Range Consistent Query Answering via Rewriting

Jef Wijsen

University of Mons

LINDA 2026

- Joint work with Aziz Amezian El Khalfioui.
- Work continuing the seminal works:

Arenas, M., Bertossi, L. E., and Chomicki, J. (1999).
Consistent query answers in inconsistent databases.
In *PODS*, pages 68–79. ACM Press.

Arenas, M., Bertossi, L. E., and Chomicki, J. (2001).
Scalar aggregation in FD-inconsistent databases.
In *ICDT*, volume 1973 of *Lecture Notes in Computer Science*, pages 39–53. Springer.

Consistent Query Answering for Boolean Queries (a.k.a. Cautious Semantics)

Fix:

- Boolean query q, mapping database instances to $\mathbb{B} = \{0, 1\}$ with $0 < 1$;
- set of constraints Σ.

CQA(q, Σ).

INPUT: An inconsistent database instance.

OUTPUT: $\min \{q(\mathbf{r}) \mid \mathbf{r}$ is a repair of $\mathbf{db}$ w.r.t. $\Sigma\}$

Note:

- An output of 1 indicates that the query is true in every repair.
- Replacing "min" with "max" yields brave semantics: an output of 1 indicates that the query is true in some repair.

Consistent Query Answering for Numerical Queries

Fix:

- Numerical query q, mapping database instances to $\mathbb{Q}_{\geq 0}$;
- set of constraints Σ.

MIN-CQA(q, Σ).

INPUT: An inconsistent database instance.

OUTPUT: $\min \{q(\mathbf{r}) \mid \mathbf{r}$ is a repair of $\mathbf{db}$ w.r.t. $\Sigma\}$[1]

MAX-CQA(q, Σ).

INPUT: An inconsistent database instance.

OUTPUT: $\max \{q(\mathbf{r}) \mid \mathbf{r}$ is a repair of $\mathbf{db}$ w.r.t. $\Sigma\}$

[1] We only consider settings where the set of repairs is finite.

Example

R	<u>*Dep*</u>	*City*	*Revenue*
	Toys	Rome	100
	Toys	Paris	100
	Shoes	Paris	80
	Shoes	Paris	90

$$q: \quad \texttt{SUM}(r) \leftarrow R(\underline{x}, \text{Paris}, r)$$

$$\Sigma = \{ \ \texttt{R PRIMARY KEY(Dep)} \ \}$$

Repairs are defined as maximal consistent subinstances.

- MIN-CQA(q, Σ) returns 80, i.e., the minimum query answer over all repairs is 80.
- MAX-CQA(q, Σ) returns 190, i.e., the maximum query answer over all repairs is 190.

Numerical Queries

Self-join-free CQs with aggregation:

Syntax:

$$\texttt{AGG}(r) \leftarrow R_1(\vec{x}_1), \ldots, R_n(\vec{x}_n)$$

where

- either $r \in \mathbb{Q}_{\geq 0}$, or r is a variable ranging over columns with entries in $\mathbb{Q}_{\geq 0}$;
- $R_i \neq R_j$ if $i \neq j$; and
- $\texttt{AGG}$ refers to an aggregate operator (e.g., $\texttt{SUM}$).

Semantics: Let $\theta_1, \ldots, \theta_\ell$ be all valuations for the variables in the rule that map every atom in the rule body to a database fact. Then, the answer is

$$\mathsf{AGG}\left(\{\{\theta_1(r), \ldots, \theta_\ell(r)\}\}\right).$$

Constraints

- The only constraints considered are primary key constraints, one per relation.
- A repair of a database instance is obtained by selecting, for each relation, a maximal consistent subset of its tuples.
- We write $R(\underline{x_1, \ldots, x_k}, x_{k+1}, \ldots, x_n)$ to denote that the primary key of R consists of its first k attributes.
- The set of constraints Σ is thus determined by the query and will henceforth be omitted.

Research Challenge

- An inconsistent database instance may have exponentially many repairs.
- For which numerical queries q can MIN-CQA(q) and/or MAX-CQA(q) be computed in polynomial time (data complexity)?
- For which numerical queries q can MIN-CQA(q) and/or MAX-CQA(q) be expressed in first-order aggregate logic?

Our focus is on the second question.

First-Order Aggregate Logics

- AGGR[nr-datalog] extends nr-datalog with rules of the form:

$$P(\vec{x}, \mathtt{AGG}(r)) \leftarrow P_1(\vec{x}_1), \ldots, P_n(\vec{x}_n).$$

Semantics are defined in the standard way. For example,

$$P(x, \mathtt{AGG}(r)) \leftarrow Body(x, y, r)$$

P	x	$\mathtt{AGG}(r)$
	a	AGG $(\{\{1, 2\}\})$
	b	AGG $(\{\{3, 3\}\})$

$Body$	x	y	r
	a	c	1
	a	c	2
	b	c	3
	b	d	3

- AGGR[FO] further allows negation (but no recursion).

Rewriting into AGGR[FO] (Examples)

R	$\underline{Dep}$	$City$	$Revenue$
	Toys	Rome	100
	Toys	Paris	100
	Shoes	Paris	80
	Shoes	Paris	90

Original numerical query:

$$\mathtt{SUM}(r) \leftarrow R(\underline{x}, \text{Paris}, r)$$

Expression in AGGR[nr-datalog] for MAX-CQA(q):

$$\begin{aligned} S(x, \mathtt{MAX}(r)) &\leftarrow R(\underline{x}, \text{Paris}, r) \\ T(\mathtt{SUM}(m)) &\leftarrow S(x, m) \end{aligned}$$

Expression in AGGR[FO] for MIN-CQA(q):

$$\begin{aligned} N(x) &\leftarrow R(\underline{x}, y, r), y \neq \text{Paris} \\ S(x, \mathtt{MIN}(r)) &\leftarrow R(\underline{x}, \text{Paris}, r), \neg N(x) \\ T(\mathtt{SUM}(m)) &\leftarrow S(x, m) \end{aligned}$$

Inexpressibility Results

- For $q: \quad \texttt{SUM}(r) \leftarrow R(\underline{x}, y, r), T(\underline{y}, x)$, neither MIN-CQA($q$) nor MAX-CQA($q$) is expressible in AGGR[FO].
- For $q: \quad \texttt{SUM}(r) \leftarrow R(\underline{x}, y, r), T_1(\underline{y}, x), T_2(\underline{y}, x)$, MIN-CQA($q$) is expressible in AGGR[FO], but MAX-CQA(q) is not.

Source of Inexpressibility

Theorem ([Hella et al., 2001])

Each query in AGGR[FO] *is Hanf-local.*

For $q: \quad \texttt{SUM}(r) \leftarrow R(\underline{x}, y, r), T(\underline{y}, x)$, MAX-CQA($q$) is not local (nor is MIN-CQA(q)).

R	$\underline{x}$	y	r
	d1	Rome	100
	d1	Paris	70
	d2	Oslo	100
	d2	Paris	70
	d3	Rome	70
	d3	Oslo	70

T	$\underline{y}$	x
	Rome	d1
	Rome	d3
	Paris	d1
	Paris	d2
	Oslo	d2
	Oslo	d3

MAX-CQA(q) returns 240, computed by solving a maximum-weight bipartite matching problem (BPM).

MAX-CQA(q) is Not Dual to MIN-CQA(q)

Consider:

$$q : \mathtt{SUM}(r) \leftarrow R(\underline{x}, y, r),\ T_1(\underline{y}, x),\ T_2(\underline{y}, x).$$

R

$\underline{x}$	y	r	
d1	Rome	1	
d2	Rome	0.5	†
d2	Paris	0.5	

T_1

$\underline{y}$	x	
Rome	d1	†
Rome	d2	
Paris	d2	

T_2

$\underline{y}$	x	
Rome	d1	
Rome	d2	†
Paris	d2	

By picking the †-tuples, we obtain a repair $\mathbf{r}_{\text{min}}$ with $q(\mathbf{r}_{\text{min}}) = 0$.

The maximal query result (1.5) is obtained in the repair that contains the top and bottom rows of each table, matching Rome–d1, and Paris–d2.

- MIN-CQA(q) is in AGGR[FO].
- MAX-CQA(q) is BPM-hard under first-order reductions (and therefore not Hanf-local, and therefore not in AGGR[FO]).

Deciding Rewritability of MIN-CQA(q)

Theorem ([Amezian El Khalfioui and Wijsen, 2024])

Let q be a self-join-free CQ with aggregation, whose aggregate operator is monotone and associative. Then,

MIN-CQA(q) *is expressible in* AGGR[FO] $\iff$ *the attack graph of q is acyclic.*

Monotone, Associative Aggregate Operators

In this study, an aggregate operator is any function

$$\mathsf{AGG} : \mathrm{Bag}_{\mathsf{fin}}(\mathbb{Q}_{\geq}) \to \mathbb{Q}_{\geq}.$$

We say that AGG is

Associative:[2]

if $\mathsf{AGG}(X \uplus Y) = \mathsf{AGG}(\{\{\mathsf{AGG}(X)\}\} \uplus Y)$.

Monotone:

if $\mathsf{AGG}(\{\{x_1, \ldots, x_n\}\}) \leq \mathsf{AGG}(\{\{y_1, \ldots, y_n\}\} \uplus Y)$
whenever $x_i \leq y_i$ for each i.

If AGG is monotone and associative, we suppose $\mathsf{AGG}(\emptyset) = 0$.

[2] $\uplus$ denotes union of multisets.

Deciding Rewritability of MAX-CQA(q)

Theorem ([Amezian El Khalfioui and Wijsen, 2026])

Let

$$q : \mathtt{SUM}(r) \leftarrow R_1(\vec{x}_1), \ldots, R_n(\vec{x}_n)$$

be self-join-free with $r \neq 0$*. Then,*

MAX-CQA(q) *is expressible in* AGGR[FO] $\iff$ *q has a kernel with an acyclic attack graph.*

- The proof of the $\Longleftarrow$ direction only uses that SUM is monotone (over $\mathbb{Q}_{\geq 0}$) and associative.
- If q has a kernel with an acyclic attack graph, then every kernel of q has an acyclic attack graph.

Kernel

Let

$$q : \mathtt{SUM}(r) \leftarrow R_1(\underline{\vec{x}_1}, \vec{y}_1), \ldots, R_n(\underline{\vec{x}_n}, y_n).$$

1. First, construct an irreducible set Σ of FDs satisfying

$$\Sigma \equiv \{ \quad \mathsf{vars}(\vec{x}_i) \rightarrow \mathsf{vars}(\vec{y}_i) \quad \}_{i=1}^{n}.$$

2. Next, use Σ to construct a self-join-free query $\hat{q}$, called kernel:

   - for each $X \rightarrow \{y\}$ in Σ, add to $\hat{q}$ an atom $R(\underline{\vec{x}}, y)$ with $\mathsf{vars}(\vec{x}) = X$;
   - then, add to $\hat{q}$ full-key atoms such that every atom of q is guarded by an atom of $\hat{q}$, and vice versa.[3]

For example,

$$q : \mathtt{SUM}(r) \leftarrow R(\underline{x}, y, r), T_1(\underline{y}, x), T_2(\underline{y}, x)$$

has kernel:

$$\hat{q} : \mathtt{SUM}(r) \leftarrow R_1(\underline{x}, y), R_2(\underline{x}, r), T_2(\underline{y}, x), R_0(\underline{x, y, r})$$

[3] We say that $R(\vec{x})$ is guarded by $S(\vec{y})$ if $\mathsf{vars}(\vec{x}) \subseteq \mathsf{vars}(\vec{y})$.

Conclusion

- We studied the following problem:

  Given a numerical query q of the form

  $$\mathtt{AGG}(r) \leftarrow R_1(\vec{x}_1), \ldots, R_n(\vec{x}_n),$$

  decide whether MIN-CQA(q) *and/or* MAX-CQA(q) *are expressible in aggregate logic.*

- Cautious and brave semantics are special cases obtained by interpreting the Boolean domain $\{0, 1\}$ numerically.
- We solved the problem for `SUM`-queries with a self-join-free body.

References

Amezian El Khalfioui, A. and Wijsen, J. (2024).
Computing range consistent answers to aggregation queries via rewriting.
Proc. ACM Manag. Data, 2(5):218:1–218:19.

Amezian El Khalfioui, A. and Wijsen, J. (2026).
Computing consistent least upper bounds in aggregate logic.
In *ICDT*, LIPIcs, pages 4:1–4:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Hella, L., Libkin, L., Nurmonen, J., and Wong, L. (2001).
Logics with aggregate operators.
J. ACM, 48(4):880–907.