_2 : C$ is coherent if there exists an answer set of P_1 such that if one fixes the values of the literals relevant to P_1, every answer set of P_2 that satisfies these values is coherent with C.

3. Encoding Semantics in ASP(Q)

Using ASP and ASP(Q), we compute the answers that hold under the considered optimal repair-based semantics from their causes (i.e., minimal subsets $\mathcal{C} \subseteq \mathcal{D}$ such that $(\mathcal{C}, \mathcal{T})$ is consistent and $(\mathcal{C}, \mathcal{T}) \models q(\vec{a})$), the conflicts and the priority relation, following the approach of [14]. All the encodings are built from some building block programs whose goals are summarized in Table 1.

Localization. To avoid considering the whole dataset in the encodings, we *localize* them to relevant facts (program Π_{Reach}), defined from the query causes using some notions of *reachability* w.r.t. the conflicts and priority relation. We define two such notions of reachability. The first one was already used to localize SAT or ASP encodings [13, 14], and we use the second one for globally-optimal repairs.

G-brave and G-AR. We show that $q(\vec{a})$ holds under G-brave semantics iff Π^G_{brave} is coherent, and under G-AR semantics iff Π^G_{AR} is incoherent, where Π^G_{brave} and Π^G_{AR} are the following ASP(Q) programs.

$$\Pi^G_{brave} = \exists^{st}(\Pi_{Reach} \cup \Pi_{Rep} \cup \Pi_{SomeCause}) \, \forall^{st}\Pi_{GImp} : C$$

$$\Pi^G_{AR} = \exists^{st}(\Pi_{Reach} \cup \Pi_{Rep} \cup \Pi_{NoCause}) \, \forall^{st}\Pi_{GImp} : C$$

with $C = \{$`:- not fail`$\}$, where `fail` does not appear in any other rule. Intuitively, an answer set of $\Pi_{Reach} \cup \Pi_{Rep} \cup \Pi_{SomeCause}$ corresponds to some $\mathcal{R} \in SRep(\mathcal{K})$ with $(\mathcal{R}, \mathcal{T}) \models q(\vec{a})$. The second program of Π^G_{brave}, Π_{GImp}, is then used to search for a global improvement of $\mathcal{R}$. Since C contains only the constraint `:- not fail` and `fail` does not appear in the head of any rule, C is always incoherent thus Π^G_{brave} is coherent iff the universal quantification is trivially satisfied, i.e., $\mathcal{R}$ has no global improvement. The program for G-AR, Π^G_{AR}, is the same as Π^G_{brave} except that $\Pi_{SomeCause}$ is replaced by $\Pi_{NoCause}$, which ensures that the repair $\mathcal{R}$ built by $\Pi_{Reach} \cup \Pi_{Rep}$ does not contain any cause for $q(\vec{a})$.

P- and C- brave and AR. For $X \in \{P, C\}$, we use the following plain ASP programs, which verify whether there exists an optimal repair that either contains some cause or does not contain any cause.

$$\Pi^X_{brave} = \Pi_{Reach} \cup \Pi_{SubRep} \cup \Pi_{XOpt} \cup \Pi_{SomeCause}$$
$$\Pi^X_{AR} = \Pi_{Reach} \cup \Pi_{SubRep} \cup \Pi_{XOpt} \cup \Pi_{NoCause}$$

X-IAR. We compute $\bigcap_{\mathcal{R} \in XRep(\mathcal{K}_\succ)} \mathcal{R}$ by checking for each fact whether it holds under X-AR (since a fact holds under X-AR iff it is in every optimal repair). It is then possible to use this set to evaluate the queries under the X-IAR semantics.

Simplified encoding for binary conflicts. When all conflicts have size 2, we can simplify the programs that compute the reachable facts, the repair, and enforce Pareto-optimality.

Tractable under-approximations of P-IAR. We use a preprocessing step that identifies a subset of the answers that hold under P-IAR (hence under all the considered semantics). We consider two tractable such under-approximations: the *grounded semantics* and the *trivially P-IAR answers* [22, 13], which have some cause such that none of its fact is attacked (w.r.t. $\rightsquigarrow$, i.e., is in $\Gamma(\emptyset)$). In practice, we compute the grounded repair $\mathcal{G}$ or the set of trivially P-IAR facts $\Gamma(\emptyset)$, then use it to filter the answers that have some cause included in it.

4. Experimental Evaluation

Setup. Following [14], we translated into ASP programs a subset of the ORBITS benchmark [13] which provides several conflict sets, priority relations, and potential answers associated with their causes built from the CQAPri benchmark [23], a synthetic benchmark adapted from $\text{LUBM}^{\exists}_{20}$ [24] to evaluate inconsistency-tolerant query answering over DL-Lite KBs. To experiment also with non-binary conflicts, [14] added a constraint that yields conflicts of size 10 and built some priority relations for this case using preference rules. As ASP(Q) solver we used CASPER [25], and as ASP solver, CLINGO [26].

Tractable approximations. One of our main findings is the surprising effectiveness of the grounded semantics as an approximation of the optimal repair-based semantics. Indeed, a large proportion of the potential answers turned out to hold under grounded semantics, and we were able to compute the grounded repair within reasonable time (except for some larger datasets for which we encountered memory issues). Moreover, even in the cases where we were not able to compute the grounded repair, we were able to compute the trivially P-IAR facts very efficiently, and over all the cases we considered, the proportion of facts that belong to the grounded repair but which are not trivially P-IAR varies from 1%-2% on the datasets with a low proportion of facts in conflicts to 15%-31% (depending on the priority relation) on the dataset with the highest proportion of facts in conflicts. Finally, the grounded repair covers a large fraction of the intersection of optimal repairs. Given the cost of checking whether each non-grounded fact holds under X-(I)AR, this suggests the interest of using grounded rather than X-IAR.

Impact of localization and simplified encodings. Unsurprisingly, localization is crucial for feasibility, even on the smallest datasets. Regarding the impact of the encodings tailored to the case where the conflicts are binary, we found that the simplification is generally helpful when X=P, mostly neutral when X=C, and when X=G, we observe a more diverse outcome, with a performance improvement in a majority of cases, but also a significant number of cases where it is the opposite.

Semantics comparison in terms of runtimes. Using globally-optimal repairs proved significantly more difficult in practice than using Pareto- or even completion-optimal repairs, in line with the worst-case complexity, and the difference becomes more pronounced as the proportion of facts involved in some conflicts increases. This suggests that even if we aim to compute G-AR and G-brave answers, a better strategy is to first test whether the candidate answer holds under P-AR semantics and P-brave semantics (which are the fastest optimal repair semantics) and only call the G-AR and G-brave procedures if the status remains unresolved.

Acknowledgments

This work was supported by the ANR AI Chair INTENDED (ANR-19-CHIA-0014); by the Italian Ministry of Industrial Development (MISE) under project EI-TWIN n. F/310168/05/X56 CUP B29J24000680005; and by the Italian Ministry of Research (MUR) under project PNRR FAIR - Spoke 9 - WP 9.1 CUP H23C22000860006.

Declaration on Generative AI

During the preparation of this work, the authors used GPT-5 for: Grammar and spelling check. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] M. Bienvenu, C. Bourgaux, R. Jean, G. Mazzotta, Using ASP(Q) to handle inconsistent prioritized data, in: Proceedings of KR, 2026.
[2] L. E. Bertossi, Database repairs and consistent query answering: Origins and further developments, in: Proceedings of PODS, 2019.
[3] M. Bienvenu, A short survey on inconsistency handling in ontology-mediated query answering, Künstliche Intell. 34 (2020) 443–451.
[4] M. Arenas, L. E. Bertossi, J. Chomicki, Consistent query answers in inconsistent databases, in: Proceedings of PODS, 1999.
[5] D. Lembo, M. Lenzerini, R. Rosati, M. Ruzzi, D. F. Savo, Inconsistency-tolerant semantics for description logics, in: Proceedings of RR, 2010.
[6] M. Bienvenu, R. Rosati, Tractable approximations of consistent query answering for robust ontology-based data access, in: Proceedings of IJCAI, 2013.
[7] C. Bourgaux, Inconsistency-tolerant semantics based on (preferred) repairs, in: Proceedings of Reasoning Web, 2025.
[8] S. Staworko, J. Chomicki, J. Marcinkowski, Prioritized repairing and consistent query answering in relational databases, Annals of Mathematics and Artificial Intelligence (AMAI) 64 (2012) 209–246.
[9] B. Kimelfeld, E. Livshits, L. Peterfreund, Detecting ambiguity in prioritized database repairing, in: Proceedings of ICDT, 2017.
[10] B. Kimelfeld, E. Livshits, L. Peterfreund, Counting and enumerating preferred database repairs, Theor. Comput. Sci. 837 (2020) 115–157.
[11] M. Bienvenu, C. Bourgaux, Querying and repairing inconsistent prioritized knowledge bases: Complexity analysis and links with abstract argumentation, in: Proceedings of KR, 2020.
[12] M. Bienvenu, C. Bourgaux, Inconsistency handling in prioritized databases with universal constraints: Complexity analysis and links with active integrity constraints, in: Proceedings of KR, 2023.
[13] M. Bienvenu, C. Bourgaux, Querying inconsistent prioritized data with ORBITS: algorithms, implementation, and experiments, in: Proceedings of KR, 2022.
[14] M. Bienvenu, C. Bourgaux, K. Inoue, R. Jean, A rule-based approach to specifying preferences over conflicting facts and querying inconsistent knowledge bases, in: Proceedings of KR, 2025.
[15] T. Eiter, G. Gottlob, On the computational cost of disjunctive logic programming: Propositional case, Ann. Math. Artif. Intell. 15 (1995) 289–323.
[16] M. Gebser, R. Kaminski, T. Schaub, Complex optimization in answer set programming, Theory Pract. Log. Program. 11 (2011) 821–839.
[17] G. Amendola, B. Cuteri, F. Ricca, M. Truszczynski, Solving problems in the polynomial hierarchy with ASP(Q), in: Proceedings of LPNMR, 2022.

[18] G. Amendola, F. Ricca, M. Truszczynski, Beyond NP: quantifying over answer sets, Theory Pract. Log. Program. 19 (2019) 705–721.
[19] G. Brewka, T. Eiter, M. Truszczynski, Answer set programming at a glance, Commun. ACM 54 (2011) 92–103.
[20] M. Gelfond, V. Lifschitz, Classical negation in logic programs and disjunctive databases, New Gener. Comput. 9 (1991) 365–386.
[21] W. Faber, G. Pfeifer, N. Leone, Semantics and complexity of recursive aggregates in answer set programming, Artif. Intell. 175 (2011) 278–298.
[22] M. Bienvenu, C. Bourgaux, F. Goasdoué, Querying inconsistent description logic knowledge bases under preferred repair semantics, in: Proceedings of AAAI, 2014.
[23] C. Bourgaux, Inconsistency handling in ontology-mediated query answering. (Gestion des incohérences pour l'accès aux données en présence d'ontologies), Ph.D. thesis, University of Paris-Saclay, France, 2016.
[24] C. Lutz, I. Seylan, D. Toman, F. Wolter, The combined approach to OBDA: Taming role hierarchies using filters, in: Proceedings of ISWC, 2013.
[25] C. Andrea, G. Mazzotta, F. Ricca, 2-ASP(Q) solving based on CEGAR, in: Proceedings of AAAI, 2026.
[26] M. Gebser, R. Kaminski, B. Kaufmann, T. Schaub, Multi-shot ASP solving with clingo, CoRR abs/1705.09811 (2017).