Completeness of Queries over Incomplete Databases

Werner Nutt

joint work with Marco Montali, Sergey Paramonov, Simon Razniewski, Ognjen Savkovic, Alex Tomasi, Fariz Darari

(VLDB‘11, CIKM‘12, BPM‘13, ISWC‘13)
Incompleteness is omnipresent in data management

- **Null values** in relational databases: Codd 1975
- **Representation systems**: Imielinski/Lipski
  - 1984 Focus on certain/possible answers
- **Query completeness** over incomplete databases: little attention
School Data Management in Bolzano

decentrally maintained database („Popcorn“)

generally incomplete require complete data
Incompleteness in the School Data

Facts in real world

$result$(Paul, Math, A)
$result$(Giulia, Math, A)

Facts in school database

$result$(Paul, Math, NULL)

Missing information in the school database:
- no entry for Giulia (missing record)
- no grade for Paul (missing value)
Consequence: Query Answers are Incorrect

Query Q: "How many pupils have grade A in Math?"

In the real world:

\[ Q(\text{Paul, Math, A}) = 2 \]
\[ Q(\text{Giulia, Math, A}) \]

According to available database:

\[ Q(\text{Paul, Math, NULL}) = 0 \]

→ If data is incomplete, query answers become incorrect.
Why are Data About Pupils Incomplete?

- Data have **not yet** been **copied** from the local school database to the central database
- The **copying** procedure has been **aborted**
- Pupils have been already registered/ classes have been formed, but pupils have **not yet** been **entered** into the database
- **Some schools** (e.g. vocational schools) administer **student grades** with Popcorn, **others not**
- School careers of **immigrants** are often not captured
But: Data are Partially Complete

- Grades of students at vocational schools are complete ...

- Grades of students at vocational schools are complete, … after reports have been handed in.

- Classes at school X are complete, when the classes have been formed … and entered into Popcorn

How can we use information about partial completeness? Meta data!
Use Metadata to Guarantee Completeness!

Suppose, we know whether parts of a db are complete, e.g.,

- "The grades from vocational schools are complete"
- "The Math grades from primary schools are complete"

→ Idea: Assess completeness of a query using completeness assertions for (parts of) tables
Reasoning about Query Completeness

You cannot,
because
information about pupils from high schools could be missing.

I want to know “How many pupils have grade A in Math?“ Can I trust the query answer?

Grades from vocational schools are complete.

All Math grades from primary schools are complete.

Biology grades from high schools are complete.

Assertions about partial completeness.

Space of possible information.
You can, because all needed information is complete in the database.
Research Questions: How can one ...

1. ... formalize completeness of query answers?

2. ... assert completeness of parts of a possibly incomplete database?

3. ... infer completeness of query answers from such assertions?

4. ... implement such reasoning techniques?

All Math grades from primary schools are complete.

Grades from vocational schools are complete.

Space of possible information

Biology grades from high schools are complete.

You can, because all needed information is complete in the database.
MAGIK (= Managing Incomplete Knowledge)

- **Database queries** = logical formulas
- **Completeness meta data** = logical formulas
- **Analysis** = logical inference
- **Implementation**: using software for logical inferences

**Interface Layer**
- Input
- TC-statements
- Gen. and Spec.
- Output

**Data Layer**
- Partially Complete Database
- Read Database Schema (database mode)
- Read/Write Virtual Schema (virtual mode)
- TCs and Queries
- Meta-information Storage
- Evaluate SQL Queries (database mode)

**Program Business Logic**
- TuProlog
- DLV Engine

**Ideas:**
1. Database queries = logical formulas
2. Completeness meta data = logical formulas
3. Analysis = logical inference
4. Implementation: using software for logical inferences
Running Example: Schema

result(name, subject, grade)

pupil(name, age, schoolName, schoolType)
Notation: Databases

Database instances are sets of ground atoms, e.g.,

\[ D = \{ \text{result(Paul, Math, NULL)}, \]
\[ \text{result(Giulia, Math, A)}, \]
\[ \text{pupil(Paul, 17, Verdi, Voc)} \}, \]

possibly containing NULLs.
Notation: Conjunctive Queries

A single block SQL queries, possibly with DISTINCT,

\[
\text{SELECT } r.\text{grade} \\
\text{FROM } \text{result } r, \text{ pupil } p \\
\text{WHERE } r.\text{name} = p.\text{name} \text{ AND} \\
\text{r.\text{subject} = 'Math' AND} \\
p.\text{age} \leq 11
\]

is expressed as a conjunctive query (CQ), using a Datalog rule:

\[
Q(g) :- \text{result}(n, \text{Math}, g), \text{pupil}(n, a, sn, st), a \leq 11
\]
Notation: Conjunctive Queries (2)

\[ Q(x) :\!-\! L(x, y), M \]

- \( L(x, y) \) conjunction of relational atoms
- \( M \) conjunction of comparisons
- \( x \) vector of *distinguished* (= output) variables
- \( y \) vector of *non-distinguished* (= existential) variables

Query answers (under *set semantics*):

\[ Q(D) = \{ \alpha x \mid \alpha L \subseteq D, \alpha \models M \} \]

**Bag semantics**: each \( \alpha \) contributes a copy of \( \alpha x \)

As a default, we assume set semantics
Possible Completeness Statements

“We get complete answers to the following queries:

- Which pupils have grade A in Math?
- Which pupils from vocational schools have grade A in Math?

Query Completeness Statements

“The database contains

- all subjects and grades of pupils from vocational schools
- all subjects studied by pupils from vocational schools

Table Completeness Statements
Formalization: Incomplete Database

When talking about incompleteness, we need a complete reference

An incomplete database $D$ is a pair of
- an ideal database $D^i$ and
- an available database $D^a$

$$D = (D^i, D^a)$$

such that

for each record in $D^a$ there is a “more informative” record in $D^i$

For databases w/o Nulls, this means

$$D^a \subseteq D^i$$
Example: An Incomplete Database

Completeness of Queries over Incomplete Databases

29.05.2014
Formalization: Query Completeness

[Motro 1989]

Query Q

“*The answer to Q is complete*”

Notation: \( \text{Compl}(Q) \)

Semantics:

\[
(D^i, D^a) \models \text{Compl}(Q) \quad \text{iff} \quad Q(D^i) = Q(D^a)
\]

To be precise, we have to distinguish between set and bag semantics.
Example: Query Completeness

\[ Q_{\text{MathA}}(n) :- \text{result}(n, \text{Math}, A) \]

\[ Q_{\text{MathA}}(D^i) = \{\text{Paul, Giulia, Maria}\} \quad Q_{\text{MathA}}(D^a) = \{\text{Paul}\} \]

\[ \text{pupil}(\text{Paul}, 17, \text{Hofer}, \text{Voc}) \]
\[ \text{pupil}(\text{Giulia}, 15, \text{Verdi}, \text{Sec}) \]
Example: Query Completeness (2)

\[ Q_{\text{MathAVoc}}(n) : - \text{result}(n, \text{Math}, A), \text{pupil}(n, a, sn, Voc) \]

\[ Q_{\text{MathAVoc}}(D^i) = \{\text{Paul}\} \quad Q_{\text{MathAVoc}}(D^a) = \{\text{Paul}\} \]

\[ \text{result(Paul, Math, A)} \]
\[ \text{result(Giulia, Math, NULL)} \]
\[ \text{result(Maria, Math, A)} \]
\[ \text{pupil(Paul, 17, Hofer, Voc)} \]
\[ \text{pupil(Giulia, 15, Verdi, Sec)} \]

\[ Q_{\text{MathAVoc}} \text{ is complete over } (D^i, D^a) \]

Completeness of Queries over Incomplete Databases
Table Completeness Statements: Idea

“The table result contains all results of pupils from vocational schools” means

“If (n,s,g) is a result record according to the ideal db, and (n, a, sn, Voc) is a pupil record in the ideal db, then (n,s,g) is in the result table of the available db”

This can be expressed by the rule

\[ \text{result}^i(n,s,g), \text{pupil}^i(n, a, sn, Voc) \Rightarrow \text{result}^a(n, s, g) \]

We write this table completeness statement as

\[ \text{Compl}( \text{result}(n, s, g) ; \text{pupil}(n, a, s, Voc)) \]

*Idea: an incomplete db satisfies the statement iff it satisfies the rule*
Table Completeness Statements [Halevy 96]

A table completeness (TC) statement for a relation $R$ is an expression

$$\text{Compl}(R(s_1,\ldots, s_n) ; G)$$

consisting of

- an $R$-atom $R(s_1,\ldots, s_n)$
- a condition $G$ such that $R(s_1,\ldots, s_n), G$ is safe.

The TC-statement $C = \text{Compl}(R(s_1,\ldots, s_n) ; G)$ can be seen as a rule

$$r_C = R^i(s_1,\ldots, s_n), G^i \rightarrow R^a(s_1,\ldots, s_n)$$

Semantics:

$$(D^i, D^a) \models C \iff (D^i, D^a) \models r_C$$

Completeness of Queries over Incomplete Databases 29.05.2014
Example: TC Statement Satisfaction

\[ \text{result}^i(n, s, g), \text{pupil}^i(n, a, sn, \text{Voc}) \rightarrow \text{result}^a(n, s, g) \]

holds over \( (D^i, D^a) \)

because \( \text{result}(Paul, Math, A) \) is in \( D^a \)

Completteness of Queries over Incomplete Databases

29.05.2014
The TC-QC Reasoning Problem

Set of table completeness statements $\mathcal{C}$

Grades from vocational schools are complete

Ungenerated assertions about partial completeness

Grades from primary schools are complete

Space of possible information

I want to know "How many pupils at vocational schools have taken Math?" Can I trust the query answer?

$\mathcal{C} \models \text{Compl}(Q)$?
Reasoning: The Principle

“Which pupils at vocational schools had an A in Math?“

\[ Q_{\text{MathAVoc}}(n) :- \text{result}(n, \text{Math}, \text{A}), \text{pupil}(n, \text{sn}, \text{Voc}) \]

1. Assume \( Q_{\text{MathAVoc}} \) returns \( n' \) over \( D_i \)

2. See which facts must be in \( D_i \)

\begin{align*}
\text{result}(n', \text{Math}, g') \\
\text{pupil}(n', \text{sn}', \text{Voc})
\end{align*}
3. Use table completeness to derive facts in $D^a$

“All results of pupils at vocational schools are available“

$$\text{result}^i(n, s, g), \text{pupil}^i(n, sn, Voc) \Rightarrow \text{result}^a(n, s, g)$$

“All pupils are available“

$$\text{pupil}^i(n, sn, st) \Rightarrow \text{pupil}^a(n, sn, st)$$
4. Query the available database

\[ Q_{\text{MathAVoc}}(D^a) = \{n'\} \rightarrow n' \text{ is also in } Q(D^a) \]

Conclusion: \( Q_{\text{MathAVoc}} \) is complete given the table completeness statements
TC-Transformation

To $C = \text{Compl}(R(s); G)$ we associate the query

$$Q_C(s) :=- R(s), G$$

and the transformation on db instances

$$T_C(D) := \{ R(t) \mid t \in Q_C(D) \}$$

For a set $C$ of TC statements we define the transformation

$$T_C(D) := \bigcup_{C \in C} T_C(D)$$
(D, T_C(D)) is an incomplete database

(D, T_C(D)) ⊨ C

(D^i, D^a) ⊨ C  iff  T_C(D^i) ⊆ D^a

In other words:

(D, T_C(D)) is the least incomplete database
  with ideal db D
  that satisfies C
TC-QC Reasoning: Relational Case

Let

- $C$ set of relational TC statements
- $Q(x) :- L$ relational query
- $L' :=$ frozen version of $L$

**Theorem:** $C \models \text{Compl}(Q)$ iff $x' \in Q(T_C(L'))$

What if $C$ or $Q$ contain comparisons?

Completeness of Queries over Incomplete Databases
Example: TC-QC with Comparisons

Query: \( Q_{pupil}(n) :- pupil(n, a, sn, st) \)

\[ C = \begin{cases} 
C_{\leq 10} & : pupil^i(n, a, sn, st) \\
C_{> 10} & : pupil^i(n, a, sn, st) 
\end{cases} \]

How can we chase \( L' = \{ pupil(n', a', sn', st') \} \) with \( C \leq 10 \)?

Idea: Case analysis!

Substitute “representative values” for \( a' < 10, \ a' = 10, \ a' > 10 \)

Substitution yields: \( [a'/9]L' = \{ pupil(n', 9, sn', st') \} \)

- We retrieve \( n' \) in all 3 cases
- The cases cover all possibilities

\( \rightarrow Q \) is complete wrt \( C \)

Completeness of Queries over Incomplete Databases 29.05.2014
TC-QC Reasoning with Comparisons

Let

- $C$ set of TC statements with comparisons
- $Q(x) :- L, M$
- $\Gamma$ set of representative value substitutions for $C, Q$

**Theorem:** The following are equivalent

- $C \models \text{Compl}(Q)$
- $\gamma x' \in Q(T_C(\gamma L'))$ for all $\gamma \in \Gamma$
Set Semantics vs. Bag Semantics

\[ Q(\mathbf{x}) \setminus L \text{ query} \]

\[(D^i, D^a) \models \text{Compl}_{\text{set}}(Q) \]

iff every answer of \(Q\) over \(D^i\) is returned over \(D^a\), too

iff \(\alpha L \subseteq D^i \Rightarrow \exists \beta \text{ s.th. } \beta L \subseteq D^a\) and \(\beta \mathbf{x} = \alpha \mathbf{x}\)

\[(D^i, D^a) \models \text{Compl}_{\text{bag}}(Q) \]

iff every answer of \(Q\) over \(D^i\) is returned over \(D^a\)

\("\text{the same number of times}\"

iff \(\alpha L \subseteq D^i \Rightarrow \alpha L \subseteq D^a\)

\("\text{no assignments get lost}\"

Completeness of Queries over Incomplete Databases 29.05.2014
TC-QC Reasoning for Bag Semantics

Let

- $C$ set of TC statements with comparisons
- $Q(x) :- L, M$
- $\Gamma$ set of representative value substitutions for $C, Q$

**Theorem:**

$C \models \text{Compl}_{bag}(Q)$ iff $\gamma L' \subseteq T_C(\gamma L')$ for all $\gamma \in \Gamma$

**Corollary:** If $C$ has no comparisons, then:

$C \models \text{Compl}_{bag}(Q)$ iff $L' \subseteq T_C(L')$
Complexity

Classes of conjunctive queries:

- **CQ**: Conjunctive queries with comparisons over dense orders
- **RQ**: Relational conjunctive queries (i.e., without comparisons)
- **LCQ**: Linear conjunctive queries (i.e., without self-joins)
- **LRQ**: Linear relational conjunctive queries
## TC-QC\textsubscript{bag} - Complexity

<table>
<thead>
<tr>
<th>TC Statement Language</th>
<th>LRQ</th>
<th>LCQ</th>
<th>RQ</th>
<th>CQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>LRQ</td>
<td>in PTIME</td>
<td>in PTIME</td>
<td>NP</td>
<td>NP</td>
</tr>
<tr>
<td>RQ</td>
<td>in PTIME</td>
<td>in PTIME</td>
<td>NP</td>
<td>NP</td>
</tr>
<tr>
<td>LCQ</td>
<td>coNP</td>
<td>coNP</td>
<td>$\Pi^P_2$</td>
<td>$\Pi^P_2$</td>
</tr>
<tr>
<td>CQ</td>
<td>coNP</td>
<td>coNP</td>
<td>$\Pi^P_2$</td>
<td>$\Pi^P_2$</td>
</tr>
</tbody>
</table>

Note, the axes are asymmetric:
- NP appears with repeated relation symbols in the query
- coNP appears with comparisons in the TC statements
### TC-QC<sub>set</sub> - Complexity

<table>
<thead>
<tr>
<th>TC Statement Language</th>
<th>LRQ</th>
<th>LCQ</th>
<th>RQ</th>
<th>CQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>LRQ</td>
<td>in PTIME</td>
<td>in PTIME</td>
<td>NP</td>
<td>$\Pi^P_2$</td>
</tr>
<tr>
<td>RQ</td>
<td>in PTIME</td>
<td>in PTIME</td>
<td>NP</td>
<td>$\Pi^P_2$</td>
</tr>
<tr>
<td>LCQ</td>
<td>coNP</td>
<td>coNP</td>
<td>$\Pi^P_2$</td>
<td>$\Pi^P_2$</td>
</tr>
<tr>
<td>CQ</td>
<td>coNP</td>
<td>coNP</td>
<td>$\Pi^P_2$</td>
<td>$\Pi^P_2$</td>
</tr>
</tbody>
</table>

*Intuition: the query has to be contained in the TC-statements ...*  
*... but that does not explain it all*
How Can One Implement Completeness Reasoning?

Idea: Map reasoning tasks to a generic reasoner

Candidate reasoners:

- SMT (SAT modulo theories) solvers?
  - encoding may be of exp. size for $\Pi^p_2$ problems

- Disjunctive Logic Programming with Answer Set Semantics?
  - can express all $\Pi^p_2$ problems
  - demo implementation for
    - conjunctive queries
    - finite domain constraints
    - keys and (acyclic) foreign keys
Completeness of Queries over Incomplete Databases

### Add new query

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q.1</td>
<td>Select the names of all pupils that attend a primary school.</td>
<td>X</td>
</tr>
<tr>
<td>Q.2</td>
<td>Select the names of all pupils that attend a primary school in the Bolzano district and that learn some language.</td>
<td>X</td>
</tr>
<tr>
<td>Q.3</td>
<td>Select the names of all 1st level pupils that attend a school in the Bolzano district and that learn some language.</td>
<td>X</td>
</tr>
<tr>
<td>Q.4</td>
<td>Select all language learners.</td>
<td>X</td>
</tr>
<tr>
<td>Q.5</td>
<td>Select all classes.</td>
<td>X</td>
</tr>
<tr>
<td>Q.6</td>
<td>Who learns English?</td>
<td>X</td>
</tr>
<tr>
<td>Q.8</td>
<td>Select all pupils from schools in Bolzano who learn a language.</td>
<td>X</td>
</tr>
<tr>
<td>Q.0</td>
<td>Give me all pupils from primary schools.</td>
<td>X</td>
</tr>
</tbody>
</table>

### Selected Query

```
SELECT DISTINCT p.pname
FROM pupil AS p, school AS s
WHERE s.type = 'primary'
AND p.sname = s.sname
```

### Back to Schema Selection

#### Result

**Query is not complete**

Completenees calculated in 6 ms
Generalization calculated in 13 ms
Specialization(s) calculated in 143 ms

**Incomplete tables**

The following parts of the tables are incomplete. Please collect the missing data and confirm it by adding the corresponding TC-statements.

<table>
<thead>
<tr>
<th>Table</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>pupil(P_pname, S_sname, P_code)</td>
<td>school(S_sname, 'primary', S_district)</td>
</tr>
</tbody>
</table>

### Complete Query Approximation

#### Query Generalization

Not available

#### Original

```
SELECT DISTINCT p.pname
FROM pupil AS p, school AS s
WHERE s.type = 'primary'
AND p.sname = s.sname
```

#### Query Specialization(s)

```
SELECT pupil1.pname
FROM pupil1 AS pupil1, school AS school1
WHERE pupil1.sname = school1.sname
AND school1.type = 'primary'
AND school1.district = 'Bolzano'
```

Maximal size of Specialization queries is original query size.
Completeness on the Semantic Web

I want to find out all movies directed by and starring Tarantino
DBPedia Misses Some Facts ...

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbpedia-owl:runtime</td>
<td>5940 (xsd:double)</td>
</tr>
<tr>
<td>dbpedia-owl:starring</td>
<td></td>
</tr>
<tr>
<td></td>
<td>dbpedia:Chris_Penn</td>
</tr>
<tr>
<td></td>
<td>dbpedia:Tim_Roth</td>
</tr>
<tr>
<td></td>
<td>dbpedia:Lawrence_Tierney</td>
</tr>
<tr>
<td></td>
<td>dbpedia:Steve_Buscemi</td>
</tr>
<tr>
<td></td>
<td>dbpedia:Harvey_Keitel</td>
</tr>
<tr>
<td></td>
<td>dbpedia:Michael_Madsen</td>
</tr>
</tbody>
</table>
IMDB Has Completeness Guarantees

Full cast and crew for Reservoir Dogs (1992) More at IMDbPro »

IMDbPro.com offers representation listings for over 120,000 individuals, including actors, directors, and producers, as well as company and employee contact details for over 50,000 companies in the entertainment industry. Click here for a free trial!

Directed by
Quentin Tarantino

Writing credits
Quentin Tarantino (written by)
Roger Avary (background radio dialog) &
Quentin Tarantino (background radio dialog)

Cast (in credits order) verified as complete

Harvey Keitel ... Mr. White - Larry Dimmick
Edward Bunker ... Mr. Blue (as Eddie Bunker)
Quentin Tarantino ... Mr. Brown

Completeness statement about the IMDB data source

Quentin Tarantino was the character Mr. Brown

Completeness of Queries over Incomplete Databases 29.05.2014
If Completeness Info Were Available in RDF ...

I want to find out **all** movies directed by and starring Tarantino

LMDb is **complete** for Tarantino movies and their actors

DBpedia is **complete** for Tarantino movies
Federated Framework

Can I get a complete answer?

```sql
SELECT ?m ?l
WHERE {
  ?m rdf:type s:Movie .
  ?m s:director dbp:tarantino .
  ?m fbo:likes ?l
}
```
Completeness of SPARQL Queries over RDF Sources

- Completeness statements in RDF
- Reasoning algorithms for queries with
  - DISTINCT
  - OPT
  - over RDFS sources
- Generation of queries with SERVICE calls
  over federated sources
- Prototypical implementation using Apache Jena
  \[http://rdfcorner.wordpress.com\]
Verifying Query Completeness over Processes

- Data often created following processes

- Many processes are executed only partially formal (pen & paper, email, phone, …)

→ Valid information may be stored in databases with delays

→ Database content is of questionable completeness
Enrolment Process in a School

Database query: How many pupils? 0
Is that correct?

Database query: How many pupils? 137
Is that correct?
Observation

- At some points, **new facts** in the real world have **not yet been stored**
  - → **queries** may give **wrong answers**
- At other points, **all facts** that hold in the real world have been **stored**
  - → **queries** give **correct answers**
Real-world effect: \( \text{pupil}^{\text{rw}}(n, s) \leftrightarrow \text{request}^{\text{rw}}(n, s) \)

Copy effect: \( \text{pupil}^{\text{rw}}(n, s) \rightarrow \text{pupil}^{\text{is}}(n, s) \)

Completeness of Queries over Incomplete Databases 29.05.2014
Transition Systems for Process Instances

Real-world effect: Generates enrolments

Copy effect: Copies the new enrolments into the school database

Completeness of Queries over Incomplete Databases
Transition Systems for Process Instances

Two concurrent process instances:

- Middle School A
- High School B
Completeness Verification

Given
- Process description
- State S
- Query Q

Question

Is it safe to pose the query Q in state S against the information system database?
Verification: Example Revisited

Middle School A
High School B

How many middle school pupils?
How many high school pupils?

Decide enrolments
Record enrolments
Decide enrolments
Record enrolments
Decide enrolments
Record enrolments
Decide enrolments
Record enrolments
Possible Applications

- **Annotation of statistics and KPI** with completeness information
- **Process mining** (trace analysis) - to validate whether queries over traces return the real state of the process
- **Auditing** – to verify whether the information about the real-world is properly stored
Conclusion

- Framework for statements about completeness of
  - query answers
  - (projections of) parts of db tables

- Complexity of TC-QC Reasoning

- Implementation based on DLV answer set programming engine

- Application to
  - Semantic Web
  - Business Processes
Questions?