Preliminary Report on Scalable Optimal-Repair Based Query Answering with Non-Binary Conflicts

Meghyn Bienvenu[1], Camille Bourgaux[2] and Ahmed Laouar[1,*]

[1] CNRS & University of Bordeaux, France
[2] DI ENS, ENS, CNRS, PSL University & Inria, Paris, France

Abstract

We present our ongoing work on implementing and benchmarking scalable SAT-based procedures for query answering under variants of three well-known inconsistency-tolerant semantics (AR, brave and IAR) based on two notions of optimal repairs (Pareto- and completion-optimal) that exploit a priority relation between conflicting facts. We focus in particular on comparing different SAT encodings that can handle non-binary conflicts.

Keywords

Ontology-mediated query answering, Inconsistency-tolerant semantics, Prioritized data, SAT encodings

1. Optimal Repair-Based Inconsistency-Tolerant Semantics

Handling data that is inconsistent with respect to expressed constraints arises in many practical settings, whether constraints are given by database dependencies (such as functional dependencies, or denial constraints) or by ontologies (such as a description logic TBox capturing domain knowledge, in which case the dataset is called an ABox). When cleaning the data to restore consistency is not possible, an alternative principled approach is to adopt inconsistency-tolerant semantics to extract meaningful information from contradictory data [1]. A central notion is that of a *repair*, defined as a maximal subset of the data that is consistent with the constraints. Since there are typically many repairs, and it is unknown which reflects the true state of the world, several repair-based semantics have been proposed. The consistent query answering approach from the database area [2, 3], called ABox Repair (*AR*) semantics [4] by KR researchers, returns those answers that hold in *every repair*. The *brave* semantics [5] considers those that hold in *at least one* repair, while the Intersection of ABox Repairs (*IAR*) semantics [4] returns answers that hold in the *intersection* of all repairs. A priority relation between conflicting facts [6, 7] can be used to define variants of the AR, brave, and IAR semantics by focusing on different kinds of optimal repairs — in particular, Pareto- and completion-optimal repairs.

Formally, we consider a knowledge base (KB) $\mathcal{K} = (\mathcal{D}, \mathcal{T})$ where $\mathcal{D}$ is a finite set of facts and $\mathcal{T}$ is a logical theory, which can be an ontology (e.g. in the DL-Lite family [8]) or a set of database constraints (e.g. denial constraints, which include functional dependencies). We denote by $SRep(\mathcal{K})$ the set of repairs of $\mathcal{K}$ and by $Conf(\mathcal{K})$ the set of *conflicts* of $\mathcal{K}$, where a conflict is a minimal subset of $\mathcal{D}$ inconsistent with $\mathcal{T}$. A *priority relation* $\succ$ for $\mathcal{K}$ is an acyclic binary relation over $\mathcal{D}$ such that $\alpha \succ \beta$ implies that $\{\alpha, \beta\}$ 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 *prioritized KB* $\mathcal{K}_\succ$ is a KB $\mathcal{K}$ with a priority relation $\succ$ for $\mathcal{K}$. A repair $\mathcal{R}$ is *Pareto-optimal* if no fact outside the repair is strictly preferred to every fact it would replace: there does not exist any $\mathcal{R}'$ such that $(\mathcal{R}', \mathcal{T})$ is consistent and there is $\beta \in \mathcal{R}' \setminus \mathcal{R}$ such that $\beta \succ \alpha$ for every $\alpha \in \mathcal{R} \setminus \mathcal{R}'$. A repair is *completion-optimal* if it is a Pareto-optimal repair of $\mathcal{K}_{\succ'}$ for some completion $\succ'$ of $\succ$, where a *completion* of $\succ$ is a total priority relation $\succ' \supseteq \succ$. We denote by $PRep(\mathcal{K}_\succ)$ and $CRep(\mathcal{K}_\succ)$ the sets of Pareto- and completion-optimal repairs, respectively. It is known that $CRep(\mathcal{K}_\succ) \subseteq PRep(\mathcal{K}_\succ) \subseteq SRep(\mathcal{K})$. For $X \in \{S, P, C\}$, given a conjunctive query

---

LINDA'26: Logical approaches to handling INconsistent DAta, July 24th 2026, Lisbon, Portugal

[*] Corresponding author.

meghyn.bienvenu@labri.fr (M. Bienvenu); camille.bourgaux@ens.fr (C. Bourgaux); ahmed.laouar@labri.fr (A. Laouar)

$q(\vec{x})$ and a candidate answer $\vec{a}$, the tuple $\vec{a}$ is an answer under X-AR (resp. X-brave, X-IAR) semantics if it holds in every (resp. some, the intersection of all) $\mathcal{R} \in XRep(\mathcal{K}_\succ)$.

Query answering under these semantics is (co)NP-complete in data complexity across the most common settings [6, 7], which naturally suggests the interest of using SAT solvers. Our work aims at extending the ORBITS system [9], which has been developed to allow for a comprehensive comparison of different SAT encodings and algorithms for X-AR, X-brave and X-IAR semantics, while being agnostic to the precise setting (database or ontology-mediated query answering). The system takes as input the conflicts $Conf(\mathcal{K})$, priority relation $\succ$, and a set of potential query answers $PotAns$ associated with the set of their *causes* $Causes(q(\vec{a}), \mathcal{K})$, where a cause of $q(\vec{a})$ is an inclusion-minimal subset $\mathcal{C} \subseteq \mathcal{D}$ such that $(\mathcal{C}, \mathcal{T})$ is consistent and entails $q(\vec{a})$. The SAT encodings used by ORBITS rely on the following characterization of the semantics: $q(\vec{a})$ holds under X-AR semantics iff every $\mathcal{R} \in XRep(\mathcal{K}_\succ)$ contains some cause of $q(\vec{a})$; it holds under X-brave semantics iff there exists $\mathcal{R} \in XRep(\mathcal{K}_\succ)$ that contains a cause of $q(\vec{a})$; and it holds under X-IAR semantics if there exists a cause of $q(\vec{a})$ which is included in every $\mathcal{R} \in XRep(\mathcal{K}_\succ)$. However, the encodings implemented in ORBITS assume that conflicts are binary, i.e. each $\mathcal{C} \in Conf(\mathcal{K})$ satisfies $|\mathcal{C}| \leq 2$. This paper focuses on the implementation within ORBITS of SAT encodings for conflicts of *arbitrary size* (n-ary conflicts). Our ongoing work also includes the computation of the conflicts in different settings and the construction of new benchmarks.

Besides ORBITS, several other systems implement some repair-based semantics (based on standard repairs or other notions of preferred repairs, cf. [10, Table 8] for an overview). The most relevant for the present work is CAvSAT [11], which employs SAT solvers to compute the consistent query answers over relational databases with denial constraints (allowing n-ary conflicts, but without a priority relation).

2. SAT Encodings: From Binary to Arbitrary Conflicts

Before discussing the case of non-binary conflicts, let us first recall one particular encoding for the binary-conflict case that is implemented in ORBITS for the *P-AR semantics*, which uses the encoding variants neg_1 for contradicting causes and P_1-max for Pareto-optimality defined in [9]. Like all of the SAT encodings in [9], this encoding uses propositional variable x_α to represent the inclusion of $\alpha \in \mathcal{D}$ in a repair candidate, and it is constructed in a modular manner from building block formulas:

- **Contradiction formula** ($\varphi_{\neg q} = \bigwedge_{\mathcal{C} \in Causes(q(\vec{a}), \mathcal{K})} \varphi_{\neg \mathcal{C}}$): for each $\mathcal{C} \in Causes(q(\vec{a}), \mathcal{K})$, the formula $\varphi_{\neg \mathcal{C}}$ ensures that at least one fact conflicting with a fact of $\mathcal{C}$ and not dominated by it w.r.t. $\succ$ must be included, preventing the full cause from being present. Blocking all causes blocks the query answer.
- **Maximality formula** ($\varphi_{\mathsf{P\text{-}max}}$) ensures that the repair candidate can be extended to a Pareto-optimal repair w.r.t. $\succ$. It verifies that every exclusion of a relevant fact is justified by the inclusion of some strictly preferred or incomparable conflicting fact.
- **Consistency formula** (φ_{cons}) ensures that no conflict is included in the repair candidate: for each $\mathcal{C} \in Conf(\mathcal{K})$ whose facts correspond to variables in the above formulas, at least one fact is excluded.

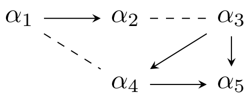
We shall illustrate on the following example how this encoding works:

Example 1. *Consider the knowledge base* $\mathcal{K} = (\mathcal{D}, \mathcal{T})$ *whose dataset* $\mathcal{D} = \{\alpha_1, \ldots, \alpha_5\}$ *consists of:*

$$\alpha_1 = \mathsf{Unpaid}(a) \quad \alpha_2 = \mathsf{Paid}(a) \quad \alpha_3 = \mathsf{Emeritus}(a) \quad \alpha_4 = \mathsf{Prof}(a) \quad \alpha_5 = \mathsf{PhDStud}(a)$$

The TBox $\mathcal{T}$ *is given on the left below. The conflicts and the priority relation* $\succ$ *are depicted on the right, where (dashed) lines represent conflicts and arrows represent priorities (from more to less preferred facts):*

$\mathsf{Paid} \sqsubseteq \neg\mathsf{Unpaid}$
$\mathsf{Prof} \sqsubseteq \neg\mathsf{Emeritus}$
$\mathsf{Prof} \sqsubseteq \mathsf{SeniorStaff}$
$\mathsf{Emeritus} \sqsubseteq \mathsf{Unpaid}$
$\mathsf{Prof} \sqsubseteq \neg\mathsf{PhDStud}$
$\mathsf{Emeritus} \sqsubseteq \mathsf{SeniorStaff}$
$\mathsf{Prof} \sqsubseteq \mathsf{Paid}$
$\mathsf{PhDStud} \sqsubseteq \neg\mathsf{Emeritus}$



The prioritized knowledge base $\mathcal{K}_\succ$ *admits two Pareto-optimal repairs:*

$$PRep(\mathcal{K}_\succ) = \{\{\alpha_1 = \mathsf{Unpaid}(a), \alpha_3 = \mathsf{Emeritus}(a)\}, \{\alpha_2 = \mathsf{Paid}(a), \alpha_4 = \mathsf{Prof}(a)\}\}$$

Consider the query $q(x) \leftarrow \mathsf{SeniorStaff}(x)$ *and candidate answer* a*:* $Causes(q(a), \mathcal{K}) = \{\{\alpha_3\}, \{\alpha_4\}\}$.

When applied to this example, the binary encoding from [9] for query answering under P-AR semantics described above introduces four propositional variables, one for each relevant fact: x_{α_1}, x_{α_2}, x_{α_3}, and x_{α_4} for α_1, α_2, α_3, and α_4, respectively. Note that α_5 is not relevant, in particular because it cannot be used to contradict a cause in the contradiction formula, being less preferred than α_3 and α_4. The formula $\Phi_{\mathsf{P\text{-}AR}}(q(a)) = \varphi_{\neg q} \land \varphi_{\mathsf{P\text{-}max}} \land \varphi_{\mathsf{cons}}$ is constructed from the building blocks as follows:

$\varphi_{\neg q}$: $(x_{\alpha_2}) \land (x_{\alpha_1} \lor x_{\alpha_3})$ *contradicts α_3 by enforcing α_2, and α_4 by enforcing α_1 or α_3

$\varphi_{\mathsf{P\text{-}max}}$: $(x_{\alpha_1} \lor x_{\alpha_4}) \land (x_{\alpha_2} \lor x_{\alpha_1} \lor x_{\alpha_3}) \land (x_{\alpha_3} \lor x_{\alpha_2}) \land (x_{\alpha_4} \lor x_{\alpha_1} \lor x_{\alpha_3})$
*enforces Pareto-optimality by ensuring that no preferred fact is unnecessarily excluded

φ_{cons} : $(\neg x_{\alpha_1} \lor \neg x_{\alpha_2}) \land (\neg x_{\alpha_1} \lor \neg x_{\alpha_4}) \land (\neg x_{\alpha_2} \lor \neg x_{\alpha_3}) \land (\neg x_{\alpha_3} \lor \neg x_{\alpha_4})$ *ensures no conflict is included

yielding a CNF with **4 variables** and **10 clauses**. The formula is unsatisfiable, which means that $q(a)$ holds under P-AR semantics. Indeed, we can see that each of the two Pareto-optimal repairs contains either $\alpha_3 = \mathsf{Emeritus}(a)$ or $\alpha_4 = \mathsf{Prof}(a)$, both of which entail $\mathsf{SeniorStaff}(a)$ via $\mathcal{T}$.

2.1. Handling Non-Binary Conflicts

The preceding encoding exploits the fact that each conflict involves at most two facts (and that self-inconsistent facts are tackled in a preprocessing step so that the remaining conflicts contain exactly two facts), which allows for a compact and efficient CNF construction. However, real-world knowledge bases can naturally contain conflicts of larger sizes. For instance, consider the constraint stating that a supervisor must have defended their PhD before their supervisee: $\mathsf{supervisor}(x, y) \land \mathsf{phdDefenseDate}(x, t_1) \land \mathsf{phdDefenseDate}(y, t_2) \land t_1 \geq t_2 \rightarrow \bot$. A violation of this constraint yields a conflict involving three facts. SAT procedures to handle n-ary conflicts have already been proposed in an extended version of the ORBITS paper [9], but have not been implemented. SAT-based encodings for n-ary conflicts have also been proposed and implemented in the CAvSAT system [11] for query answering under the standard AR semantics in the database setting with denial constraints. The aim of this work is to implement the proposed encodings for optimal repair-based semantics, evaluate them experimentally, and propose improvements that preserve the compactness of binary-conflict encodings whenever possible.

The encodings suggested in ORBITS [9] for n-ary conflicts rely upon auxiliary variables. Specifically, the contradiction part of the encoding introduces an auxiliary variable $x_{\mathcal{C},\mathcal{B}}$ for each cause $\mathcal{C}$ and each set of facts $\mathcal{B}$ that can be used to contradict some $\alpha \in \mathcal{C}$ (i.e. $\mathcal{B} \cup \{\alpha\}$ is a conflict *and* for all $\beta \in \mathcal{B}$, $\alpha \not\succ \beta$), which is used to force that all facts in some such set $\mathcal{B}$ are included, thereby blocking the inclusion of α (hence $\mathcal{C}$). In a similar manner, the maximality part of the encoding introduces auxiliary variables of the form $x_{\alpha,\mathcal{B}}$ for each 'relevant' fact α (with relevance defined via a suitable notion of reachability in the associated directed conflict hypergraph) and for each conflicting set $\mathcal{B}$ for α, which serve to ensure that if x_α is set to false, then there must be a full conflicting (and not less preferred) set $\mathcal{B}$ to justify α's exclusion, so the repair candidate extends to a Pareto-optimal repair.

The changes to the encoding introduced to handle n-ary conflicts come, however, at a cost: even when all conflicts in the considered KB happen to be binary, the n-ary encoding strategy produces a larger formula than the binary-encoding one, as we now illustrate on our running example. The n-ary encoding from [9] utilizes a total of 13 variables: four base variables x_{α_i} for $i \in \{1, 2, 3, 4\}$, three 'contradiction' auxiliary variables and six 'maximality' auxiliary variables, each associated respectively with a cause and a conflict, or with a reachable fact and a conflict (their indexing is visible in the encoding below). The CNF formula $\Phi_{\mathsf{P\text{-}AR}}(q(a)) = \varphi_{\neg q} \land \varphi_{\mathsf{P\text{-}max}} \land \varphi_{\mathsf{cons}}$ under the n-ary encoding is constructed from the following three modified building blocks:

$$\begin{array}{rlr}
\varphi_{\neg q}: & (x_{(\{\alpha_3\},\{\alpha_2\})}) \wedge (\neg x_{(\{\alpha_3\},\{\alpha_2\})} \vee x_{\alpha_2}) \wedge & \text{*contradicts } \alpha_3 \\
& (x_{(\{\alpha_4\},\{\alpha_1\})} \vee x_{(\{\alpha_4\},\{\alpha_3\})}) \wedge (\neg x_{(\{\alpha_4\},\{\alpha_1\})} \vee x_{\alpha_1}) \wedge (\neg x_{(\{\alpha_4\},\{\alpha_3\})} \vee x_{\alpha_3}) & \text{*contradicts } \alpha_4 \\
\varphi_{\text{P-max}}: & (x_{\alpha_1} \vee x_{(\alpha_1,\{\alpha_4\})}) \wedge (\neg x_{(\alpha_1,\{\alpha_4\})} \vee x_{\alpha_4}) \wedge & \text{*P-optimality for } \alpha_1 \\
& (x_{\alpha_2} \vee x_{(\alpha_2,\{\alpha_1\})} \vee x_{(\alpha_2,\{\alpha_3\})}) \wedge (\neg x_{(\alpha_2,\{\alpha_1\})} \vee x_{\alpha_1}) \wedge (\neg x_{(\alpha_2,\{\alpha_3\})} \vee x_{\alpha_3}) \wedge & \text{*P-optimality for } \alpha_2 \\
& (x_{\alpha_3} \vee x_{(\alpha_3,\{\alpha_2\})}) \wedge (\neg x_{(\alpha_3,\{\alpha_2\})} \vee x_{\alpha_2}) \wedge & \text{*P-optimality for } \alpha_3 \\
& (x_{\alpha_4} \vee x_{(\alpha_4,\{\alpha_1\})} \vee x_{(\alpha_4,\{\alpha_3\})}) \wedge (\neg x_{(\alpha_4,\{\alpha_1\})} \vee x_{\alpha_1}) \wedge (\neg x_{(\alpha_4,\{\alpha_3\})} \vee x_{\alpha_3}) & \text{*P-optimality for } \alpha_4 \\
\varphi_{\text{cons}}: & (\neg x_{\alpha_1} \vee \neg x_{\alpha_4}) \wedge (\neg x_{\alpha_1} \vee \neg x_{\alpha_2}) \wedge (\neg x_{\alpha_2} \vee \neg x_{\alpha_3}) \wedge (\neg x_{\alpha_3} \vee \neg x_{\alpha_4}) & \text{*no conflict fully included}
\end{array}$$

yielding a CNF with **13 variables** and **19 clauses**, compared to **4 variables** and **10 clauses** for the binary encoding on the same example, despite all conflicts being binary. The overhead stems entirely from the auxiliary variables introduced by the n-ary encoding strategy to handle conflicts of arbitrary size, which are unnecessary here. This illustrates the core challenge: supporting n-ary conflicts without penalizing the binary cases that arise most frequently in practice.

Hybrid encoding. The first improvement we propose exploits the observation that binary conflicts do not require auxiliary variables: when a conflict $\mathcal{C}$ satisfies $|\mathcal{C}| = 2$, the encoding can directly use the base variables x_α rather than introducing auxiliary variables $x_{\mathcal{C},\mathcal{B}}$ or $x_{\alpha,\mathcal{B}}$. The hybrid encoding therefore treats binary and n-ary conflicts differently within the same formula: binary conflicts are handled as in the original ORBITS encoding, while auxiliary variables are introduced only for conflicts of size strictly greater than two. When all conflicts are binary, the hybrid encoding coincides exactly with the binary encoding of ORBITS, recovering its compactness. When n-ary conflicts are present alongside binary ones, the overhead is limited to those conflicts that genuinely require it.

Variable sharing. The second improvement targets the auxiliary variables themselves. In the original n-ary encoding proposed in [9], auxiliary variables are indexed by pairs $(\mathcal{C}, \mathcal{B})$ or $(\alpha, \mathcal{B})$, meaning that the same set of facts $\mathcal{B}$ may give rise to multiple distinct variables across different building blocks. We propose an encoding (shared-vars) that replaces all such variables by a single shared variable $x_\mathcal{B}$, indexed only by the set $\mathcal{B}$ itself. This variable sharing reduces the total number of auxiliary variables, and applies uniformly to all variants of $\varphi_{\neg q}$ and $\varphi_{\text{P-max}}$. A similar variable-sharing strategy is adopted in CAvSAT [11], where auxiliary variables are indexed by (fact, near-violation) pairs and thus shared across causes involving the same fact. However, CAvSAT introduces auxiliary variables for every fact involved in some conflict or cause, whereas our approach, inheriting the query-focused design of ORBITS, restricts them to facts appearing in or reachable from the query's causes.

Combined encoding. Both improvements can be applied simultaneously, yielding a combined encoding that is both hybrid and implements variable sharing: binary conflicts are handled without auxiliary variables, and the auxiliary variables introduced for n-ary conflicts are shared across building blocks. Using our example, the encoding recovers exactly the binary encoding of ORBITS (4 variables, 10 clauses), a direct consequence of the hybrid component eliminating auxiliary variables when a conflict is binary. The combined encoding thus achieves the best of both worlds: it handles n-ary conflicts when needed, without penalizing cases where conflicts are binary.

2.2. Implementation and Preliminary Experiments

We implemented a Java-based system, extending the initial ORBITS system, to support n-ary conflicts. The system takes as input a hypergraph representation of conflicts, a priority relation and a set of potential answers with their causes, which can be either provided directly or computed by the system from raw data and constraints using a dedicated pre-processing module. It implements the four variants of the n-ary encoding: n-ary, shared-vars, hybrid and shared-vars-hybrid. It also supports the original strategy of the ORBITS system for binary conflicts, which relies on a different data structure to represent the conflicts and priority relation (directed conflict graph), allowing for a more direct construction of the encodings when the conflicts are known to be binary.

We report here on our preliminary evaluation on the Food Inspection benchmark from [9], originally used by Dixit and Kolaitis [11] to evaluate the n-ary SAT encoding of CAvSAT. It contains data on restaurant inspections in New York and Chicago, with 523K facts across four relations, each with a key constraint and one with an additional functional dependency. All conflicts are binary, and around 37% of facts belong to some conflict, yielding 192K conflicting facts and 219K conflicts. We use the six queries from the original benchmark, with priority relations as defined in [9]. We use the Simple algorithm of ORBITS [9] with the SAT4J solver. The experiments were run with a timeout of 15 minutes, and the reported times are averages over 3 repetitions.

		q1	**q2**	**q3**	**q4**	**q5**	**q6**
#potential answers		1	450	52	31 997	16 926	1 062
#P-AR answers		1	435	18	26 062	15 390	287
binary	enc	—	592	422	43 836	11 717	19 874
	sol	—	400	273	38 301	10 345	10 532
	filter	**16**	**1 025**	**718**	**82 222**	**22 118**	**30 511**
n-ary	enc	—	1 891	1 050	TIMEOUT	TIMEOUT	TIMEOUT
	sol	—	19 408	21 121			
	filter	55	21 371	22 241			
shared-vars	enc	—	1 832	867	129 982	36 194	67 644
	sol	—	413	352	42 823	11 074	11 616
	filter	54	2 315	1 290	172 929	47 362	79 408
hybrid	enc	—	1 332	735	96 770	27 001	40 907
	sol	—	381	281	36 395	9 867	10 497
	filter	55	1 785	1 083	133 294	36 957	51 567
shared-vars-hybrid	enc	—	1 337	745	99 184	27 130	40 693
	sol	—	380	282	36 303	9 851	10 581
	filter	58	1 792	1 102	135 615	37 073	51 426

Table 1
Query statistics and answer filtering time (ms) under the P-AR semantics, on the Food Inspection benchmark.

Table 1 reports query statistics and the total encoding (enc), solving (sol), and filtering times (in ms, summed over all candidate answers) per strategy. As expected, the binary strategy remains the fastest across all non-trivial queries (shown in **bold**), since it was specifically designed for this case. The n-ary encoding performs poorly in this setting, timing out on three out of six queries and being significantly slower on the remaining ones, confirming that a naïve uniform treatment of conflicts is inadequate even when all conflicts happen to be binary. The hybrid and shared-vars-hybrid strategies recover most of the efficiency of the binary encoding, consistently outperforming both the n-ary and shared-vars strategies. The shared-vars strategy alone, however, brings a lesser benefit. Note that, although the binary, hybrid, and shared-vars-hybrid strategies produce the same encoding – yielding comparable solving times (highlighted in orange) – their total filtering times differ due to encoding overheads.

3. Ongoing and Future Work

Several directions remain open. Experiments on benchmarks with genuine n-ary conflicts are needed to fully assess the potential of the shared-vars and shared-vars-hybrid strategies, as the current evaluation is limited to a binary-conflict dominated setting. Several existing benchmarks can be adapted: the Physicians benchmarks from [9], the CQAPri benchmark, also adapted in [9] and further extended with n-ary conflicts in recent work on ASP-based rules [12] and the StackOverflow benchmark from [13], which provides real-world relational data with naturally occurring conflicts. We are also in the process of creating a new benchmark, based on the DBpedia ontology [14]. A main challenge is the generation of n-ary conflicts: identifying the right constraints, modeling them in a form that fits our framework, and efficiently computing the resulting conflicts on large real-world datasets are all non-trivial tasks that shape the design of the benchmark. Finally, evaluating the relative performance of the different semantics (X-AR, X-brave and X-IAR for $X \in \{S, P, C\}$) under the n-ary conflict framework is also part of our ongoing work. Going further, we plan to explore how best to implement and exploit tractable approximations, such as the grounded semantics from [7] and the k-support semantics from [5].

Acknowledgments

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

References

[1] M. Bienvenu, C. Bourgaux, Inconsistency-tolerant querying of description logic knowledge bases, in: Tutorial Lectures of the 12th International Reasoning Web Summer School (RW), 2016.

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

[3] L. E. Bertossi, Database repairs and consistent query answering: Origins and further developments, in: Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS), 2019.

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

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

[6] 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.

[7] M. Bienvenu, C. Bourgaux, Querying and repairing inconsistent prioritized knowledge bases: Complexity analysis and links with abstract argumentation, in: Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning (KR), 2020.

[8] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati, Tractable reasoning and efficient query answering in description logics: The DL-Lite family, Journal of Automated Reasoning (JAR) (2007).

[9] M. Bienvenu, C. Bourgaux, Querying inconsistent prioritized data with ORBITS: Algorithms, implementation, and experiments, in: Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning (KR), 2022.

[10] C. Bourgaux, Inconsistency-tolerant semantics based on (preferred) repairs, in: Joint Proceedings of the 20th and 21st Reasoning Web Summer Schools (RW), 2025.

[11] A. A. Dixit, P. G. Kolaitis, A SAT-based system for consistent query answering, in: Proceedings of the 22nd International Conference on Theory and Applications of Satisfiability Testing (SAT), 2019.

[12] M. Bienvenu, C. Bourgaux, R. Jean, A rule-based approach to specifying preferences over conflicting facts and querying inconsistent knowledge bases, in: Proceedings of the 22nd International Conference on Principles of Knowledge Representation and Reasoning (KR), 2025.

[13] Z. Fan, P. Koutris, X. Ouyang, J. Wijsen, LinCQA: Faster consistent query answering with linear time guarantees, Proceedings of the ACM on Management of Data (PACMMOD) 1 (2023) 38:1–38:25.

[14] DBpedia Association, DBpedia Ontology (DBO) — DBpedia Association, https://www.dbpedia.org/resources/ontology/, 2025. Accessed: 2025-10-03.