Certain Query Answering for Primary Keys in First-Order Logic [1] [2] [3]

Jef Wijsen

UMONS

HIGHLIGHTS of **logic**, games and automata
Brussels, 6–9 September 2016

[1] Joint work with Paris Koutris, Univ. of Wisconsin-Madison
[2] first presented at PODS 2015 [KW15]
[3] awarded ACM SIGMOD Research Highlight Award 2015 [KW16]

Disjunctive Tuples in the Relational Data Model

Data model

Disjunction is modeled by (primary) key violations.

Example (Keys are underlined)

T

Talk	*Speaker*	*Session*
6	Jef	10A
6	Jef	11
7	Mickael	10A

S

Session	*Chair*	*Day*
2A	Victor	Wed.
10A	Luc	Fri.
10A	Sven	Fri.
11	Claire	Fri.

- Talk 6 will be in **either** session 10A **or** session 11.
- Session 10A will be chaired by **either** Luc **or** Sven.

Definition (Block)

A block is a maximal set of tuples of the same relation with the same value for the key. (Blocks are separated by dashed lines.)

Disjunctive Tuples in the Relational Data Model

Data model

Disjunction is modeled by (primary) key violations.

Example (Keys are underlined)

T

Talk	*Speaker*	*Session*
6	Jef	10A
6	Jef	11
7	Mickael	10A

S

Session	*Chair*	*Day*
2A	Victor	Wed.
10A	Luc	Fri.
10A	Sven	Fri.
11	Claire	Fri.

- Talk 6 will be in **either** session 10A **or** session 11.
- Session 10A will be chaired by **either** Luc **or** Sven.

Definition (Block)

A block is a maximal set of tuples of the same relation with the same value for the key. (Blocks are separated by dashed lines.)

Certainty Semantics

Definition (Repair and Certainty)

A repair is obtained by selecting exactly one tuple from each block.
A Boolean query is certain if it is true in all repairs.

Example

T

Talk	*Speaker*	*Session*
6	Jef	10A
6	Jef	11
7	Mickael	10A

S

Session	*Chair*	*Day*
2A	Victor	Wed.
10A	Luc	Fri.
10A	Sven	Fri.
11	Claire	Fri.

Will talk 6 take place on Friday?

$\exists s \exists u \exists w \, (T(\underline{\text{'6'}}, u, s) \wedge S(\underline{s}, w, \text{'Fri.'}))$ is **certain**.

Will talk 6 be chaired by Claire?

$\exists s \exists u \exists w \, (T(\underline{\text{'6'}}, u, s) \wedge S(\underline{s}, \text{'Claire'}, w))$ is **not certain**.

Certainty Semantics

Definition (Repair and Certainty)

A repair is obtained by selecting exactly one tuple from each block.
A Boolean query is certain if it is true in all repairs.

Example

T

Talk	*Speaker*	*Session*
6	Jef	10A
6	Jef	11
7	Mickael	10A

S

Session	*Chair*	*Day*
2A	Victor	Wed.
10A	Luc	Fri.
10A	Sven	Fri.
11	Claire	Fri.

Will talk 6 take place on Friday?

$\exists s \exists u \exists w \, (T(\underline{\text{'6'}}, u, s) \wedge S(\underline{s}, w, \text{'Fri.'}))$ is **certain**.

Will talk 6 be chaired by Claire?

$\exists s \exists u \exists w \, (T(\underline{\text{'6'}}, u, s) \wedge S(\underline{s}, \text{'Claire'}, w))$ is **not certain**.

What is the Complexity of Certainty Semantics?

Definition

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

Input A database instance (possibly with key violations)

Question Is q certain?

Complexity Classification Task

Input A Boolean first-order query q

Question What complexity classes does CERTAINTY(q) belong to?

Complexity classes of interest:

$$\mathbf{FO} \subseteq \mathbf{P} \subseteq \mathbf{coNP}$$

- Complexity in **FO** is of interest to database practitioners, because it allows for implementation in SQL.

What is the Complexity of Certainty Semantics?

Definition

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

Input A database instance (possibly with key violations)

Question Is q certain?

Complexity Classification Task

Input A Boolean first-order query q

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

$$\mathbf{FO} \subseteq \mathbf{P} \subseteq \mathbf{coNP}$$

- Complexity in **FO** is of interest to database practitioners, because it allows for implementation in SQL.

Main Result

We solved the aforementioned complexity classification task when the input queries q are conjunctive and self-join-free (i.e., no relation name occurs more than once in q):

Theorem (Complexity Classification)

For every self-join-free Boolean conjunctive query q, the following hold:

1. CERTAINTY(q) *is either in* **P** *or* **coNP**-*complete (and the dichotomy is decidable);*
2. *it can be decided whether* CERTAINTY(q) *is in* **FO**; *and*
3. *if* CERTAINTY(q) *is in* **FO**, *then its first-order definition can be computed effectively.*

- The theorem settles a conjecture that had been open for 10 years.

Main Result

We solved the aforementioned complexity classification task when the input queries q are conjunctive and self-join-free (i.e., no relation name occurs more than once in q):

Theorem (Complexity Classification)

For every self-join-free Boolean conjunctive query q, the following hold:

1. CERTAINTY(q) *is either in* **P** *or* **coNP**-*complete (and the dichotomy is decidable);*
2. *it can be decided whether* CERTAINTY(q) *is in* **FO**; *and*
3. *if* CERTAINTY(q) *is in* **FO**, *then its first-order definition can be computed effectively.*

- The theorem settles a conjecture that had been open for 10 years.

Examples of Different Complexities

Example

$$
\begin{aligned}
q_1 &= \exists s \exists u \exists w \left(T(\underline{\text{'6'}}, u, s) \wedge S(\underline{s}, w, \text{'Fri.'}) \right) \\
q_2 &= \exists x \exists y \left(T(\underline{x}, x, y) \wedge S(\underline{y}, y, x) \right)^{a} \\
q_3 &= \exists s \exists t \exists u \exists w \exists x \left(T(\underline{t}, x, u) \wedge S(\underline{s}, x, w) \right)^{b}
\end{aligned}
$$

Our results allow us to tell that

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

---

[a] A meaningless query for our example database.

[b] "Does some session chair also give a talk?"

Construction of a First-Order Definition

Example

$$q_1 \quad = \quad \exists s \exists u \exists w \, (T(\underline{\text{`6'}}, u, s) \wedge S(\underline{s}, w, \text{`Fri.'}))$$

A first-order definition of CERTAINTY(q_1) is as follows:

$$\begin{aligned} \varphi_1 = \exists s \exists u (& T(\underline{\text{`6'}}, u, s) \wedge \\ & \forall s \forall u ( T(\underline{\text{`6'}}, u, s) \rightarrow \exists w \exists z (S(\underline{s}, w, z) \wedge \\ & \qquad \forall w \forall z (S(\underline{s}, w, z) \rightarrow z = \text{`Fri.'}))))\end{aligned}$$

Thus,

$$q_1 \text{ is certain} \iff \varphi_1 \text{ is true}$$

Open Problems

Conjecture

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

Conjecture

For every query q that is a finite disjunction of Boolean conjunctive queries, CERTAINTY(q) *is in* **P** *or* **coNP***-complete.*

Caveat

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).

Open Problems

Conjecture

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

Conjecture

For every query q that is a finite disjunction of Boolean conjunctive queries, CERTAINTY(q) *is in* **P** *or* **coNP***-complete.*

Caveat

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).

References

Andrei A. Bulatov.

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

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.

Paraschos Koutris and Jef Wijsen.

Consistent query answering for primary keys.
SIGMOD Record, 45(1):15–22, 2016.