View-Based Query Processing

Paolo Guagliardo

KRDB Research Centre
Free University of Bozen-Bolzano, Italy

11 November 2010

Freie Universität Bozen
Libera Università di Bolzano
Free University of Bozen - Bolzano

KRDB Research Centre
Knowledge Representation meets DataBases

Outline

Outline

View-based query processing

Definition and motivation

> The problem of computing the answer to a query based on a set of views

Application areas:

- **query optimisation**

  find a “better” query providing the same answer
- **data warehousing**

  select which views to materialise
- **data integration**

  check whether relevant queries can be answered using only a given set of sources
- **security and privacy**

  ensure that the views do not provide enough information to answer the sensitive queries

Preliminaries and notation

A **relational structure** $\mathcal{I}$ over a finite alphabet of relation symbols is a pair $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where

- $\Delta^{\mathcal{I}}$ is a domain of objects
- $\cdot^{\mathcal{I}}$ is a function associating each relation symbol r with a set of k-tuples of objects, with k the arity of r

A **query** is a function from relational structures to sets of tuples of a certain arity

Preliminaries and notation

A **relational structure** $\mathcal{I}$ over a finite alphabet of relation symbols is a pair $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where

- $\Delta^{\mathcal{I}}$ is a domain of objects
- $\cdot^{\mathcal{I}}$ is a function associating each relation symbol r with a set of k-tuples of objects, with k the arity of r

A **query** is a function from relational structures to sets of tuples of a certain arity

$\mathcal{R}$	finite alphabet of symbols (**database signature**)
$\mathbf{D}$	a relational structure over $\mathcal{R}$ (**database**)
$\mathcal{V}$	finite set of **view symbols** not in $\mathcal{R}$
$V^{\mathcal{R}}$	formula **defining** $V \in \mathcal{V}$ (in terms of the symbols in $\mathcal{R}$)
$\mathbf{E}$	a relational structure over $\mathcal{V}$, called a **$\mathcal{V}$-extension**

Semistructured data

Semistructured database

- a finite directed graph
  with edges labelled by elements of a finite alphabet $\mathcal{R}$
- represented as a finite relational structure $\mathbf{D}$
  over the set $\mathcal{R}$ of binary relation symbols

Semistructured data

Semistructured database

- a finite directed graph
  with edges labelled by elements of a finite alphabet $\mathcal{R}$
- represented as a finite relational structure $\mathbf{D}$
  over the set $\mathcal{R}$ of binary relation symbols

Regular Path Query (RPQ)

- a binary query defined in terms of a regular language over $\mathcal{R}$
- the answer $Q(\mathbf{D})$ to an RPQ Q over a database $\mathbf{D}$
  is the set of pairs of objects connected in $\mathbf{D}$
  by a sequence of (directed) edges forming a word in $\mathcal{L}(Q)$

Semistructured data

Semistructured database

- a finite directed graph
  with edges labelled by elements of a finite alphabet $\mathcal{R}$
- represented as a finite relational structure $\mathbf{D}$
  over the set $\mathcal{R}$ of binary relation symbols

Regular Path Query (RPQ)

- a binary query defined in terms of a regular language over $\mathcal{R}$
- the answer $Q(\mathbf{D})$ to an RPQ Q over a database $\mathbf{D}$
  is the set of pairs of objects connected in $\mathbf{D}$
  by a sequence of (directed) edges forming a word in $\mathcal{L}(Q)$

Two-way Regular Path Query (2RPQ)

- an RPQ extended with backward navigation of database edges

Answering, rewriting and losslessness

Two main approaches to view-based query processing:

- Query answering

  Compute the tuples satisfying the query in all databases consistent with the views (**certain answers**)

- Query rewriting

  Reformulate the query in terms of the views,
  then evaluate the rewriting over the view extensions

Answering, rewriting and losslessness

Two main approaches to view-based query processing:

- Query answering

  Compute the tuples satisfying the query in all databases consistent with the views (**certain answers**)

- Query rewriting

  Reformulate the query in terms of the views,
  then evaluate the rewriting over the view extensions

Related issue:

- Losslessness

  Determine whether there is information loss

  - w.r.t. query answering
  - w.r.t. query rewriting

Outline

Assumptions on the views

Let $\mathbf{D}$ be a database and let $\mathcal{V}^{\mathcal{R}}(\mathbf{D})$ be the $\mathcal{V}$-extension $\mathbf{E}$ such that $V(\mathbf{E}) = V^{\mathcal{R}}(\mathbf{D})$ for each $V \in \mathcal{V}$

A $\mathcal{V}$-extension $\mathbf{E}$ is

- **sound** w.r.t. $\mathbf{D}$ iff $\mathbf{E} \subseteq \mathcal{V}^{\mathcal{R}}(\mathbf{D})$
- **exact** w.r.t. $\mathbf{D}$ iff $\mathbf{E} = \mathcal{V}^{\mathcal{R}}(\mathbf{D})$

Certain answers

The certain answers to Q under sound views $\mathcal{V}$ w.r.t. a $\mathcal{V}$-extension $\mathbf{E}$ is the set of all tuples t such that $t \in Q(\mathbf{D})$ for every database $\mathbf{D}$ w.r.t. which $\mathbf{E}$ is sound

$$\mathrm{cert}^{\text{sound}}_{Q,\mathcal{V}}(\mathbf{E}) = \bigcap \left\{ Q(\mathbf{D}) \mid \text{for all } \mathbf{D} \text{ s.t. } \mathbf{E} \subseteq \mathcal{V}^{\mathcal{R}}(\mathbf{D}) \right\}$$

The certain answers to Q under exact views $\mathcal{V}$ w.r.t. a $\mathcal{V}$-extension $\mathbf{E}$ is the set of all tuples t such that $t \in Q(\mathbf{D})$ for every database $\mathbf{D}$ for which $\mathbf{E} = \mathcal{V}^{\mathcal{R}}(\mathbf{D})$

$$\mathrm{cert}^{\text{exact}}_{Q,\mathcal{V}}(\mathbf{E}) = \bigcap \left\{ Q(\mathbf{D}) \mid \text{for all } \mathbf{D} \text{ s.t. } \mathbf{E} = \mathcal{V}^{\mathcal{R}}(\mathbf{D}) \right\}$$

Rewriting

Definition

Q_{rw} is a rewriting of a query Q under sound views $\mathcal{V}$ iff
for every database $\mathbf{D}$ and every $\mathcal{V}$-extension $\mathbf{E}$ s.t. $\mathbf{E} \subseteq \mathcal{V}^{\mathcal{R}}(\mathbf{D})$

$$Q_{\mathsf{rw}}(\mathbf{E}) \subseteq Q(\mathbf{D})$$

Definition

Q_{rw} is a rewriting of a query Q under exact views $\mathcal{V}$ iff
for every database $\mathbf{D}$

$$Q_{\mathsf{rw}}\big(\mathcal{V}^{\mathcal{R}}(\mathbf{D})\big) \subseteq Q(\mathbf{D})$$

Rewriting

Definition

Q_{rw} is a rewriting of a query Q under sound views $\mathcal{V}$ iff
for every database $\mathbf{D}$ and every $\mathcal{V}$-extension $\mathbf{E}$ s.t. $\mathbf{E} \subseteq \mathcal{V}^{\mathcal{R}}(\mathbf{D})$

$$Q_{\text{rw}}(\mathbf{E}) \subseteq Q(\mathbf{D})$$

Definition

Q_{rw} is a rewriting of a query Q under exact views $\mathcal{V}$ iff
for every database $\mathbf{D}$

$$Q_{\text{rw}}\big(\mathcal{V}^{\mathcal{R}}(\mathbf{D})\big) \subseteq Q(\mathbf{D})$$

The above definitions are equivalent
when the language used for expressing the rewritings is monotonic

Rewriting

Definition

Q_{rw} is a rewriting of a query Q under sound views $\mathcal{V}$ iff
for every database $\mathbf{D}$ and every $\mathcal{V}$-extension $\mathbf{E}$ s.t. $\mathbf{E} \subseteq \mathcal{V}^{\mathcal{R}}(\mathbf{D})$

$$Q_{\mathsf{rw}}(\mathbf{E}) \subseteq Q(\mathbf{D})$$

Definition

Q_{rw} is a rewriting of a query Q under exact views $\mathcal{V}$ iff
for every database $\mathbf{D}$

$$Q_{\mathsf{rw}}\big(\mathcal{V}^{\mathcal{R}}(\mathbf{D})\big) \subseteq Q(\mathbf{D})$$

The above definitions are equivalent
when the language used for expressing the rewritings is monotonic

Exact rewriting if the subset inclusion is an equality

Maximally contained rewritings

Let $\mathcal{L}_{\mathsf{r}}$ be a query class in which rewritings are expressed

Definition

A rewriting $Q_{\mathsf{rw}} \in \mathcal{L}_{\mathsf{r}}$ of Q under sound views $\mathcal{V}$ is $\mathcal{L}_{\mathsf{r}}$-maximal iff every other rewriting $Q'_{\mathsf{rw}} \in \mathcal{L}_{\mathsf{r}}$ of Q is s.t. for each database $\mathbf{D}$ and each $\mathcal{V}$-extension $\mathbf{E} \subseteq \mathcal{V}^{\mathcal{R}}(\mathbf{D})$

$$Q_{\mathsf{rw}}(\mathbf{E}) \not\subset Q'_{\mathsf{rw}}(\mathbf{E})$$

Definition

A rewriting $Q_{\mathsf{rw}} \in \mathcal{L}_{\mathsf{r}}$ of Q under exact views $\mathcal{V}$ is $\mathcal{L}_{\mathsf{r}}$-maximal iff every other rewriting $Q'_{\mathsf{rw}} \in \mathcal{L}_{\mathsf{r}}$ of Q is s.t. for each database $\mathbf{D}$

$$Q_{\mathsf{rw}}\big(\mathcal{V}^{\mathcal{R}}(\mathbf{D})\big) \not\subset Q'_{\mathsf{rw}}\big(\mathcal{V}^{\mathcal{R}}(\mathbf{D})\big)$$

Outline

Rewriting of regular expressions [Calvanese et al., 2002]

Problem

Given a regular expression E_0
and a finite set $\mathcal{E} = \{E_1, \ldots, E_k\}$ of regular expressions,
re-express (if possible) E_0 in terms of $E_1, \ldots, E_k$

Rewriting of regular expressions [Calvanese et al., 2002]

Problem

Given a regular expression E_0
and a finite set $\mathcal{E} = \{E_1, \ldots, E_k\}$ of regular expressions,
re-express (if possible) E_0 in terms of $E_1, \ldots, E_k$

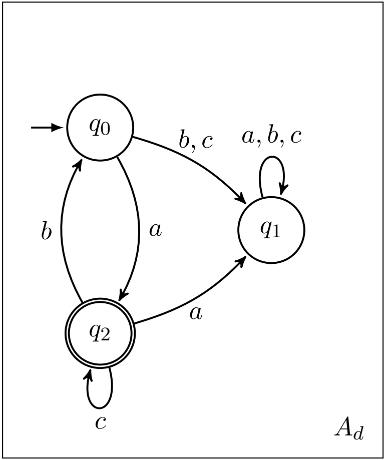
Solution

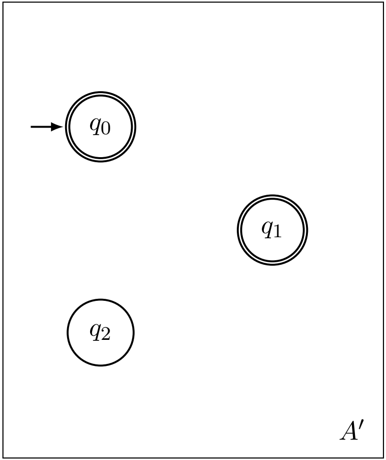
1. Construct a deterministic automaton A_d accepting $\mathcal{L}(E_0)$
2. Construct an automaton A' accepting exactly those words that are **not** in any rewriting of E_0
3. the complement of A' is the maximal rewriting of E_0 w.r.t. $\mathcal{E}$

Rewriting of regular expressions

next

Rewriting of $a \cdot (c + b \cdot a)^*$ in terms of $\{a,\ a \cdot c^* \cdot b,\ c\}$,
with $\mathrm{re}(e_1) = a$, $\mathrm{re}(e_2) = a \cdot c^* \cdot b$ and $\mathrm{re}(e_3) = c$

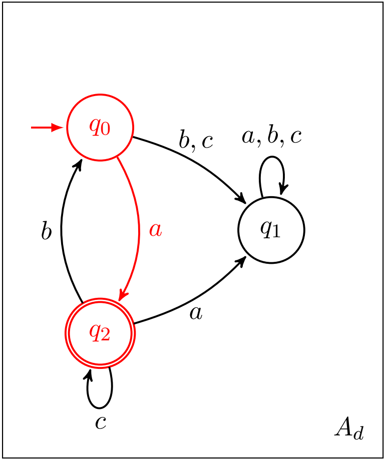


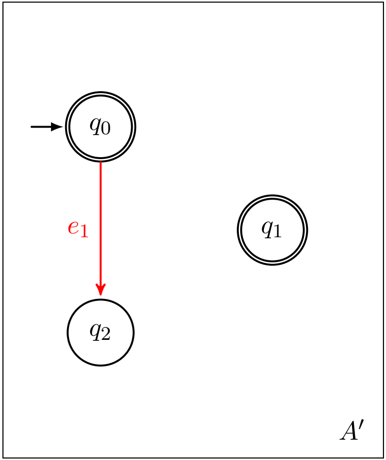


Rewriting of regular expressions

next

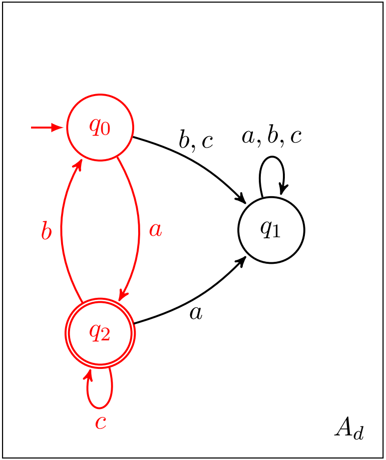
Rewriting of $a \cdot (c + b \cdot a)^*$ in terms of $\{a,\ a \cdot c^* \cdot b,\ c\}$,
with $\mathrm{re}(e_1) = a$, $\mathrm{re}(e_2) = a \cdot c^* \cdot b$ and $\mathrm{re}(e_3) = c$

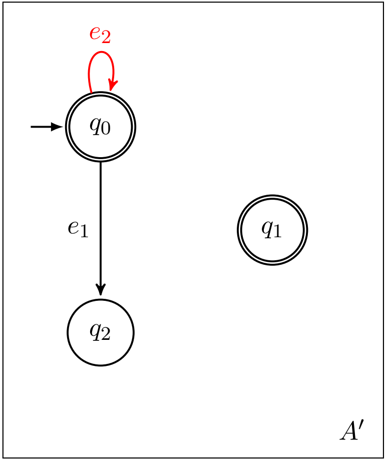




Rewriting of regular expressions

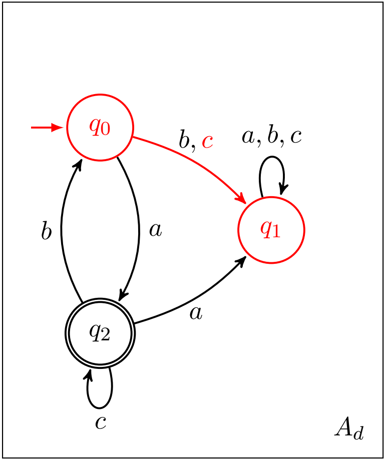
Rewriting of $a \cdot (c + b \cdot a)^*$ in terms of $\{a,\ a \cdot c^* \cdot b,\ c\}$,
with $\mathrm{re}(e_1) = a$, $\mathrm{re}(e_2) = a \cdot c^* \cdot b$ and $\mathrm{re}(e_3) = c$

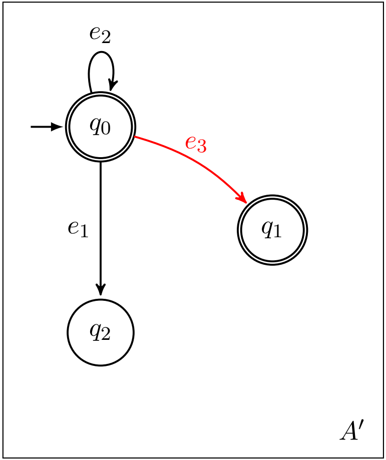




Rewriting of regular expressions

Rewriting of $a \cdot (c + b \cdot a)^*$ in terms of $\{a,\ a \cdot c^* \cdot b,\ c\}$,
with $\mathrm{re}(e_1) = a$, $\mathrm{re}(e_2) = a \cdot c^* \cdot b$ and $\mathrm{re}(e_3) = c$

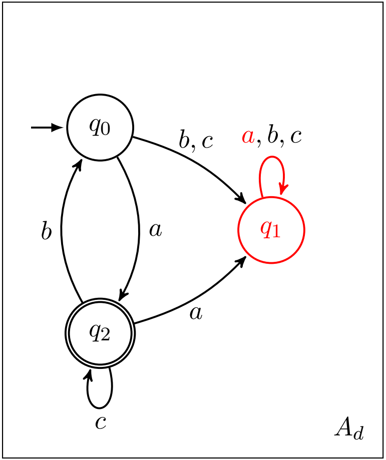


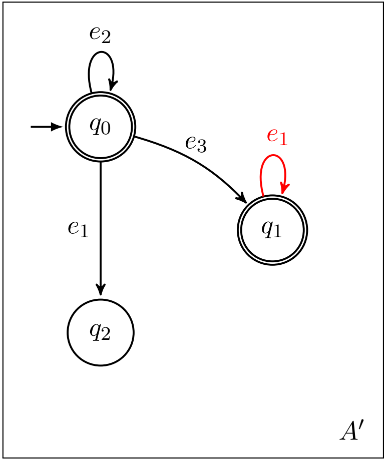


Rewriting of regular expressions

next

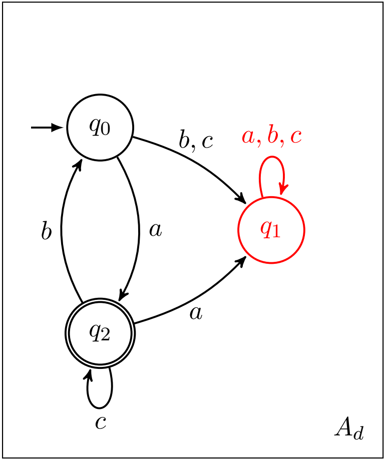
Rewriting of $a \cdot (c + b \cdot a)^*$ in terms of $\{a,\ a \cdot c^* \cdot b,\ c\}$,
with $\mathrm{re}(e_1) = a$, $\mathrm{re}(e_2) = a \cdot c^* \cdot b$ and $\mathrm{re}(e_3) = c$

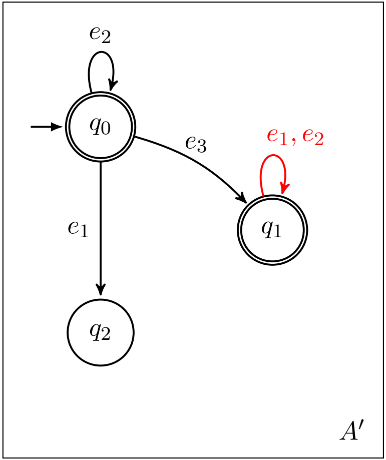




Rewriting of regular expressions

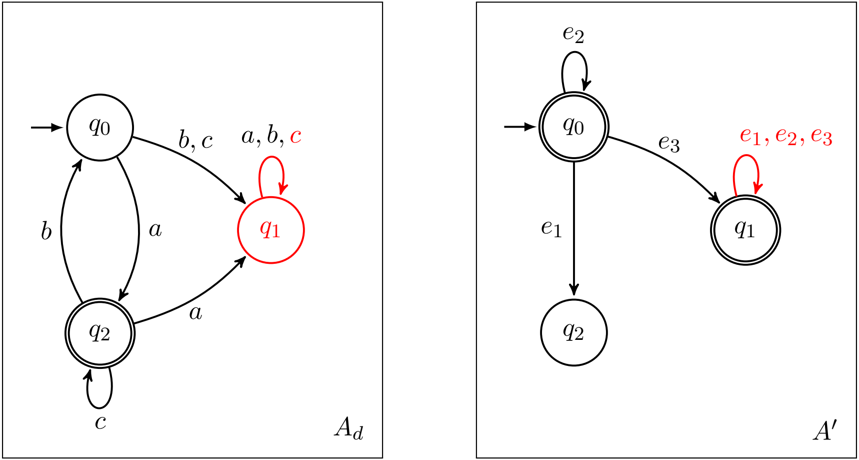
Rewriting of $a \cdot (c + b \cdot a)^*$ in terms of $\{a,\ a \cdot c^* \cdot b,\ c\}$,
with $\mathrm{re}(e_1) = a$, $\mathrm{re}(e_2) = a \cdot c^* \cdot b$ and $\mathrm{re}(e_3) = c$





Rewriting of regular expressions

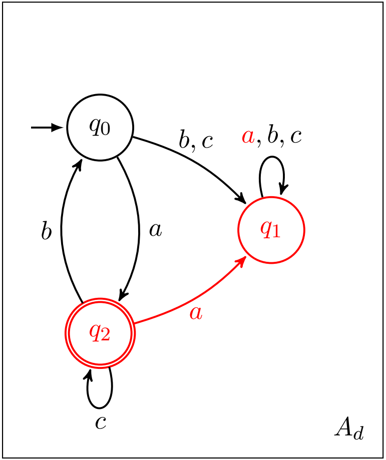
Rewriting of $a \cdot (c + b \cdot a)^*$ in terms of $\{a,\ a \cdot c^* \cdot b,\ c\}$,
with $\mathrm{re}(e_1) = a$, $\mathrm{re}(e_2) = a \cdot c^* \cdot b$ and $\mathrm{re}(e_3) = c$

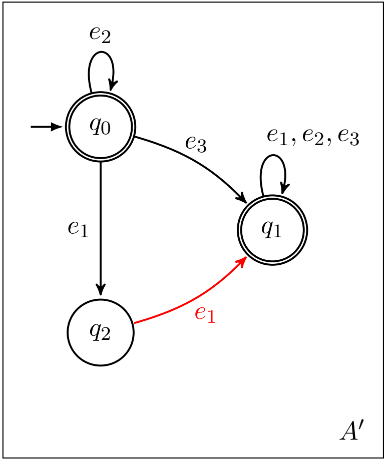


Rewriting of regular expressions

next

Rewriting of $a \cdot (c + b \cdot a)^*$ in terms of $\{a,\ a \cdot c^* \cdot b,\ c\}$,
with $\mathrm{re}(e_1) = a$, $\mathrm{re}(e_2) = a \cdot c^* \cdot b$ and $\mathrm{re}(e_3) = c$

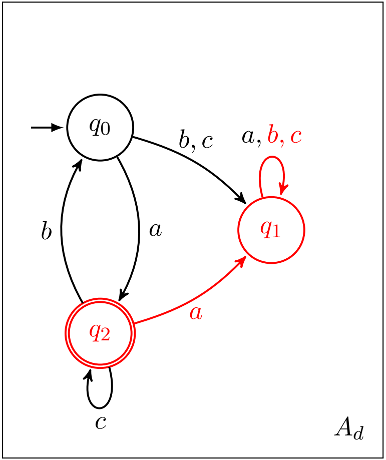


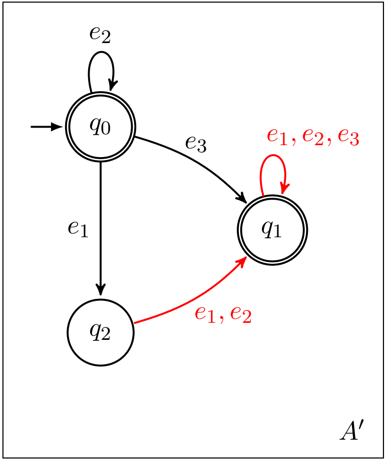


Rewriting of regular expressions

next

Rewriting of $a \cdot (c + b \cdot a)^*$ in terms of $\{a,\ a \cdot c^* \cdot b,\ c\}$, with $\mathrm{re}(e_1) = a$, $\mathrm{re}(e_2) = a \cdot c^* \cdot b$ and $\mathrm{re}(e_3) = c$

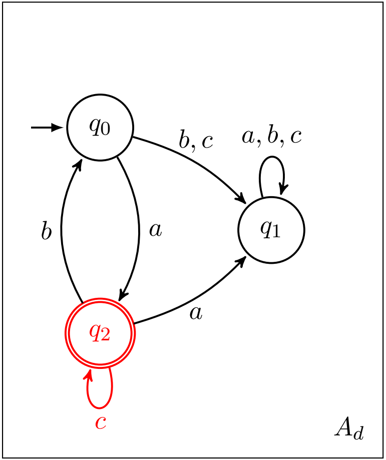


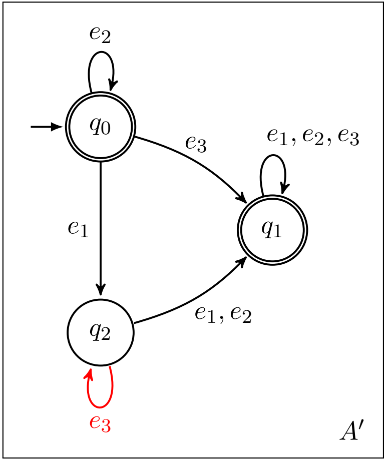


Rewriting of regular expressions

▸ next

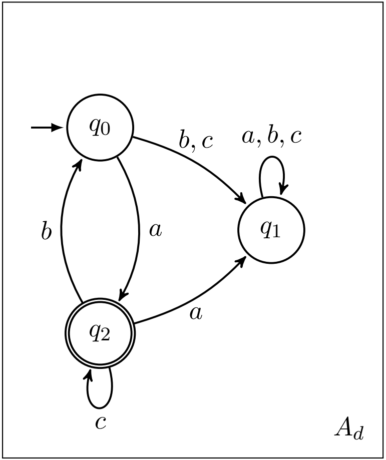
Rewriting of $a \cdot (c + b \cdot a)^*$ in terms of $\{a,\ a \cdot c^* \cdot b,\ c\}$,
with $\mathrm{re}(e_1) = a$, $\mathrm{re}(e_2) = a \cdot c^* \cdot b$ and $\mathrm{re}(e_3) = c$

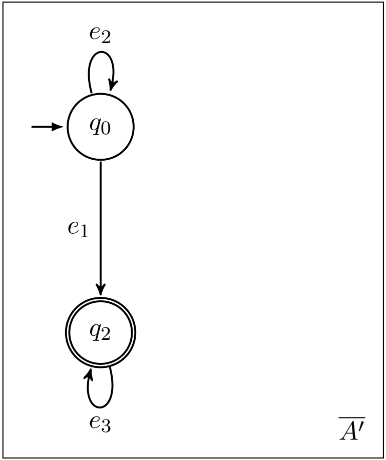




Rewriting of regular expressions

Rewriting of $a \cdot (c + b \cdot a)^*$ in terms of $\{a,\ a \cdot c^* \cdot b,\ c\}$,
with $\mathrm{re}(e_1) = a$, $\mathrm{re}(e_2) = a \cdot c^* \cdot b$ and $\mathrm{re}(e_3) = c$

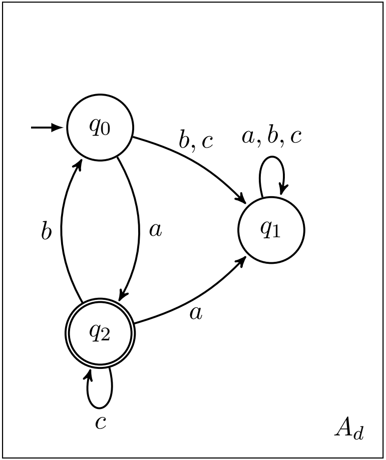


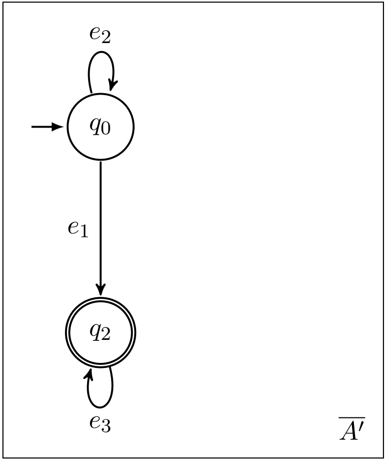


Rewriting of regular expressions

next

Rewriting of $a \cdot (c + b \cdot a)^*$ in terms of $\{a,\ a \cdot c^* \cdot b,\ c\}$, with $\mathrm{re}(e_1) = a$, $\mathrm{re}(e_2) = a \cdot c^* \cdot b$ and $\mathrm{re}(e_3) = c$





The rewriting is $R = e_2{}^* \cdot e_1 \cdot e_3{}^*$ and $\exp(R) = (a \cdot c^* \cdot b)^* \cdot a \cdot c^*$

back

Exactness of rewritings

Definition A rewriting R of E_0 is **exact** if $\mathcal{L}\big(\exp(R)\big) = \mathcal{L}(E_0)$

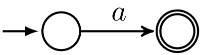
Rewriting of regular expressions

Exactness of rewritings

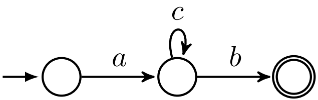
back

Definition A rewriting R of E_0 is **exact** if $\mathcal{L}\big(\exp(R)\big) = \mathcal{L}(E_0)$

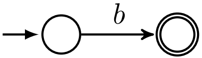
- A_1 for $\mathrm{re}(e_1) = a$

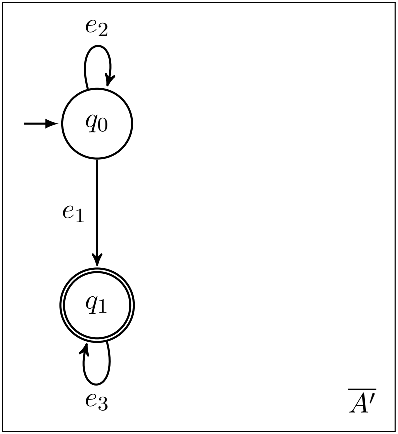


- A_2 for $\mathrm{re}(e_2) = a \cdot c^* \cdot b$



- A_3 for $\mathrm{re}(e_3) = b$





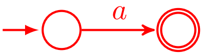
Rewriting of regular expressions

Exactness of rewritings

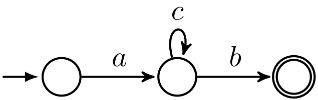
back

Definition A rewriting R of E_0 is **exact** if $\mathcal{L}\big(\exp(R)\big) = \mathcal{L}(E_0)$

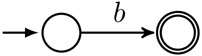
- A_1 for $\mathrm{re}(e_1) = a$

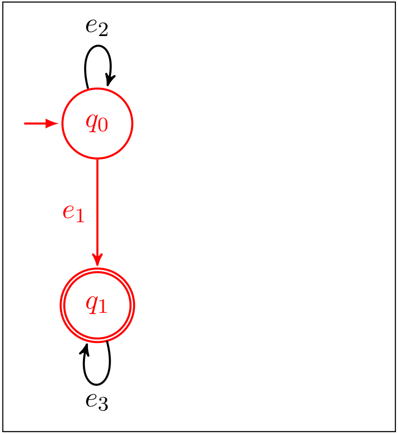


- A_2 for $\mathrm{re}(e_2) = a \cdot c^* \cdot b$



- A_3 for $\mathrm{re}(e_3) = b$



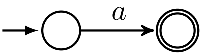


Rewriting of regular expressions

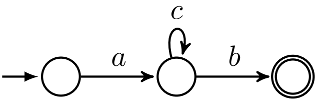
Exactness of rewritings

Definition A rewriting R of E_0 is **exact** if $\mathcal{L}\big(\exp(R)\big) = \mathcal{L}(E_0)$

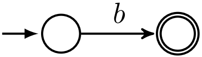
- A_1 for $\mathrm{re}(e_1) = a$

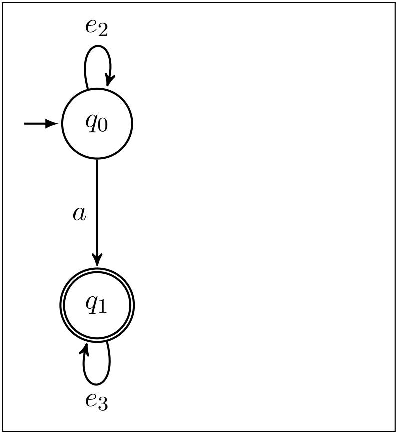


- A_2 for $\mathrm{re}(e_2) = a \cdot c^* \cdot b$



- A_3 for $\mathrm{re}(e_3) = b$





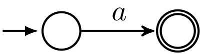
Rewriting of regular expressions

Exactness of rewritings

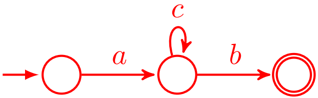
back

Definition A rewriting R of E_0 is **exact** if $\mathcal{L}\big(\exp(R)\big) = \mathcal{L}(E_0)$

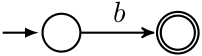
- A_1 for $\mathrm{re}(e_1) = a$

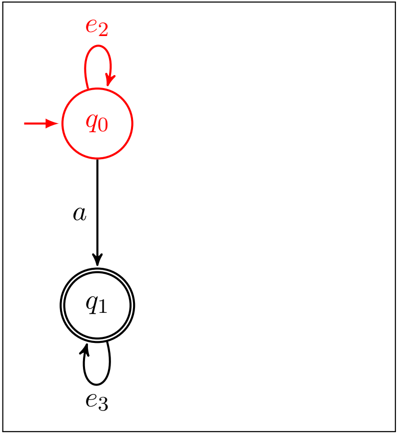


- A_2 for $\mathrm{re}(e_2) = a \cdot c^* \cdot b$



- A_3 for $\mathrm{re}(e_3) = b$



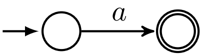


Rewriting of regular expressions

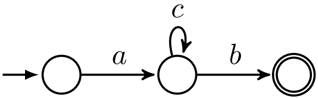
Exactness of rewritings

Definition A rewriting R of E_0 is **exact** if $\mathcal{L}\big(\exp(R)\big) = \mathcal{L}(E_0)$

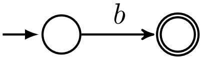
- A_1 for $\mathrm{re}(e_1) = a$

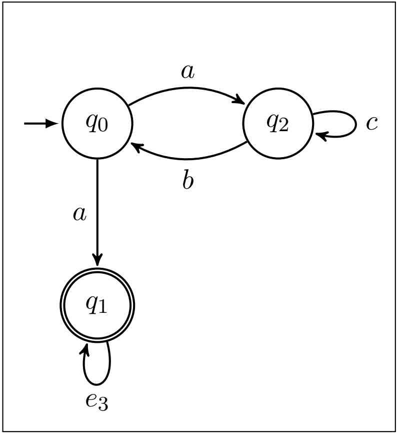


- A_2 for $\mathrm{re}(e_2) = a \cdot c^* \cdot b$



- A_3 for $\mathrm{re}(e_3) = b$





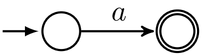
Rewriting of regular expressions

Exactness of rewritings

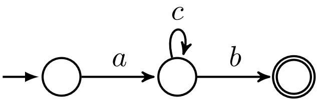
back

Definition A rewriting R of E_0 is **exact** if $\mathcal{L}(\exp(R)) = \mathcal{L}(E_0)$

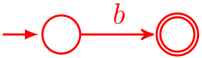
- A_1 for $\mathrm{re}(e_1) = a$

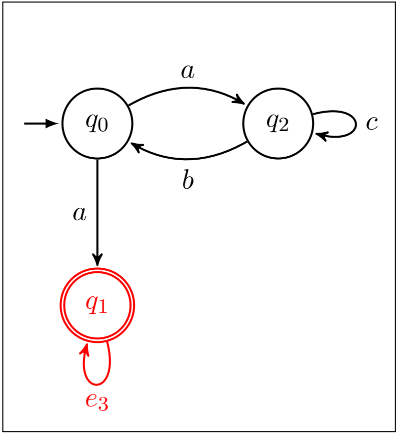


- A_2 for $\mathrm{re}(e_2) = a \cdot c^* \cdot b$



- A_3 for $\mathrm{re}(e_3) = b$



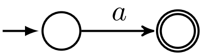


Rewriting of regular expressions

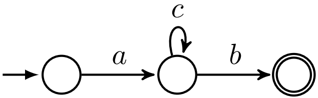
Exactness of rewritings

Definition A rewriting R of E_0 is **exact** if $\mathcal{L}\big(\exp(R)\big) = \mathcal{L}(E_0)$

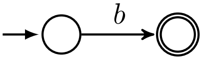
- A_1 for $\mathrm{re}(e_1) = a$

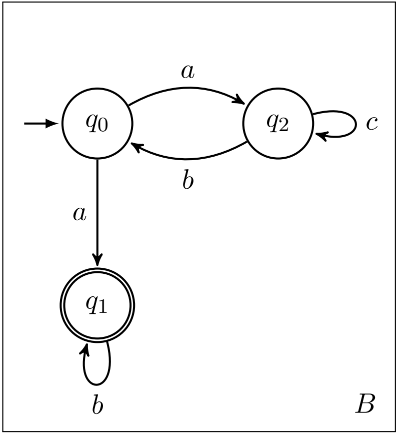


- A_2 for $\mathrm{re}(e_2) = a \cdot c^* \cdot b$



- A_3 for $\mathrm{re}(e_3) = b$





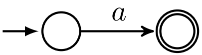
Rewriting of regular expressions

back

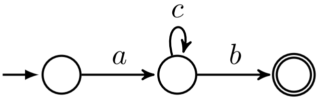
Exactness of rewritings

Definition A rewriting R of E_0 is **exact** if $\mathcal{L}\big(\exp(R)\big) = \mathcal{L}(E_0)$

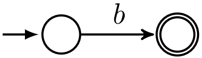
- A_1 for $\mathrm{re}(e_1) = a$

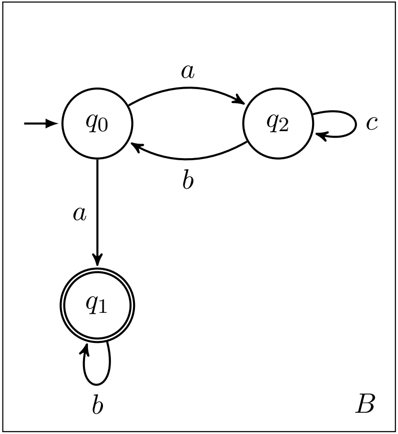


- A_2 for $\mathrm{re}(e_2) = a \cdot c^* \cdot b$



- A_3 for $\mathrm{re}(e_3) = b$





$\Rightarrow$ R is an exact rewriting of E_0 iff $\mathcal{L}(A_d \cap \overline{B}) = \varnothing$

Rewriting of regular expressions

Complexity results

Complexity of the proposed method:

- Generation of the maximal rewriting **2EXPTIME**
- Existence of an exact rewriting **2EXPSPACE**

Rewriting of regular expressions

Complexity results

Complexity of the proposed method:

- Generation of the maximal rewriting **2EXPTIME**
- Existence of an exact rewriting **2EXPSPACE**

Complexity of the decision problem:

- Existence of a nonempty rewriting **EXPSPACE-complete**
- Existence of an exact rewriting **2EXPSPACE-complete**

Rewriting of regular expressions

Complexity results

Complexity of the proposed method:

- Generation of the maximal rewriting **2EXPTIME**
- Existence of an exact rewriting **2EXPSPACE**

Complexity of the decision problem:

- Existence of a nonempty rewriting **EXPSPACE-complete**
- Existence of an exact rewriting **2EXPSPACE-complete**

$\Rightarrow$ The method is essentially optimal

Outline

Answering and rewriting

Most work focused on **conjunctive queries**
$\Rightarrow$ no distinction between answering and rewriting

Query rewriting $\equiv$ Query answering

The maximal rewriting computes the certain answers

Answering and rewriting

Most work focused on **conjunctive queries**
⇒ no distinction between answering and rewriting

Query rewriting ≡ Query answering

The maximal rewriting computes the certain answers

Difference between answering and rewriting can be pointed out in the context of RPQs in semistructured databases

[Calvanese et al., 2007]

Relationship between answering and rewriting

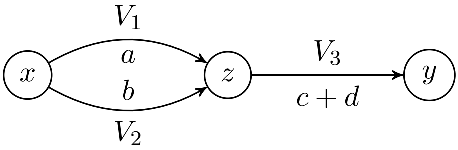
$Q = a \cdot c + b \cdot d$
$\mathcal{V} = \{V_1, V_2, V_3\}$, with $V_1^{\mathcal{R}} = a$, $V_2^{\mathcal{R}} = b$, $V_3^{\mathcal{R}} = c + d$

- The RPQ-maximal rewriting of Q is empty

Relationship between answering and rewriting

$Q = a \cdot c + b \cdot d$
$\mathcal{V} = \{V_1, V_2, V_3\}$, with $V_1^{\mathcal{R}} = a$, $V_2^{\mathcal{R}} = b$, $V_3^{\mathcal{R}} = c + d$

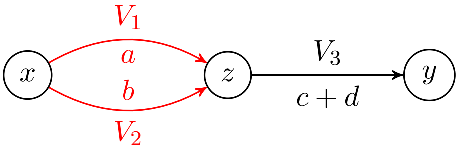
- The RPQ-maximal rewriting of Q is empty
- $\mathrm{cert}_{Q,\mathcal{V}} = \{(x, y) \mid \exists z. V_1(x, z) \wedge V_2(x, z) \wedge V_3(z, y)\}$



Relationship between answering and rewriting

$Q = a \cdot c + b \cdot d$
$\mathcal{V} = \{V_1, V_2, V_3\}$, with $V_1^{\mathcal{R}} = a$, $V_2^{\mathcal{R}} = b$, $V_3^{\mathcal{R}} = c + d$

- The RPQ-maximal rewriting of Q is empty
- $\mathrm{cert}_{Q,\mathcal{V}} = \{(x, y) \mid \exists z. V_1(x, z) \wedge V_2(x, z) \wedge V_3(z, y)\}$

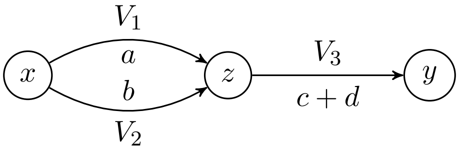


Rewriting ignores that V_1 and V_2 connect the same pair of objects, while answering takes it into account

Relationship between answering and rewriting

$Q = a \cdot c + b \cdot d$
$\mathcal{V} = \{V_1, V_2, V_3\}$, with $V_1^{\mathcal{R}} = a$, $V_2^{\mathcal{R}} = b$, $V_3^{\mathcal{R}} = c + d$

- The RPQ-maximal rewriting of Q is empty
- $\mathrm{cert}_{Q,\mathcal{V}} = \{(x, y) \mid \exists z. V_1(x, z) \wedge V_2(x, z) \wedge V_3(z, y)\}$



Rewriting ignores that V_1 and V_2 connect the same pair of objects, while answering takes it into account

$\Rightarrow$ Query answering is more precise than query rewriting (can match non-linear patterns in a database)

Outline

Losslessness w.r.t. query answering

Is the information content of a set of views sufficient to answer completely a given query?

Definition

A set of view $\mathcal{V}$ is said to be **lossless** w.r.t. a query Q, if for every database $\mathbf{D}$ we have

$$Q(\mathbf{D}) = \mathrm{cert}_{Q,\mathcal{V}}\big(\mathcal{V}^{\mathcal{R}}(\mathbf{D})\big)$$

Losslessness w.r.t. query rewriting

Definition (Exactness)

Q_{rw} is an **exact rewriting** of Q w.r.t. $\mathcal{V}$
if for every database $\mathbf{D}$ we have

$$Q(\mathbf{D}) = Q_{\mathsf{rw}}\big(\mathcal{V}^{\mathcal{R}}(\mathbf{D})\big)$$

Losslessness w.r.t. query rewriting

Definition (Exactness)

Q_{rw} is an **exact rewriting** of Q w.r.t. $\mathcal{V}$
if for every database $\mathbf{D}$ we have

$$Q(\mathbf{D}) = Q_{\mathsf{rw}}\big(\mathcal{V}^{\mathcal{R}}(\mathbf{D})\big)$$

Definition (Perfectness)

Q_{rw} is a **perfect rewriting** of Q w.r.t. $\mathcal{V}$
if for every database $\mathbf{D}$ we have

$$\mathrm{cert}_{Q,\mathcal{V}}\big(\mathcal{V}^{\mathcal{R}}(\mathbf{D})\big) = Q_{\mathsf{rw}}\big(\mathcal{V}^{\mathcal{R}}(\mathbf{D})\big)$$

Losslessness in the context of 2RPQs

Q a 2RPQ
$\mathcal{V}$ a set of 2RPQ views
$Q^{\mathsf{max}}_{\mathsf{rw}}$ the 2RPQ-maximal rewriting of Q w.r.t. $\mathcal{V}$

For every database $\mathbf{D}$,

$$Q^{\mathsf{max}}_{\mathsf{rw}}\big(\mathcal{V}^{\mathcal{R}}(\mathbf{D})\big) \quad \subseteq \quad \mathrm{cert}_{Q,\mathcal{V}}\big(\mathcal{V}^{\mathcal{R}}(\mathbf{D})\big) \quad \subseteq \quad Q(\mathbf{D})$$

- $Q^{\mathsf{max}}_{\mathsf{rw}}$ perfect

  $\Rightarrow$ no loss due to the rewriting

- $\mathcal{V}$ lossless w.r.t. Q

  $\Rightarrow$ no loss related to answering the query based on a set of views

- $Q^{\mathsf{max}}_{\mathsf{rw}}$ exact, that is, $Q^{\mathsf{max}}_{\mathsf{rw}}\big(\mathcal{V}^{\mathcal{R}}(\mathbf{D})\big) = Q(\mathbf{D})$

  $\Rightarrow$ perfectness of the rewriting + losslessness of the views

Losslessness in the context of 2RPQs

Q	a 2RPQ
$\mathcal{V}$	a set of 2RPQ views
$Q_{\text{rw}}^{\text{max}}$	the 2RPQ-maximal rewriting of Q w.r.t. $\mathcal{V}$

For every database $\mathbf{D}$,

$$Q_{\text{rw}}^{\text{max}}\big(\mathcal{V}^{\mathcal{R}}(\mathbf{D})\big) \quad = \quad \text{cert}_{Q,\mathcal{V}}\big(\mathcal{V}^{\mathcal{R}}(\mathbf{D})\big) \quad \subseteq \quad Q(\mathbf{D})$$

- $Q_{\text{rw}}^{\text{max}}$ perfect

  $\Rightarrow$ no loss due to the rewriting

- $\mathcal{V}$ lossless w.r.t. Q

  $\Rightarrow$ no loss related to answering the query based on a set of views

- $Q_{\text{rw}}^{\text{max}}$ exact, that is, $Q_{\text{rw}}^{\text{max}}\big(\mathcal{V}^{\mathcal{R}}(\mathbf{D})\big) = Q(\mathbf{D})$

  $\Rightarrow$ perfectness of the rewriting + losslessness of the views

Losslessness in the context of 2RPQs

Q a 2RPQ

$\mathcal{V}$ a set of 2RPQ views

$Q^{\mathsf{max}}_{\mathsf{rw}}$ the 2RPQ-maximal rewriting of Q w.r.t. $\mathcal{V}$

For every database $\mathbf{D}$,

$$Q^{\mathsf{max}}_{\mathsf{rw}}\big(\mathcal{V}^{\mathcal{R}}(\mathbf{D})\big) \quad \subseteq \quad \mathrm{cert}_{Q,\mathcal{V}}\big(\mathcal{V}^{\mathcal{R}}(\mathbf{D})\big) \quad = \quad Q(\mathbf{D})$$

- $Q^{\mathsf{max}}_{\mathsf{rw}}$ perfect

  $\Rightarrow$ no loss due to the rewriting

- $\mathcal{V}$ lossless w.r.t. Q

  $\Rightarrow$ no loss related to answering the query based on a set of views

- $Q^{\mathsf{max}}_{\mathsf{rw}}$ exact, that is, $Q^{\mathsf{max}}_{\mathsf{rw}}\big(\mathcal{V}^{\mathcal{R}}(\mathbf{D})\big) = Q(\mathbf{D})$

  $\Rightarrow$ perfectness of the rewriting + losslessness of the views

Losslessness in the context of 2RPQs

Q a 2RPQ
$\mathcal{V}$ a set of 2RPQ views
$Q_{\mathsf{rw}}^{\mathsf{max}}$ the 2RPQ-maximal rewriting of Q w.r.t. $\mathcal{V}$

For every database $\mathbf{D}$,

$$Q_{\mathsf{rw}}^{\mathsf{max}}\big(\mathcal{V}^{\mathcal{R}}(\mathbf{D})\big) \quad = \quad \mathrm{cert}_{Q,\mathcal{V}}\big(\mathcal{V}^{\mathcal{R}}(\mathbf{D})\big) \quad = \quad Q(\mathbf{D})$$

- $Q_{\mathsf{rw}}^{\mathsf{max}}$ perfect

  $\Rightarrow$ no loss due to the rewriting

- $\mathcal{V}$ lossless w.r.t. Q

  $\Rightarrow$ no loss related to answering the query based on a set of views

- $Q_{\mathsf{rw}}^{\mathsf{max}}$ exact, that is, $Q_{\mathsf{rw}}^{\mathsf{max}}\big(\mathcal{V}^{\mathcal{R}}(\mathbf{D})\big) = Q(\mathbf{D})$

  $\Rightarrow$ perfectness of the rewriting + losslessness of the views

Outline

Determinacy and rewriting

$\mathcal{V}$ **determines** Q (denoted $\mathcal{V} \twoheadrightarrow Q$) iff

$$\mathcal{V}^{\mathcal{R}}(\mathbf{D}_1) = \mathcal{V}^{\mathcal{R}}(\mathbf{D}_2) \text{ implies } Q(\mathbf{D}_1) = Q(\mathbf{D}_2)$$

Determinacy and rewriting

$\mathcal{V}$ **determines** Q (denoted $\mathcal{V} \twoheadrightarrow Q$) iff

$$\mathcal{V}^{\mathcal{R}}(\mathbf{D}_1) = \mathcal{V}^{\mathcal{R}}(\mathbf{D}_2) \text{ implies } Q(\mathbf{D}_1) = Q(\mathbf{D}_2)$$

If Q has an exact rewriting Q_{rw} under exact views $\mathcal{V}$, then $\mathcal{V} \twoheadrightarrow Q$

- converse in general does not hold

Determinacy and rewriting

$\mathcal{V}$ **determines** Q (denoted $\mathcal{V} \twoheadrightarrow Q$) iff

$$\mathcal{V}^{\mathcal{R}}(\mathbf{D}_1) = \mathcal{V}^{\mathcal{R}}(\mathbf{D}_2) \text{ implies } Q(\mathbf{D}_1) = Q(\mathbf{D}_2)$$

If Q has an exact rewriting Q_{rw} under exact views $\mathcal{V}$, then $\mathcal{V} \twoheadrightarrow Q$

- converse in general does not hold

$\mathcal{L}_{\mathsf{v}}$ view language (for defining views)
$\mathcal{L}_{\mathsf{q}}$ query language (for querying the database)
$\mathcal{L}_{\mathsf{r}}$ rewriting language (for expressing rewritings)

We say that $\mathcal{L}_{\mathsf{r}}$ is **complete** for $\mathcal{L}_{\mathsf{v}}$-to-$\mathcal{L}_{\mathsf{q}}$ rewritings iff $\mathcal{L}_{\mathsf{r}}$ can be used to rewrite $Q \in \mathcal{L}_{\mathsf{q}}$ using views $\mathcal{V}$ defined in $\mathcal{L}_{\mathsf{v}}$ whenever $\mathcal{V} \twoheadrightarrow Q$

Determinacy and rewriting [Nash et al., 2010]

Questions:

- For $Q \in \mathcal{L}_\mathsf{q}$ and views $\mathcal{V}$ with definitions in $\mathcal{L}_\mathsf{v}$, is it decidable whether $\mathcal{V} \twoheadrightarrow Q$?
- Is $\mathcal{L}_\mathsf{q}$ complete for $\mathcal{L}_\mathsf{v}$-to-$\mathcal{L}_\mathsf{q}$ rewritings?
  - If not, how to extend $\mathcal{L}_\mathsf{q}$ in order express such rewritings?

Two cases:

restricted	finite database instances only
unrestricted	possibly infinite databases

Query languages considered:

FO	first-order logic
CQ	conjunctive queries
UCQ	unions of conjunctive queries

Deciding determinacy

Theorem

If satisfiability in $\mathcal{L}_q$ or validity in $\mathcal{L}_v$ is undecidable, then it is also undecidable whether $\mathcal{V} \twoheadrightarrow Q$ where $Q \in \mathcal{L}_q$ and $\mathcal{V}$ is a set of views defined in $\mathcal{L}_v$

Deciding determinacy

Theorem

If satisfiability in $\mathcal{L}_q$ or validity in $\mathcal{L}_v$ is undecidable, then it is also undecidable whether $\mathcal{V} \twoheadrightarrow Q$ where $Q \in \mathcal{L}_q$ and $\mathcal{V}$ is a set of views defined in $\mathcal{L}_v$

Corollary

Determinacy is undecidable whenever queries or view definitions are expressed in FO

Outline

FO queries and views

⏩ skip

Determinacy: **undecidable** (corollary in previous slide)

In the **unrestricted** case, FO is **complete** for FO-to-FO rewritings

- does not hold with finite database instances

FO queries and views

skip

Determinacy: **undecidable** (corollary in previous slide)

In the **unrestricted** case, FO is **complete** for FO-to-FO rewritings

- does not hold with finite database instances

Theorem

Any language complete for FO-to-FO rewritings for finite instances must express all computable queries

UCQ queries and views

Determinacy: **undecidable**

Decidable whether a UCQ can be rewritten using a UCQ in terms of a set of views expressed as UCQs

$\Rightarrow$ UCQ is **not complete** for UCQ-to-UCQ rewritings (otherwise contradiction to undecidability of determinacy)

UCQ queries and views

Determinacy: **undecidable**

Decidable whether a UCQ can be rewritten using a UCQ
in terms of a set of views expressed as UCQs

$\Rightarrow$ UCQ is **not complete** for UCQ-to-UCQ rewritings
(otherwise contradiction to undecidability of determinacy)

Theorem

Any language complete for UCQ-to-CQ rewritings must express non-monotonic queries

- Proof in the next slide ⏩ skip

UCQ queries and views

Proof.

Let $\mathcal{R} = \{P, R\}$ with P, R unary, and let $\mathcal{V} = \{V_1, V_2\}$ with

$$V_1^{\mathcal{R}}(x) = \exists u \, . \, R(u) \land P(x) \quad ; \quad V_2^{\mathcal{R}}(x) = R(x) \lor P(x)$$

Let $Q(x) = P(x)$.

UCQ queries and views

Proof.

Let $\mathcal{R} = \{P, R\}$ with P, R unary, and let $\mathcal{V} = \{V_1, V_2\}$ with

$$V_1^{\mathcal{R}}(x) = \exists u \,.\, R(u) \wedge P(x) \quad ; \quad V_2^{\mathcal{R}}(x) = R(x) \vee P(x)$$

Let $Q(x) = P(x)$. Then, $\mathcal{V} \twoheadrightarrow Q$ because

- If $R(\mathbf{D}) \neq \varnothing$, then $Q(\mathbf{D}) = V_1(\mathbf{D})$
- If $R(\mathbf{D}) = \varnothing$, then $Q(\mathbf{D}) = V_2(\mathbf{D})$

UCQ queries and views

Proof.

Let $\mathcal{R} = \{P, R\}$ with P, R unary, and let $\mathcal{V} = \{V_1, V_2\}$ with

$$V_1^{\mathcal{R}}(x) = \exists u \,.\, R(u) \wedge P(x) \quad ; \quad V_2^{\mathcal{R}}(x) = R(x) \vee P(x)$$

Let $Q(x) = P(x)$. Then, $\mathcal{V} \twoheadrightarrow Q$ because

- If $R(\mathbf{D}) \neq \varnothing$, then $Q(\mathbf{D}) = V_1(\mathbf{D})$
- If $R(\mathbf{D}) = \varnothing$, then $Q(\mathbf{D}) = V_2(\mathbf{D})$

Let $\mathbf{D}_1$ such that $P(\mathbf{D}_1) = \{a, b\}$ and $R(\mathbf{D}_1) = \varnothing$

$\mathbf{D}_2$ such that $P(\mathbf{D}_2) = \{a\}$ and $R(\mathbf{D}_1) = \{b\}$

UCQ queries and views

Proof.

Let $\mathcal{R} = \{P, R\}$ with P, R unary, and let $\mathcal{V} = \{V_1, V_2\}$ with

$$V_1^{\mathcal{R}}(x) = \exists u \,.\, R(u) \wedge P(x) \quad ; \quad V_2^{\mathcal{R}}(x) = R(x) \vee P(x)$$

Let $Q(x) = P(x)$. Then, $\mathcal{V} \twoheadrightarrow Q$ because

- If $R(\mathbf{D}) \neq \varnothing$, then $Q(\mathbf{D}) = V_1(\mathbf{D})$
- If $R(\mathbf{D}) = \varnothing$, then $Q(\mathbf{D}) = V_2(\mathbf{D})$

Let $\mathbf{D}_1$ such that $P(\mathbf{D}_1) = \{a, b\}$ and $R(\mathbf{D}_1) = \varnothing$

$\mathbf{D}_2$ such that $P(\mathbf{D}_2) = \{a\}$ and $R(\mathbf{D}_1) = \{b\}$

$V_1(\mathbf{D}_1) = \varnothing \subseteq \{a\} = V_1(\mathbf{D}_2)$ and $V_2(\mathbf{D}_1) = \{a, b\} = V_2(\mathbf{D}_2)$
but $Q(\mathbf{D}_1) = \{a, b\} \not\subseteq \{a\} = Q(\mathbf{D}_2)$

UCQ queries and views

Proof.

Let $\mathcal{R} = \{P, R\}$ with P, R unary, and let $\mathcal{V} = \{V_1, V_2\}$ with

$$V_1^{\mathcal{R}}(x) = \exists u \;.\; R(u) \wedge P(x) \quad ; \quad V_2^{\mathcal{R}}(x) = R(x) \vee P(x)$$

Let $Q(x) = P(x)$. Then, $\mathcal{V} \twoheadrightarrow Q$ because

- If $R(\mathbf{D}) \neq \varnothing$, then $Q(\mathbf{D}) = V_1(\mathbf{D})$
- If $R(\mathbf{D}) = \varnothing$, then $Q(\mathbf{D}) = V_2(\mathbf{D})$

Let $\mathbf{D}_1$ such that $P(\mathbf{D}_1) = \{a, b\}$ and $R(\mathbf{D}_1) = \varnothing$

$\mathbf{D}_2$ such that $P(\mathbf{D}_2) = \{a\}$ and $R(\mathbf{D}_1) = \{b\}$

$V_1(\mathbf{D}_1) = \varnothing \subseteq \{a\} = V_1(\mathbf{D}_2)$ and $V_2(\mathbf{D}_1) = \{a, b\} = V_2(\mathbf{D}_2)$

but $Q(\mathbf{D}_1) = \{a, b\} \not\subseteq \{a\} = Q(\mathbf{D}_2)$

$\Rightarrow$ the mapping that associates each query extension $Q(\mathbf{D})$ to the corresponding extension $\mathcal{V}^{\mathcal{R}}(\mathbf{D})$ is non-monotonic $\square$

CQ queries and views

Determinacy: **open problem**

Decidable whether a CQ can be rewritten as a CQ
in terms of a set of views defined by means of CQs

Completeness of CQ for CQ-to-CQ rewritings
would imply decidability of determinacy

CQ queries and views

Determinacy: **open problem**

Decidable whether a CQ can be rewritten as a CQ
in terms of a set of views defined by means of CQs

Completeness of CQ for CQ-to-CQ rewritings
would imply decidability of determinacy

Unfortunately CQ is **not complete** for CQ-to-CQ rewritings

- a path query $P_n(x, y)$ on a binary relation R returns the pairs $\langle x, y \rangle$ for which there is an R-path of length n from x to y
- $\{P_3, P_4\} \twoheadrightarrow P_5$ because P_5 has the FO rewriting

$$P_5(x, y) \equiv \exists z \big[P_4(x, z) \wedge \forall v \big(P_3(v, z) \rightarrow P_4(v, y) \big) \big]$$

but P_5 has no CQ rewriting in terms of P_3 and P_4

Guarded fragment (GF)

Fragment of FOL consisting of only quantified formulas of the form

$$\forall\overline{x}\big(G(\overline{x},\overline{y}) \rightarrow \phi(\overline{x},\overline{y})\big)$$

where G is a relation symbol and $\phi(\overline{x},\overline{y})$ is guarded as well

Key restriction: all free variable occurring in $\phi(\overline{x},\overline{y})$
must also occur in the **guard** $G(\overline{x},\overline{y})$

Guarded fragment (GF)

Fragment of FOL consisting of only quantified formulas of the form

$$\forall \overline{x}\big(G(\overline{x}, \overline{y}) \to \phi(\overline{x}, \overline{y})\big)$$

where G is a relation symbol and $\phi(\overline{x}, \overline{y})$ is guarded as well

Key restriction: all free variable occurring in $\phi(\overline{x}, \overline{y})$
must also occur in the **guard** $G(\overline{x}, \overline{y})$

Not expressible in GF

- that a relation is transitive
- that a relation is a partial function

Guarded fragment (GF)

Fragment of FOL consisting of only quantified formulas of the form

$$\forall \overline{x}\big(G(\overline{x}, \overline{y}) \rightarrow \phi(\overline{x}, \overline{y})\big)$$

where G is a relation symbol and $\phi(\overline{x}, \overline{y})$ is guarded as well

Key restriction: all free variable occurring in $\phi(\overline{x}, \overline{y})$
must also occur in the **guard** $G(\overline{x}, \overline{y})$

Not expressible in GF

- that a relation is transitive
- that a relation is a partial function

⇒ Views defined in GF always consist of sub-tuples of a relation in the database

Packed fragment (PF)

Useful generalisation of the guarded fragment
allowing for **safe products** as guards

$$G(x_1, \ldots, x_n) = \bigwedge_{k=1,\ldots,m} \exists \overline{y}\ A_k(\overline{x}, \overline{y})$$

which for every pair of free variables x_i, x_j with $i \neq j$
has an atom A_k in which x_i, x_j both occur free

Packed fragment (PF)

Useful generalisation of the guarded fragment
allowing for **safe products** as guards

$$G(x_1, \ldots, x_n) = \bigwedge_{k=1,\ldots,m} \exists \overline{y}\ A_k(\overline{x}, \overline{y})$$

which for every pair of free variables x_i, x_j with $i \neq j$
has an atom A_k in which x_i, x_j both occur free

- transitivity and functionality not expressible in PF
- "until" operator of temporal logic in PF but not in GF

Packed fragment (PF)

Useful generalisation of the guarded fragment
allowing for **safe products** as guards

$$G(x_1, \ldots, x_n) = \bigwedge_{k=1,\ldots,m} \exists \overline{y}\ A_k(\overline{x}, \overline{y})$$

which for every pair of free variables x_i, x_j with $i \neq j$
has an atom A_k in which x_i, x_j both occur free

- transitivity and functionality not expressible in PF
- "until" operator of temporal logic in PF but not in GF

Properties (same as GF)

- validity problem is 2EXPTIME-complete
- every satisfiable formula is satisfiable on a finite model

PF queries and views [Marx, 2007]

No difference between unrestricted and finite case
(because of the finite model property)

Determinacy: **2EXPTIME-complete**

PF is **complete** for PF-to-PF rewritings

PF queries and views [Marx, 2007]

No difference between unrestricted and finite case
(because of the finite model property)

Determinacy: **2EXPTIME-complete**

PF is **complete** for PF-to-PF rewritings

Restriction to packed (U)CQs:

- PCQ is complete for PCQ-to-PCQ rewritings
- UPCQ is complete for UPCQ-to-UPCQ rewritings

Outline

Summary I

Two main approaches to view-based query processing:

- **answering** aims at finding the certain answers (answers to the query in all databases consistent with the views)
- **rewriting** aims at reformulating the query in terms of the views and then evaluating the rewritten query over the view extensions

Query rewriting is an approximation of query answering

Characterised when no loss of information occurs
w.r.t. rewriting and w.r.t. quality of views

Summary II

Given a set of views and a query expressed in a language $\mathcal{L}$

Determinacy

Decide whether the views determine the answer to the query

Completeness of rewritings

Can $\mathcal{L}$ be used for rewriting the query in terms of the views whenever the latter determine the answer to the former?

Language $\mathcal{L}$	Determinacy	Complete for $\mathcal{L}$-to-$\mathcal{L}$ rewritings?
FO	undecidable	YES (unrestricted) NO (finite)
UCQ	undecidable	NO
CQ	open	NO
PF	2EXPTIME-complete	YES

References

Calvanese, D., De Giacomo, G., Lenzerini, M., and Vardi, M. Y. (2002).
Rewriting of regular expressions and regular path queries.
Journal of Computer and System Sciences, 64(3):443 – 465.

Calvanese, D., De Giacomo, G., Lenzerini, M., and Vardi, M. Y. (2007).
View-based query processing: On the relationship between rewriting, answering and losslessness.
Theoretical Computer Science, 371(3):169 – 182.

Marx, M. (2007).
Queries determined by views: pack your views.
In *Proceedings of PODS '07*, pages 23–30, New York, NY, USA. ACM.

Nash, A., Segoufin, L., and Vianu, V. (2010).
Views and queries: Determinacy and rewriting.
ACM Trans. Database Syst., 35(3):1–41.