Rewriting, Answering, and Losslessness: A Clarification by the "Four Italians"

Diego Calvanese

KRDB Research Centre for Knowledge and Data
Free University of Bozen-Bolzano, Italy

Department of Computing Science
Umeå University, Sweden

VardiFest

31 July – 1 August 2022 – Haifa, Israel

View-based query processing (VBQP)

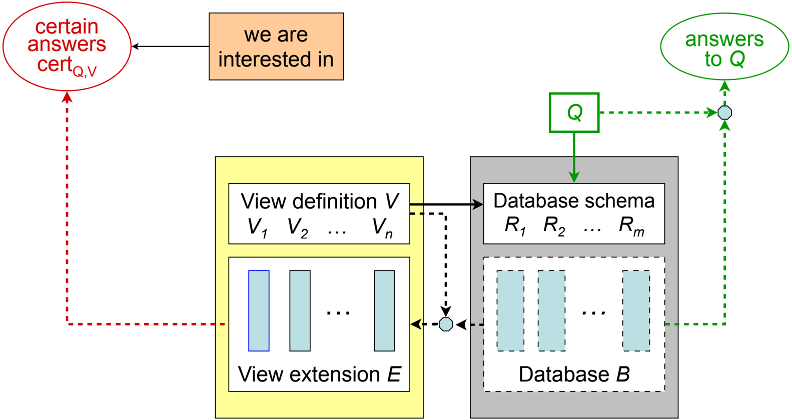
VBQP amounts to computing the answer to a query by **relying solely on a set of views**

Relevant problem in data integration, data warehousing, query optimization, authorization, etc.

Two different approaches:

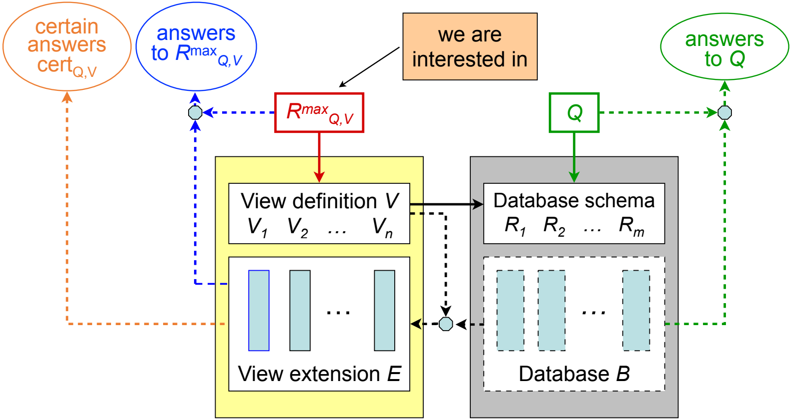
- view-based query answering
- view-based query rewriting

View-based query answering (QA)



Open world assumption (sound views): $\mathcal{E} \subseteq \mathcal{V}(\mathcal{B})$

View-based query rewriting (QR)



Open world assumption (sound views): $\mathcal{E} \subseteq \mathcal{V}(\mathcal{B})$

$R^{max}_{Q,\mathcal{V}}$ expressed in the "same" language as Q (but on $\mathcal{V}$-symbols)

View-based query processing before 2000

- VBQP studied in the DB theory community mostly for the case of conjunctive queries (i.e., select-project-join SQL queries) and variants.
- Confusion between (view-based) QA and QR:
  - For CQs, QA and QR coincide (i.e., $R^{max}_{Q,\mathcal{V}}$ computes $cert_{Q,\mathcal{V}}$).
  - However, they **do not coincide in general**.
- Need to better understand the relationship between, the query, the rewriting, and the certain answers.

View-based query processing after 2000

Inspired by the first nice result on rewriting of RPQs, the *Four Italians* started to look into VBQP for graph databases.

- Richer setting than CQs, in which we have a more fine-grained distinction between different interesting notions.
- Nice playground for sophisticated automata-theoretic techniques.
- Space for the application of a further powerful tool, namely CSP.

View-based query processing after 2000

Inspired by the first nice result on rewriting of RPQs, the *Four Italians* started to look into VBQP for graph databases.

- Richer setting than CQs, in which we have a more fine-grained distinction between different interesting notions.
- Nice playground for sophisticated automata-theoretic techniques.
- Space for the application of a further powerful tool, namely CSP.

This led to a fruitful line of research and a long-standing collaboration.

VBQP for graph databases

- Graph DB is a directed graph with edge labels in an alphabet Σ (basic binary relations).
- Queries and views are variants of RPQs (i.e., RPQs, 2RPQs, CRPQs, C2RPQs):
  - an RPQ is a regular expression (or an automaton) over the edge labels
  - in RPQs, edges are traversed only forward (r), and in 2RPQs also backward (r^-)
  - the result of a query Q is the set of pairs of nodes connected by a path in $\mathcal{L}(Q)$
  - C(2)RPQs are as CQs, but with (2)RPQs instead of predicates

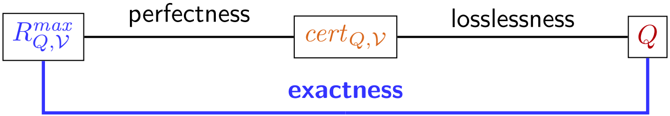
VBQP for graph databases

- Graph DB is a directed graph with edge labels in an alphabet Σ (basic binary relations).
- Queries and views are variants of RPQs (i.e., RPQs, 2RPQs, CRPQs, C2RPQs):
  - an RPQ is a regular expression (or an automaton) over the edge labels
  - in RPQs, edges are traversed only forward (r), and in 2RPQs also backward (r^-)
  - the result of a query Q is the set of pairs of nodes connected by a path in $\mathcal{L}(Q)$
  - C(2)RPQs are as CQs, but with (2)RPQs instead of predicates

In this setting, we were interested in better understanding the relationships between:

- the maximally contained rewriting $R^{max}_{Q,\mathcal{V}}$
- the certain answers $cert_{Q,\mathcal{V}}$ (viewed as a query)
- the original query Q

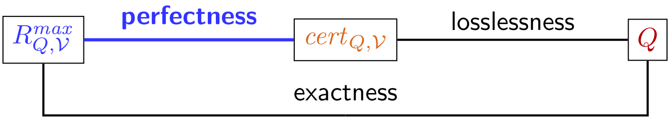
Exactness: comparing $R^{max}_{Q,\mathcal{V}}$ and Q



The maximal rewriting $R^{max}_{Q,\mathcal{V}}$ of Q wrt views $\mathcal{V}$ is **exact** if
for every database $\mathcal{B}$ we have that $Q(\mathcal{B}) = R^{max}_{Q,\mathcal{V}}(\mathcal{V}(\mathcal{B}))$.

Exactness means losslessness of the rewriting wrt the query.
(Note that exactness = perfectness + losslessness.)

Perfectness: comparing $R^{max}_{Q,\mathcal{V}}$ and $cert_{Q,\mathcal{V}}$

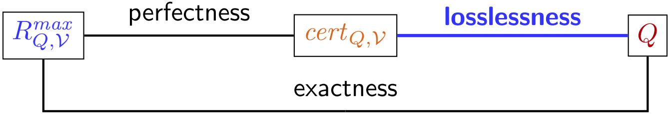


The maximal rewriting $R^{max}_{Q,\mathcal{V}}$ of Q wrt views $\mathcal{V}$ is **perfect** if
for every database $\mathcal{B}$ and every view extension $\mathcal{E}$ with $\mathcal{E} \subseteq \mathcal{V}(\mathcal{B})$ we have $cert_{Q,\mathcal{V}}(\mathcal{E}) = R^{max}_{Q,\mathcal{V}}(\mathcal{E})$.

Perfectness means that the maximal rewriting is powerful enough to compute the certain answers.

Perfectness allows us to compute $cert_{Q,\mathcal{V}}$ by evaluating $R^{max}_{Q,\mathcal{V}}$ over the view extension.

Losslessness: comparing $cert_{Q,\mathcal{V}}$ and Q



A set of views $\mathcal{V}$ is **lossless** wrt a query Q, if
for every database $\mathcal{B}$ we have that $Q(\mathcal{B}) = cert_{Q,\mathcal{V}}(\mathcal{V}(\mathcal{B}))$.

Losslessness means that the views are powerful enough to precisely answer the query.

Losslessness means that if we had access to the database, we could compute $cert_{Q,\mathcal{V}}$ by evaluating Q over the database.

The role of automata for VBQP in graph databases

In our work, we have developed and relied on different automata-theoretic characterizations:

- QR for RPQs
- QA for RPQs under various assumptions (closed vs. open domain, sound vs. exact views) via ad-hoc automata constructions.
- QA for 2RPQs via two-way automata
- QR for 2RPQs

The role of automata for VBQP in graph databases

In our work, we have developed and relied on different automata-theoretic characterizations:

- QR for RPQs
- QA for RPQs under various assumptions (closed vs. open domain, sound vs. exact views) via ad-hoc automata constructions.
- QA for 2RPQs via two-way automata
- QR for 2RPQs

In (almost) all cases we obtained instances of

"*Moshe's Automata-theoretic Meta-theorem*"

By using automata (and not doing anything stupid)
you get optimal complexity result.

VBQP in graph databases and CSP

Many of our results rely on a **characterization of QA for (2)RPQs via non-uniform CSP**.

- We associate to the query Q and view definitions $\mathcal{V}$ the constraints template $CT_{Q,\mathcal{V}}$:
  - structure over the alphabet $\mathcal{V} \cup \{U_i, U_f\}$ (for RPQs);
  - keeps track how the states of the NFA for Q change when following in the DB path according to the views.
- We associate to the view extension $\mathcal{E}$ and two objects c, d the constraint instance $\mathcal{E}^{c,d}$, which is also a structure over $\mathcal{V} \cup \{U_i, U_f\}$.

Characterization of QA via non-uniform CSP

$(c, d) \notin cert_{Q,\mathcal{V}}$ iff there is a homomorphism from $\mathcal{E}^{c,d}$ to $CT_{Q,\mathcal{V}}$

We have exploited this characterization also for various problems related to VBQP for RPQs:

- QA, QR
- losslessness
- perfectness
- view-based query containment

This fruitful research over many years resulted in almost 30 papers with collectively almost 1500 citations (and 3 papers still contributing to Moshe's h-index).

Thanks Moshe for making this possible!