Processing Regular Path Queries Using Views

or

What Do We Need for Integrating Semistructured Data?

Diego Calvanese

University of Rome “La Sapienza”

joint work with *G. De Giacomo, M. Lenzerini, M.Y. Vardi*

Logic-based Methods for Information Integration

Vienna – August 23, 2003

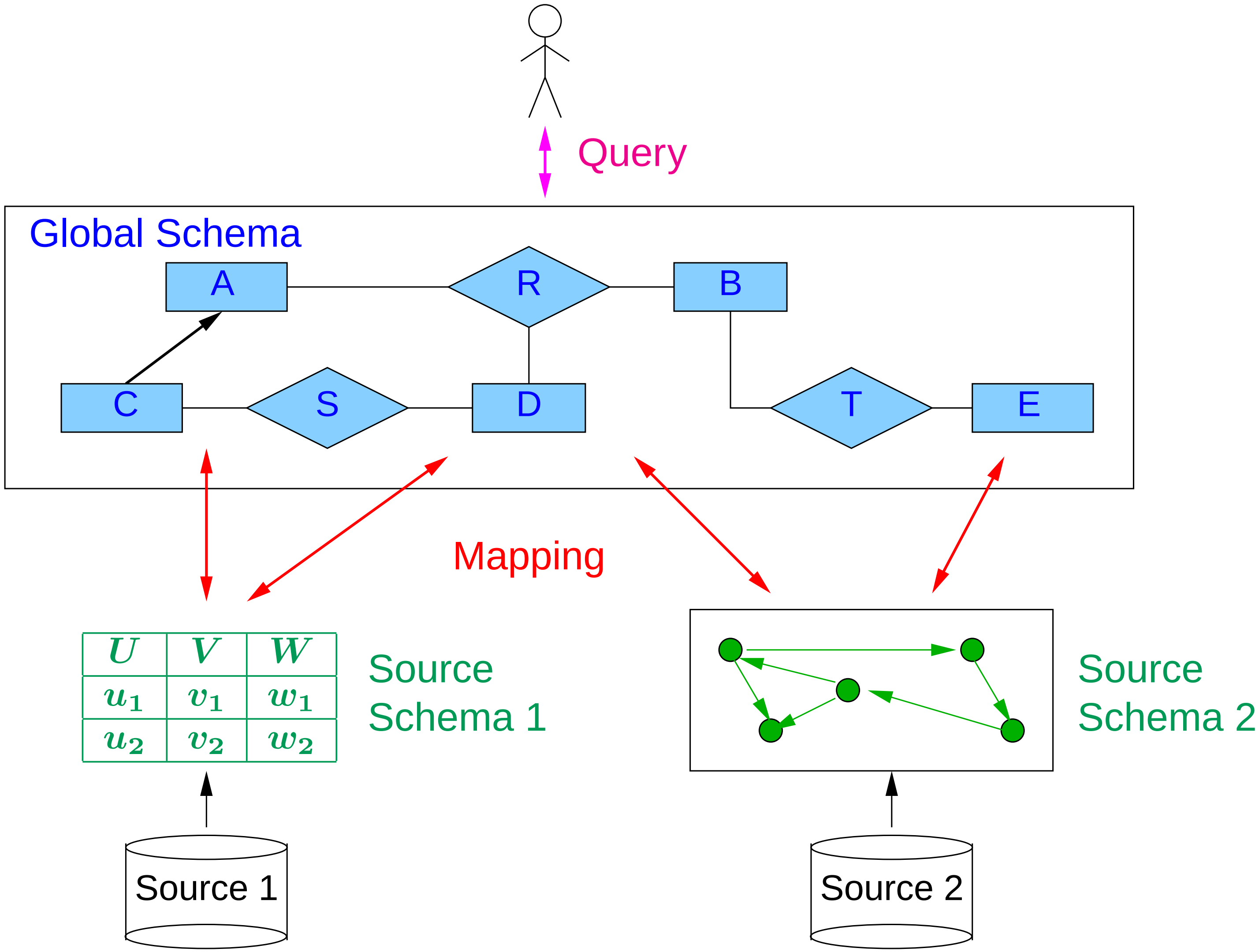
Data integration

Deals with the problem of providing a uniform access to a collection of data stored in multiple, autonomous, and heterogeneous data sources.

Basic problem in:

- management of distributed information systems
- data warehousing
- data re-engineering
- enterprise knowledge management
- querying multiple sources on the web
- e-commerce, e-business, e-government, e-$\cdots$
- integration of data from distributed scientific experiments
- $\cdots$

Framework for data integration



Quality in query answering

Among the various tasks in data integration, we deal with how to answer queries expressed on the global schema:

$\leadsto$ View-based query processing

Quality in query answering

Among the various tasks in data integration, we deal with how to answer queries expressed on the global schema:
$\leadsto$ View-based query processing

The data integration system should be designed in such a way that suitable quality criteria are met. Here, we concentrate on:

- Soundness: the answer to a query includes only what is known to be true
- Completeness: the answer to a query includes all that is known to be true

We aim at getting exactly what is known. But, what is known depends on how the data integration system is modeled

Formal framework

A data integration system $\mathcal{I}$ is a triple $\langle\mathcal{G}, \mathcal{S}, \mathcal{M}\rangle$, where

- $\mathcal{G}$ is the global schema
- $\mathcal{S}$ is the source schema
- $\mathcal{M}$ is the mapping between $\mathcal{G}$ and $\mathcal{S}$

Formal framework

A data integration system $\mathcal{I}$ is a triple $\langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$, where

- $\mathcal{G}$ is the global schema
- $\mathcal{S}$ is the source schema
- $\mathcal{M}$ is the mapping between $\mathcal{G}$ and $\mathcal{S}$

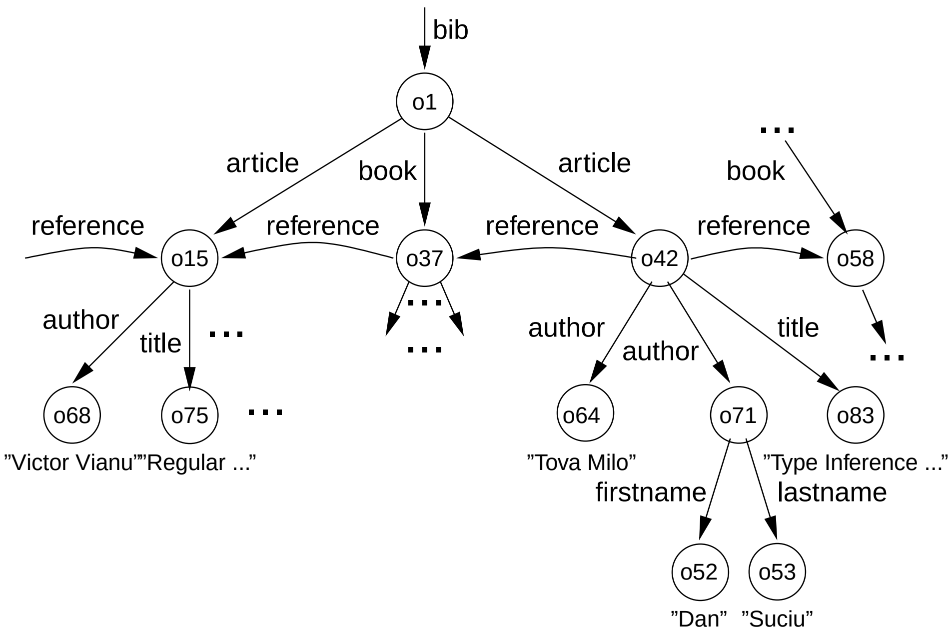
Semantics of $\mathcal{I}$: which are the (global) databases that satisfy $\mathcal{I}$?

- We start from a source database $\mathcal{D}$, representing the data at the sources
- The (global) databases $\mathcal{B}$ that satisfy $\mathcal{I}$ wrt $\mathcal{D}$ are those that:
  - are legal wrt the global schema $\mathcal{G}$, and
  - satisfy the mapping $\mathcal{M}$ wrt $\mathcal{D}$

Semistructured data

Semistructured data are an abstraction for data on the web, structured documents, XML:

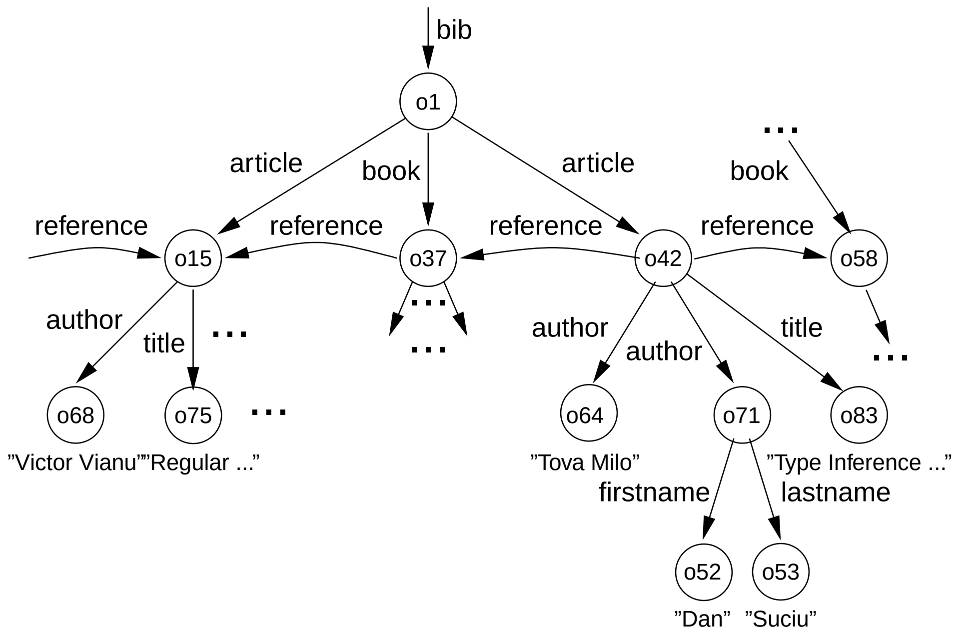
- A semistructured database is an edge-labeled graph



Semistructured data

Semistructured data are an abstraction for data on the web, structured documents, XML:

- A semistructured database is an edge-labeled graph
- Queries need to provide the ability to navigate the graph: regular path queries (RPQs) and 2-way regular-path-queries (2RPQs)



$$Q_1(x, y) \leftarrow$$
$$x \; ((\textbf{article} + \textbf{book})\cdot\textbf{ref}^*\cdot\textbf{title}) \; y$$

$$Q_2(x, y) \leftarrow$$
$$x \; (\textbf{article}\cdot(\textbf{ref} + \textbf{ref}^-)^*\cdot\textbf{title}) \; y$$

Integrating semistructured data

We consider data integration systems $\mathcal{I} = \langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$ where:

- The global schema $\mathcal{G}$ simply fixes the set of labels of the database
- The sources in $\mathcal{S}$ are binary relations
- The mapping $\mathcal{M}$ is of type local-as-view (LAV): to each source s it associates a 2RPQ view V_s over $\mathcal{G}$

Integrating semistructured data

We consider data integration systems $\mathcal{I} = \langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$ where:

- The global schema $\mathcal{G}$ simply fixes the set of labels of the database

  Example: $\mathcal{G} = \{\textbf{article}, \textbf{ref}, \textbf{title}, \textbf{author}, \ldots\}$

- The sources in $\mathcal{S}$ are binary relations

  Example: $\mathcal{S} = \{\mathsf{s}_1, \mathsf{s}_2, \mathsf{s}_3\}$, where

  - s_1 stores for each bibliography its articles
  - s_2 stores for each publ. the ones it references directly or indirectly
  - s_3 stores for each publication its title

- The mapping $\mathcal{M}$ is of type local-as-view (LAV): to each source s it associates a 2RPQ view V_s over $\mathcal{G}$

  Example:

$$
\begin{aligned}
V_{\mathsf{s}_1}(b, a) &\leftarrow b\ (\textbf{article})\ a \\
V_{\mathsf{s}_2}(p_1, p_2) &\leftarrow p_1\ (\textbf{ref}^*)\ p_2 \\
V_{\mathsf{s}_3}(p, t) &\leftarrow p\ (\textbf{title})\ t
\end{aligned}
$$

Assumptions on the sources

Let $\mathcal{D}$ be a source database and $\mathcal{B}$ a global database that satisfies $\mathcal{I}$ wrt $\mathcal{D}$:

sound source: $s(\mathcal{D}) \subseteq V_s(\mathcal{B})$

all tuples in the source satisfy V_s, but there may be other tuples satisfying V_s that are not in the source

complete source: $s(\mathcal{D}) \supseteq V_s(\mathcal{B})$

all tuples that satisfy V_s are in the source, but there may be also tuples in the source not satisfying V_s

exact source: $s(\mathcal{D}) = V_s(\mathcal{B})$

the tuples in the source are exactly those that satisfy V_s (i.e., both sound and complete)

We will assume that sources are sound (unless we explicitly say otherwise)

View-based query processing tasks

View-based query answering: compute the set of certain answers to a query over the global schema

$\leadsto$ is the basic basic query processing task

View-based query rewriting: reformulate a query over the global schema in terms of the sources

$\leadsto$ provides an indirect means for view-based query answering

Query containment and **view-based query containment**: check whether the answers to one query are contained in the answers to another query, possibly taking into account the views in the mapping

$\leadsto$ allow for establishing quality criteria of the answering process

View-based query answering

Given:

- a semistructured data integration system $\mathcal{I} = \langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$
- a source database $\mathcal{D}$
- a 2RPQ Q over $\mathcal{G}$
- a pair of objects (c, d)

check whether (c, d) is a certain answer to Q wrt $\mathcal{I}$ and $\mathcal{D}$

A certain answer is a tuple that is in the answer to Q for every database $\mathcal{B}$ that satisfies $\mathcal{I}$ wrt $\mathcal{D}$

View-based query answering is the basic query processing task in data integration [Levy+al '95, Rajaraman+al '95, Abiteboul+Duschka '98, — ICDE'00, — PODS'00, — LICS'00, . . .]

View-based query answering for 2RPQs

Technique based on search for a counterexample database:

1. it is sufficient to restrict the attention to counterexamples of a special form (canonical databases)
2. represent canonical databases by means of words
3. construct two-way finite automaton that accepts words encoding canonical counterexample databases
4. check for emptiness of the automaton

View-based query answering for 2RPQs

Technique based on search for a counterexample database:

1. it is sufficient to restrict the attention to counterexamples of a special form (canonical databases)
2. represent canonical databases by means of words
3. construct two-way finite automaton that accepts words encoding canonical counterexample databases
4. check for emptiness of the automaton

The non-emptiness of the automaton can be rephrased in terms of constraint satisfaction (CSP)

$\leadsto$ tight relationship between view-based query answering and CSP

Constraint satisfaction problems

Let $\mathcal{A}$ and $\mathcal{B}$ be relational structures over the same alphabet

A homomorphism h is a mapping from $\mathcal{A}$ to $\mathcal{B}$ such that for every relation R, if $(c_1, \ldots, c_n) \in R(\mathcal{A})$, then $(h(c_1), \ldots, h(c_n)) \in R(\mathcal{B})$.

Non-uniform constraint satisfaction problem $CSP(\mathcal{B})$: the set of relational structures $\mathcal{A}$ such that there is a homomorphism from $\mathcal{A}$ to $\mathcal{B}$.

Complexity:

- $CSP(\mathcal{B})$ is in NP
- there are structures $\mathcal{B}$ for which $CSP(\mathcal{B})$ is NP-hard

CSP and view-based query answering for 2RPQs

Consider $\mathcal{I} = \langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$ and a 2RPQ query Q over $\mathcal{G}$:

- We can define a relational structure $CT_{Q,\mathcal{M}}$, called constraint template of Q wrt $\mathcal{M}$
- Given a source database $\mathcal{D}$ and two objects c, d, we can define another relational structure $CI_{\mathcal{D}}^{c,d}$ over the same alphabet, called the constraint instance

$CT_{Q,\mathcal{M}}$ can be computed in exponential time in Q and polynomial time in $\mathcal{M}$

CSP and view-based query answering for 2RPQs

Consider $\mathcal{I} = \langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$ and a 2RPQ query Q over $\mathcal{G}$:

- We can define a relational structure $CT_{Q,\mathcal{M}}$, called constraint template of Q wrt $\mathcal{M}$
- Given a source database $\mathcal{D}$ and two objects c, d, we can define another relational structure $CI_{\mathcal{D}}^{c,d}$ over the same alphabet, called the constraint instance

$CT_{Q,\mathcal{M}}$ can be computed in exponential time in Q and polynomial time in $\mathcal{M}$

Theorem: (c, d) is not a certain answer to Q wrt $\mathcal{I}$ and $\mathcal{D}$

if and only if

there is a homomorphism from $CI_{\mathcal{D}}^{c,d}$ to $CT_{Q,\mathcal{M}}$, i.e, $CI_{\mathcal{D}}^{c,d} \in CSP(CT_{Q,\mathcal{M}})$

$\leadsto$ Characterization of view-based query answering for 2RPQs in terms of CSP

Complexity of view-based answering for RPQs and 2RPQs

From [ICDE'00] for RPQs and [PODS'00] for 2RPQs

Assumption on domain	Assumption on sources	Complexity		
		data	*expression*	*combined*
closed	**all sound**	*coNP*	*coNP*	*coNP*
	all exact	*coNP*	*coNP*	*coNP*
	arbitrary	*coNP*	*coNP*	*coNP*
open	**all sound**	*coNP*	*PSPACE*	*PSPACE*
	all exact	*coNP*	*PSPACE*	*PSPACE*
	arbitrary	*coNP*	*PSPACE*	*PSPACE*

Consequence of complexity results

+ The view-based query answering algorithm provides a set of answers that is sound and complete

– A coNP data complexity does not allow for effective deployment of the query answering algorithm

  Note that coNP-hardness holds already for queries and views that are unions of simple paths (no reflexive-transitive closure)

Consequence of complexity results

+ The view-based query answering algorithm provides a set of answers that is sound and complete

– A coNP data complexity does not allow for effective deployment of the query answering algorithm

  Note that coNP-hardness holds already for queries and views that are unions of simple paths (no reflexive-transitive closure)

$\leadsto$ Adopt an indirect approach to answering queries to a data integration system, via query rewriting

Query rewriting

A rewriting R of a query Q is a query over the source alphabet that, when evaluated over a source database $\mathcal{D}$, provides only certain answers for Q

- We consider rewritings belonging to a certain class $\mathcal{C}$ (e.g., 2RPQs)
- We want rewritings that are maximal among those in $\mathcal{C}$
- We aim at rewritings that are exact, i.e., "equivalent" to the query

Query rewriting

A rewriting R of a query Q is a query over the source alphabet that, when evaluated over a source database $\mathcal{D}$, provides only certain answers for Q

- We consider rewritings belonging to a certain class $\mathcal{C}$ (e.g., 2RPQs)
- We want rewritings that are maximal among those in $\mathcal{C}$
- We aim at rewritings that are exact, i.e., "equivalent" to the query

Example: Given $\mathbf{s}_1$, $\mathbf{s}_2$, $\mathbf{s}_3$ and the mapping

$$V_{\mathbf{s}_1}(b, a) \leftarrow b\ (\textbf{article})\ a \quad V_{\mathbf{s}_2}(p_1, p_2) \leftarrow p_1\ (\textbf{ref}^*)\ p_2 \quad V_{\mathbf{s}_3}(p, t) \leftarrow p\ (\textbf{title})\ t$$

Consider $\quad Q(x, y) \leftarrow x\ (\textbf{article}\cdot(\textbf{ref} + \textbf{ref}^-)^*\cdot\textbf{title})\ y$

Query rewriting

A rewriting R of a query Q is a query over the source alphabet that, when evaluated over a source database $\mathcal{D}$, provides only certain answers for Q

- We consider rewritings belonging to a certain class $\mathcal{C}$ (e.g., 2RPQs)
- We want rewritings that are maximal among those in $\mathcal{C}$
- We aim at rewritings that are exact, i.e., "equivalent" to the query

Example: Given $\mathbf{s}_1$, $\mathbf{s}_2$, $\mathbf{s}_3$ and the mapping

$$V_{\mathbf{s}_1}(b, a) \leftarrow b\ (\textbf{article})\ a \quad V_{\mathbf{s}_2}(p_1, p_2) \leftarrow p_1\ (\textbf{ref}^*)\ p_2 \quad V_{\mathbf{s}_3}(p, t) \leftarrow p\ (\textbf{title})\ t$$

Consider $\quad Q(x, y) \leftarrow x\ (\textbf{article}\cdot(\textbf{ref} + \textbf{ref}^-)^*\cdot\textbf{title})\ y$

- $R_1(x, y) \leftarrow x\ (\mathbf{s}_1\cdot\mathbf{s}_2\cdot\mathbf{s}_3)\ y$ is an RPQ rewriting of Q

Query rewriting

A rewriting R of a query Q is a query over the source alphabet that, when evaluated over a source database $\mathcal{D}$, provides only certain answers for Q

- We consider rewritings belonging to a certain class $\mathcal{C}$ (e.g., 2RPQs)
- We want rewritings that are maximal among those in $\mathcal{C}$
- We aim at rewritings that are exact, i.e., “equivalent” to the query

Example: Given s_1, s_2, s_3 and the mapping

$$V_{\mathsf{s}_1}(b,a) \leftarrow b\ (\mathsf{article})\ a \quad V_{\mathsf{s}_2}(p_1,p_2) \leftarrow p_1\ (\mathsf{ref}^*)\ p_2 \quad V_{\mathsf{s}_3}(p,t) \leftarrow p\ (\mathsf{title})\ t$$

Consider $\quad Q(x,y) \leftarrow x\ (\mathsf{article}\cdot(\mathsf{ref} + \mathsf{ref}^-)^*\cdot\mathsf{title})\ y$

- $R_1(x,y) \leftarrow x\ (\mathsf{s}_1\cdot\mathsf{s}_2\cdot\mathsf{s}_3)\ y$ is an RPQ rewriting of Q
- $R_2(x,y) \leftarrow x\ (\mathsf{s}_1\cdot(\mathsf{s}_2 + \mathsf{s}_2{}^-)\cdot\mathsf{s}_3)\ y$ is a 2RPQ rewriting of Q

Query rewriting

A rewriting R of a query Q is a query over the source alphabet that, when evaluated over a source database $\mathcal{D}$, provides only certain answers for Q

- We consider rewritings belonging to a certain class $\mathcal{C}$ (e.g., 2RPQs)
- We want rewritings that are maximal among those in $\mathcal{C}$
- We aim at rewritings that are exact, i.e., "equivalent" to the query

Example: Given s_1, s_2, s_3 and the mapping

$$V_{\mathsf{s}_1}(b,a) \leftarrow b\ (\mathsf{article})\ a \quad V_{\mathsf{s}_2}(p_1,p_2) \leftarrow p_1\ (\mathsf{ref}^*)\ p_2 \quad V_{\mathsf{s}_3}(p,t) \leftarrow p\ (\mathsf{title})\ t$$

Consider $\quad Q(x,y) \leftarrow x\ (\mathsf{article}\cdot(\mathsf{ref} + \mathsf{ref}^-)^*\cdot\mathsf{title})\ y$

- $R_1(x,y) \leftarrow x\ (\mathsf{s}_1\cdot\mathsf{s}_2\cdot\mathsf{s}_3)\ y$ is an RPQ rewriting of Q
- $R_2(x,y) \leftarrow x\ (\mathsf{s}_1\cdot(\mathsf{s}_2 + \mathsf{s}_2{}^-)\cdot\mathsf{s}_3)\ y$ is a 2RPQ rewriting of Q
- $R_3(x,y) \leftarrow x\ (\mathsf{s}_1\cdot(\mathsf{s}_2 + \mathsf{s}_2{}^-)^*\cdot\mathsf{s}_3)\ y$ is a 2RPQ-maximal rewriting of Q that is also exact

Complexity of query rewriting for RPQs and 2RPQs

We consider RPQ/2RPQ queries and views, and rewritings that are RPQs/2RPQs:

- Existence of a nonempty rewriting is EXPSPACE-complete
- The shortest nonempty rewriting may be of double exponential size
- Existence of an exact rewriting is 2EXPSPACE-complete

Upper bounds by automata-based techniques
Lower bounds by reductions from bounded tiling problems
(from [PODS'99, JCSS'02] for RPQs and [PODS'00] for 2RPQs)

Complexity of query rewriting for RPQs and 2RPQs

We consider RPQ/2RPQ queries and views, and rewritings that are RPQs/2RPQs:

- Existence of a nonempty rewriting is EXPSPACE-complete
- The shortest nonempty rewriting may be of double exponential size
- Existence of an exact rewriting is 2EXPSPACE-complete

Upper bounds by automata-based techniques
Lower bounds by reductions from bounded tiling problems
(from [PODS'99, JCSS'02] for RPQs and [PODS'00] for 2RPQs)

Note that the complexity is in the size of the query and the views, and not in the size of the data

Query answering by rewriting

To answer a query Q wrt a data integration system $\mathcal{I} = \langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$ and a source database $\mathcal{D}$:

1. re-express Q in terms of the sources $\mathcal{S}$, i.e., compute a rewriting of Q
2. directly evaluate the rewriting over $\mathcal{D}$

Query answering by rewriting

To answer a query Q wrt a data integration system $\mathcal{I} = \langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$ and a source database $\mathcal{D}$:

1. re-express Q in terms of the sources $\mathcal{S}$, i.e., compute a rewriting of Q
2. directly evaluate the rewriting over $\mathcal{D}$

Comparison with direct approach to query answering:

- **+** We can consider rewritings in a class with polynomial data complexity (e.g., 2RPQs) $\leadsto$ the data complexity for query answering is polynomial
- **+/–** We have traded expression complexity for data complexity
- **–** We may lose completeness (i.e., not obtain all certain answers)

Query answering by rewriting

To answer a query Q wrt a data integration system $\mathcal{I} = \langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$ and a source database $\mathcal{D}$:

1. re-express Q in terms of the sources $\mathcal{S}$, i.e., compute a rewriting of Q
2. directly evaluate the rewriting over $\mathcal{D}$

Comparison with direct approach to query answering:

+ We can consider rewritings in a class with polynomial data complexity (e.g., 2RPQs) $\leadsto$ the data complexity for query answering is polynomial

+/– We have traded expression complexity for data complexity

– We may lose completeness (i.e., not obtain all certain answers)

We need to establish the "quality" of a rewriting:

- When does the (maximal) rewriting compute all certain answers?
- What do we gain or lose by considering rewritings in a given class?

Query containment

We need techniques to compare the results of queries and/or rewritings

Basic task is query containment:

Q_1 is contained in Q_2 if $Q_1(\mathcal{B}) \subseteq Q_2(\mathcal{B})$ for every database $\mathcal{B}$

Complexity of containment for queries over semistructured data:

Language	Complexity	
RPQs	PSPACE	[PODS'99]
2RPQs	PSPACE	[PODS'00]
Tree-2RPQs	PSPACE	[DBPL'01]
Conjunctive-2RPQs	EXPSPACE	[KR'00]
Datalog in Unions of C2RPQs	2EXPTIME	[ICDT'03]

View-based containment

In a data integration setting, traditional containment does not do the right job:

- we may need to compare a query over the global schema with a query over the sources
- we must take into account the information in the mapping $\mathcal{M}$, considering also that views are sound

View-based containment

In a data integration setting, traditional containment does not do the right job:

- we may need to compare a query over the global schema with a query over the sources
- we must take into account the information in the mapping $\mathcal{M}$, considering also that views are sound

We need to resort to view-based containment:

Compare the results of two queries over the global schema / the sources for all source databases $\mathcal{D}$ and all databases $\mathcal{B}$ that satisfy $\mathcal{M}$ wrt $\mathcal{D}$.

- for a query over the global schema, the result is the certain answers wrt $\mathcal{D}$
- for a query over the sources, the result is the evaluation over $\mathcal{D}$

Complexity of view-based containment for 2RPQs

Let $Q_i^{\mathcal{G}}$ be a query over the global schema
$Q_i^{\mathcal{S}}$ be a query over the sources

Case	Complexity
(1) $Q_1^{\mathcal{G}} \subseteq_{\mathcal{M}} Q_2^{\mathcal{G}}$	NEXPTIME-complete
(2) $Q_1^{\mathcal{S}} \subseteq_{\mathcal{M}} Q_2^{\mathcal{G}}$	PSPACE-complete
(3) $Q_1^{\mathcal{G}} \subseteq_{\mathcal{M}} Q_2^{\mathcal{S}}$	NEXPTIME-complete
(4) $Q_1^{\mathcal{S}} \subseteq_{\mathcal{M}} Q_2^{\mathcal{S}}$	PSPACE-complete

Upper bounds exploit characterization of certain answers via CSP [PODS'03]

Allows for:

(2) establishing whether a given query over the sources is a rewriting

(3) determining whether a rewriting is perfect (i.e, provides all certain answers)

(4) comparing rewritings

Conclusions

- Established decidability and characterized complexity of fundamental query processing tasks for integration of semistructured data:
  - view-based query answering
  - view-based query rewriting
  - query containment and view-based query containment
- Basic technical tools to establish upper bounds:
  - two-way word automata
  - characterization of certain answers in terms of constraint satisfaction

Conclusions

- Established decidability and characterized complexity of fundamental query processing tasks for integration of semistructured data:
  - view-based query answering
  - view-based query rewriting
  - query containment and view-based query containment
- Basic technical tools to establish upper bounds:
  - two-way word automata
  - characterization of certain answers in terms of constraint satisfaction

Further work

- Extend results to exact views
- Take into account constraints (e.g., DTDs, keys, . . .)
- Identify and study most expressive “well-behaved” query language for semistructured data

Thank you!