Modéliser et Interroger des Données Incertaines

Jef Wijsen

UMONS

Séminaire Jeunes, Mons, 13 April 2016

Préambule

- Recherche en collaboration avec Paraschos Koutris, University of Wisconsin-Madison, USA
- Notre travail [KW15] est récipiendaire du prix

  ACM SIGMOD Research Highlight Award 2015[☺]

  "for representing a definitive milestone in solving an important problem"
- Cet exposé est organisé comme suit: ...
- Pour ceux qui n'ont jamais suivi un cours de Bases de données: ...

[☺] ACM=la première association scientifique dans le domaine de l'informatique
SIGMOD=Special Interest Group on Management of Data

Préambule

- Recherche en collaboration avec Paraschos Koutris, University of Wisconsin-Madison, USA
- Notre travail [KW15] est récipiendaire du prix

  ACM SIGMOD Research Highlight Award 2015[☺]

  "for representing a definitive milestone in solving an important problem"
- Cet exposé est organisé comme suit: ...
- Pour ceux qui n'ont jamais suivi un cours de Bases de données: ...

[☺] ACM=la première association scientifique dans le domaine de l'informatique
SIGMOD=Special Interest Group on Management of Data

Préambule

- Recherche en collaboration avec Paraschos Koutris, University of Wisconsin-Madison, USA
- Notre travail [KW15] est récipiendaire du prix

  ACM SIGMOD Research Highlight Award 2015[☺]

  "for representing a definitive milestone in solving an important problem"
- Cet exposé est organisé comme suit: ...
- Pour ceux qui n'ont jamais suivi un cours de Bases de données: ...

[☺] ACM=la première association scientifique dans le domaine de l'informatique
SIGMOD=Special Interest Group on Management of Data

Préambule

- Recherche en collaboration avec Paraschos Koutris, University of Wisconsin-Madison, USA
- Notre travail [KW15] est récipiendaire du prix

  ACM SIGMOD Research Highlight Award 2015[☺]

  "for representing a definitive milestone in solving an important problem"
- Cet exposé est organisé comme suit: ...
- Pour ceux qui n'ont jamais suivi un cours de Bases de données: ...

[☺] ACM=la première association scientifique dans le domaine de l'informatique
SIGMOD=Special Interest Group on Management of Data

Modeling Uncertainty in the Relational Data Model

Starting Idea

Let us model uncertainty by primary key violations.

Example (Primary keys are underlined)

ManagedBy

Dept	*Mgr*	*Budget*
CIA	Barack	60M
CIA	Barack	65M
MI6	James	15M

WorksFor

Agent	*Dept*
Sherlock	MI6
James	CIA
James	MI6

- The budget of CIA is either 60M or 65M.
- James works for either CIA or MI6 (but not both).

Definition (Block)

A block is a maximal set of tuples of the same relation that agree on the primary key (representing a disjunction of alternative tuples).

Modeling Uncertainty in the Relational Data Model

Starting Idea

Let us model uncertainty by primary key violations.

Example (Primary keys are underlined)

ManagedBy

Dept	*Mgr*	*Budget*
CIA	Barack	60M
CIA	Barack	65M
MI6	James	15M

WorksFor

Agent	*Dept*
Sherlock	MI6
James	CIA
James	MI6

- The budget of CIA is either 60M or 65M.
- James works for either CIA or MI6 (but not both).

Definition (Block)

A block is a maximal set of tuples of the same relation that agree on the primary key (representing a disjunction of alternative tuples).

Certain Answers

Definition (Repair and Certain Answers)

A repair is obtained by selecting exactly one tuple from each block.
The certain answer to a query q is the intersection of the query answers over all repairs.

Example

WorksFor	*Agent*	*Dept*
	Sherlock	MI6
	James	CIA
	James	MI6

- Who works for MI6? $\rightsquigarrow q = \{a \mid \mathit{WorksFor}(\underline{a}, \text{'MI6'})\}$
- The certain answer to q contains Sherlock, but not James.

Certain Answers

Definition (Repair and Certain Answers)

A repair is obtained by selecting exactly one tuple from each block.
The certain answer to a query q is the intersection of the query answers over all repairs.

Example

WorksFor	*Agent*	*Dept*
	Sherlock	MI6
	James	CIA
	James	MI6

- Who works for MI6? $\rightsquigarrow q = \{a \mid \mathit{WorksFor}(\underline{a}, \text{'MI6'})\}$
- The certain answer to q contains Sherlock, but not James.

Is it Difficult to Compute Consistent Answers? I

Example

WorksFor	*Agent*	*Dept*
	Sherlock	MI6
	James	CIA
	James	MI6

- $q = \{a \mid \mathit{WorksFor}(\underline{a}, \text{'MI6'})\}$
- It is not difficult to see that the certain answer to q is obtained by the following query:

$$\{a \mid \mathit{WorksFor}(\underline{a}, \text{'MI6'}) \wedge \underbrace{\neg \exists d\, (\mathit{WorksFor}(\underline{a}, d) \wedge d \neq \text{'MI6'})}_{\text{agent } a \text{ works for no other department}}\}$$

Is it Difficult to Compute Consistent Answers? II

Example

ManagedBy

Dept	*Mgr*	*Budget*
CIA	Barack	60M
CIA	Barack	65M
MI6	James	15M

WorksFor

Agent	*Dept*
Sherlock	MI6
James	CIA
James	MI6

- Get the budget of self-managed departments (i.e., managed by an agent of the department).

$$q = \{b \mid \exists d \exists m\,(\mathit{ManagedBy}(\underline{d}, m, b) \wedge \mathit{WorksFor}(\underline{m}, d))\}$$

- It is known [Wij10] that there is no query in first-order logic that returns the certain answer to q.

Is it Difficult to Compute Consistent Answers? III

Definition

For every query q in first-order logic, the problem CERTAINTY(q) is the following:

Input A database instance (possibly with primary-key violations)

Question Is the certain answer to q non-empty?

Note:

- We use a decision problem (non-emptiness check) for convenience.
- The complexity is data complexity (i.e., q is not part of the input).

Is it Difficult to Compute Consistent Answers? IV

Complexity Classification Task

Input A query q in first-order logic

Question What complexity classes does CERTAINTY(q) belong to?
Complexity classes of interest:

$$\mathbf{FO} \subsetneq \mathbf{L} \subseteq \mathbf{NL} \subseteq \mathbf{P} \subseteq \mathbf{coNP}$$

Note:

- CERTAINTY(q) belongs to the descriptive complexity class **FO** iff there exists a query in first-order logic that computes the certain answer to q.

Is it Difficult to Compute Consistent Answers? V

Example

$$
\begin{aligned}
q_1 &= \{a \mid \mathit{WorksFor}(\underline{a}, \text{`MI6'})\} \\
q_2 &= \{b \mid \exists d \exists m\, (\mathit{ManagedBy}(\underline{d}, m, b) \wedge \mathit{WorksFor}(\underline{m}, d))\} \\
q_3 &= \{b \mid \exists d \exists m \exists x\, (\mathit{ManagedBy}(\underline{d}, x, b) \wedge \mathit{WorksFor}(\underline{m}, x))\}^a
\end{aligned}
$$

- CERTAINTY(q_1) is in **FO**;
- CERTAINTY(q_2) is in **P** but not in **FO** [Wij10]; and
- CERTAINTY(q_3) is **coNP**-complete [CM05].

[a] "Get budgets for departments whose manager's name is also the name of a department."

What Can Cause Exponential Growth?

Relation with exponentially many repairs

WorksFor

Agent	*Dept*
1	MI6
1	CIA
2	MI6
2	CIA
⋮	⋮
n	MI6
n	CIA

This *WorksFor* relation contains $2n$ tuples and has 2^n distinct repairs.

Main Result

Theorem (Complexity Classification)

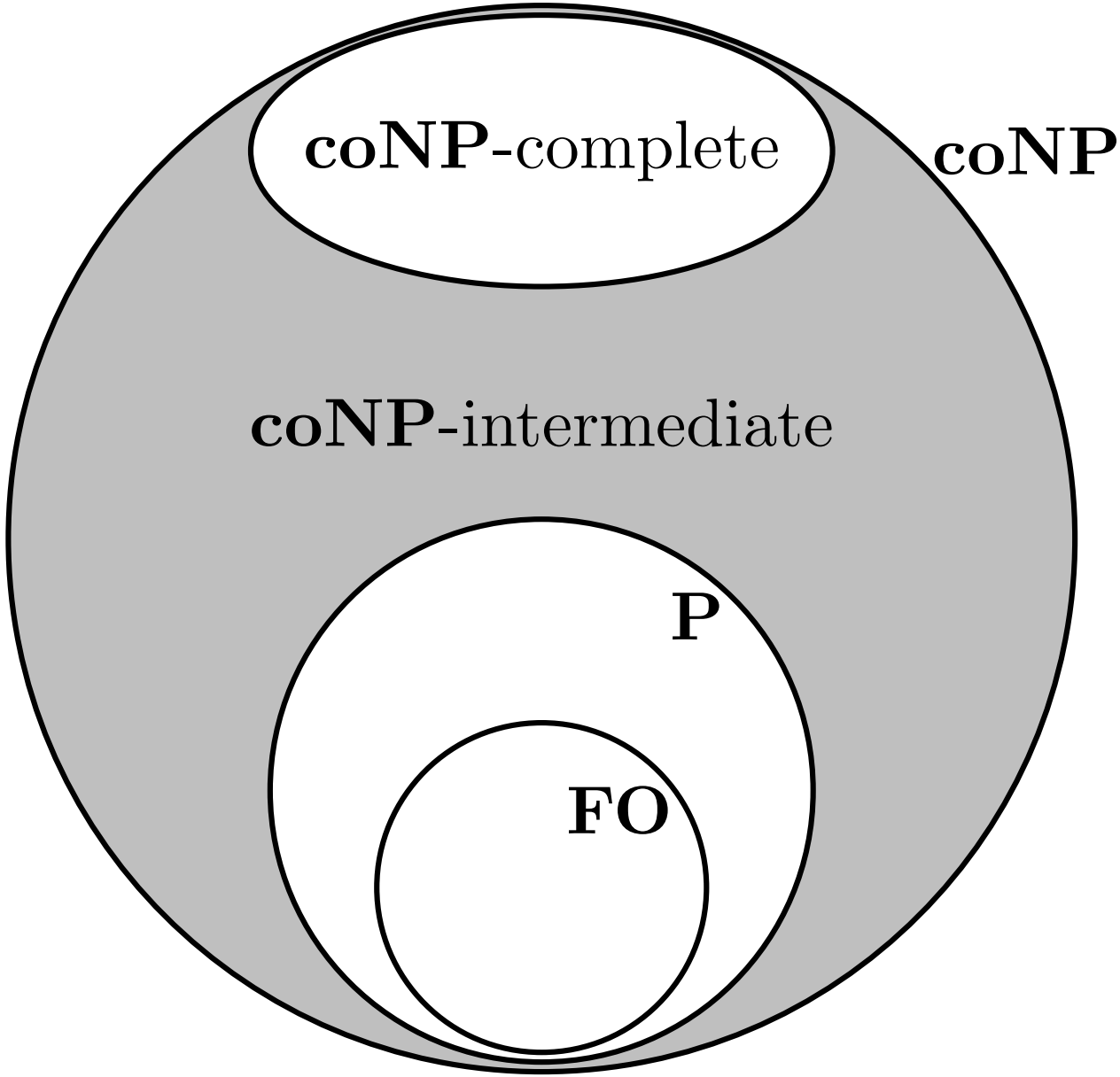
For every query q in first-order logic that is conjunctive and self-join-free, the following hold:

1. CERTAINTY(q) *is either in* **P** *or* **coNP**-*complete (and the dichotomy is decidable); and*
2. *it can be decided whether* CERTAINTY(q) *is in* **FO**.

Note:

- A query in first-order logic is conjunctive it it uses only conjunction ($\wedge$) and existential quantification ($\exists$).
- A conjunctive query is self-join-free if no relation name occurs more than once in it.
- For example, $\{a \mid \exists d\,(\mathit{WorksFor}(\underline{a}, d) \wedge \mathit{WorksFor}(\text{`}\underline{\text{Sherlock}}\text{'}, d))\}$ is conjunctive but not self-join-free.

The Geography of **coNP** (assuming **P** ≠ **coNP**)



Why is this Interesting?

Theoretically Proving dichotomy theorems for large problem classes is challenging. Our main theorem settles a dichotomy that was an open conjecture for 10 years.

Practically

- Uncertainty and certain answers arise in many applications. Currently, the only support for uncertainty in SQL is the NULL value ☹.
- If CERTAINTY(q) is in FO, then it can be solved by means of standard SQL database technology. This works in practice.

Why is this Interesting?

Theoretically Proving dichotomy theorems for large problem classes is challenging. Our main theorem settles a dichotomy that was an open conjecture for 10 years.

Practically

- Uncertainty and certain answers arise in many applications. Currently, the only support for uncertainty in SQL is the NULL value ☹.
- If CERTAINTY(q) is in **FO**, then it can be solved by means of standard SQL database technology. This works in practice.
  (Euh... Demande à Alexandre/Fabian/Damien/Franck.)

Why is this Interesting?

Theoretically Proving dichotomy theorems for large problem classes is challenging. Our main theorem settles a dichotomy that was an open conjecture for 10 years.

Practically

- Uncertainty and certain answers arise in many applications. Currently, the only support for uncertainty in SQL is the NULL value ☹.
- If CERTAINTY(q) is in **FO**, then it can be solved by means of standard SQL database technology. This works in practice.
  (Euh... Demande à Alexandre/Fabian/Damien/Franck.)

Why is this Interesting?

Theoretically Proving dichotomy theorems for large problem classes is challenging. Our main theorem settles a dichotomy that was an open conjecture for 10 years.

Practically

- Uncertainty and certain answers arise in many applications. Currently, the only support for uncertainty in SQL is the NULL value ☹.
- If CERTAINTY(q) is in **FO**, then it can be solved by means of standard SQL database technology. This works in practice.
  (Euh... Demande à Alexandre/Fabian/Damien/Franck.)

Future Work

- For a master thesis, write a polynomial-time program:

  Input Self-join-free conjunctive query q s.t. CERTAINTY(q) is in **P**; a database

  Output The certain answer to q

- For a PhD thesis, show the following:

Conjecture

For every conjunctive query q, CERTAINTY(q) *is in* **P** *or* **coNP**-*complete.*

- For *"gloire éternelle,"* show the following:

Conjecture

For every union of conj. queries q, CERTAINTY(q) *is in* **P** *or* **coNP**-*complete.*

It is known [Fon13] that the latter conjecture implies Bulatov's complexity dichotomy theorem for conservative CSP [Bul11], the proof of which is very involved (the full paper contains 66 pages).

Future Work

- For a master thesis, write a polynomial-time program:

  Input Self-join-free conjunctive query q s.t. CERTAINTY(q) is in **P**; a database

  Output The certain answer to q

- For a PhD thesis, show the following:

Conjecture

For every conjunctive query q, CERTAINTY(q) *is in* **P** *or* **coNP***-complete.*

- For *"gloire éternelle,"* show the following:

Conjecture

For every union of conj. queries q, CERTAINTY(q) *is in* **P** *or* **coNP***-complete.*

It is known [Fon13] that the latter conjecture implies Bulatov's complexity dichotomy theorem for conservative CSP [Bul11], the proof of which is very involved (the full paper contains 66 pages).

Future Work

- For a master thesis, write a polynomial-time program:

  Input Self-join-free conjunctive query q s.t. CERTAINTY(q) is in **P**; a database

  Output The certain answer to q

- For a PhD thesis, show the following:

Conjecture

For every conjunctive query q, CERTAINTY(q) *is in* **P** *or* **coNP**-*complete.*

- For *"gloire éternelle,"* show the following:

Conjecture

For every union of conj. queries q, CERTAINTY(q) *is in* **P** *or* **coNP**-*complete.*

It is known [Fon13] that the latter conjecture implies Bulatov's complexity dichotomy theorem for conservative CSP [Bul11], the proof of which is very involved (the full paper contains 66 pages).

Remerciement

Merci à tous, en premier lieu à Quentin !

References I

Andrei A. Bulatov.

Complexity of conservative constraint satisfaction problems.
ACM Trans. Comput. Log., 12(4):24, 2011.

Jan Chomicki and Jerzy Marcinkowski.

Minimal-change integrity maintenance using tuple deletions.
Inf. Comput., 197(1-2):90–121, 2005.

Gaëlle Fontaine.

Why is it hard to obtain a dichotomy for consistent query answering?
In *LICS*, pages 550–559. IEEE Computer Society, 2013.

Paris Koutris and Jef Wijsen.

The data complexity of consistent query answering for self-join-free conjunctive queries under primary key constraints.
In Tova Milo and Diego Calvanese, editors, *PODS*. ACM, 2015.

Jef Wijsen.

A remark on the complexity of consistent conjunctive query answering under primary key violations.
Inf. Process. Lett., 110(21):950–955, 2010.