Data Wrangling - Reading
On the Representation and Querying of Sets of Possible Worlds

Théo Delemazure

February 2020

Introduction

This short report discuss the journal paper *"On the Representation and Querying of Sets of Possible Worlds"* by **Serge Abiteboul** (*Inria, France*), **Paris Kanellakis** (*Brown University, USA* at the time), and **Gosta Grahne** (*University of Helsinki, Finland* at the time) in 1991 on *Theoretical computer Science*. The first version of this paper was published in 1987 at the SIGMOD conference.

This paper was published during the early years of the research in database theory and it was quoted hundreds of time since its first publication. The research was led at the Inria laboratory, in which **Serge Abiteboul** is a researcher.

Serge Abiteboul is an influential researcher in the database community and he is mostly known for publishing the book *Fundations of databases* with **V. Vianu** and **R. Hull**, a book which is still a reference for theoretical research on databases. **Paris Kanellakis** also worked on this book (according to *google scholar*) but he unfortunately died in 1995, year of publication of the book. He was a really important researcher in theoretical computer science and the ACM created in 1996 the *Paris Kenallis award* to reward research in theoretical computer science with an important practical impact.

In the first section, I will give details on the content of this paper. In the second section, I will quickly discuss interests and limitations of this work. Finally, in a third section, we will discuss the research which followed this work on incomplete databases.

1 Description and content of the paper

This paper define some problems that can be expressed on databases with missing data, i.e. we can define a set of *possible world* which is not restricted to a unique instance. The authors show the theoretical complexity of each problem for various restriction on the parameters and the inputs.

In this section, I will quickly explain the different problems defined by the author, present their results, and how they proved them.

1.1 Representation of possible worlds

To represent missing information in a theoretical way, the authors introduce different family of *relational table* with increasing levels of complexity. We denote $\mathcal{I}$ the set of possible world associated to some table. The different kind of tables are the following:

- instance : In an `instance`, there is no missing information, the set of possible world $\mathcal{I} = \{I\}$ is a singleton. See an example Table 1.
- table : In a `table`, some constants are replaced by variables, but there is **no relations** between variables. See an example Table 2.

- e-table : An `e-table` is a `table` in which some variables **are equal** (i.e. variables can represent different elements of the table). See an example Table 3.
- i-table : An `i-table` is a `table` in which some variables **must have different values**. See an example Table 4.
- g-table : A `g-table` is a `table` in which some variables must have different values and some variables represent different elements of the table. See an example Table 5.
- c-table : A `c-table` is a `g-table` in which a conjunction of equalities and inequalities between variables is associated to each row. A row appears on the possible world **if the condition attached to it is True in this possible world**. See an example Table 6.

a	b	c
3	2	1
2	0	4
1	1	2

Table 1: `instance`

a	b	c
3	x_1	1
x_2	0	x_3
1	1	x_4

Table 2: `table`

a	b	c
3	x_1	1
x_1	0	x_2
1	1	x_3

Table 3: `e-table`

a	b	c
$x_1 \neq x_3$		
3	x_1	1
x_2	0	x_3
1	1	x_4

Table 4: `i-table`

a	b	c
$x_1 \neq x_2$		
3	x_1	1
x_1	0	x_2
1	1	x_3

Table 5: `g-table`

a	b	c	
$x_1 \neq x_3$, $x_2 \neq x_3$			
3	x_1	1	$x_4 = x_4$
x_1	0	x_2	$x_1 \neq x_2$
1	1	x_3	$x_3 \neq 0$

Table 6: `c-table`

We easily see that :

$$\texttt{instance} \subset \texttt{table} \subset \texttt{e-table} \neq \texttt{i-table} \subset \texttt{g-table} \subset \texttt{c-table}$$

The authors also add another level of complexity with `views`. By the mean of query q, one can obtain a set of possible worlds $\mathcal{I}_{out}$ from another set $\mathcal{I}_{in}$ associated to some table :

$$\mathcal{I}_{out} = \{\ q(I) \mid I \in \mathcal{I}_{in}\ \}$$

For that, they introduce different query families, that all have **PTIME data complexity**:

- Positive existential queries $\mathcal{Q}_{\exists}$ are queries using the relational algebra operators σ^+, Π, ρ, $\bowtie$ and $\cup$ where σ^+ is a *positive select*, i.e. without $\neq$ in the selection condition.
- First order logic (FOL) queries use all the above operators and **allow inequality** in selection condition.
- DATALOG queries use all the above operators (without $\neq$) and **allow recursion**.

1.2 Definition of the problems

The authors then define various problems on incomplete tables and sets of possible worlds. Due to the number of problems considered, this fundamental paper **gives a lot of important theoretical results** on incomplete databases. For every problem, we apply a query q to each set of possible worlds and these queries are *parameters of the problem.*

- The containment problem ($\textsc{Cont}(q_1, q_2)$) : *Is a set of possible worlds $q_1(\mathcal{I}^1)$ a subset of a second set of possible worlds $q_2(\mathcal{I}^2)$?* More formally, we check $q_1(\mathcal{I}^1) \subseteq q_2(\mathcal{I}^2)$.

- The membership problem (MEMB(q)) : *Is a given instance I an element of a set of possible world $q(\mathcal{I})$?* More formally, we check $I \in q(\mathcal{I})$.
- The uniqueness problem (UNIQ(q)) : *Is a set of possible world $q(\mathcal{I})$ equal to a singleton $\{I\}$?* More formally, we check $\{I\} = q(\mathcal{I})$.
- The possibility problem (POSSIB(q, k)) : We want to know if *a given set of facts $t_1, ..., t_k$ is a possible answer*, i.e. $\exists I \in q(\mathcal{I}), s.t. \forall i, t_i \in I$. We denote $k = *$ if the number of facts can be any natural number.
- The certainty problem (CERT(q, k)) : We want to know if *a given set of facts $t_1, ..., t_k$ is a necessary answer*, i.e. $\forall I \in q(\mathcal{I}), \forall i, t_i \in I$. We denote $k = *$ if the number of facts can be any natural number.

1.3 Theorems and Proofs

The rest of the paper consists in a succession of theorems and proofs on theoretical complexities of the problems defined above. As this is the case for a lot of theoretical papers with complexity results, there are two possibilities for each problem $\mathcal{P}$:

- The problem $\mathcal{P}$ is in PTIME and the method used to prove it is to either **reduce the problem** to a known PTIME problem, or **directly give an algorithm** which solve $\mathcal{P}$.
- The problem is NP-complete, coNP-complete or even higher in the polynomial hierarchy, like in our case, Π_2^p-COMPLETE. There is two steps to prove it. For instance, if it is NP-complete:
  1. $\mathcal{P} \in$ NP, which is not the hardest part nor the most interesting one in most cases.
  2. $\mathcal{P}$ is NP-hard. Researchers use a known NP-complete problem and reduce it to $\mathcal{P}$ (with a polynomial reduction). In the theoretical database field, the NP-complete problem used is often the *Graph 3-coloring* problem. In this paper, other problems are frequently used like *3DNF tautology* or $\forall 3DNF$ (coNP-completeness), *3CNF satisfiability* or $\exists 3CNF$ (NP-completeness) and $\forall\exists 3CNF$ (Π_2^p-completeness).

Complexities for each problem and each particular restriction on inputs and parameters are summed up in the Annex (Tables 7 to 11). A thing I like a lot about this paper is that **they try to find the precise complexity (with higher and lower bound) for as much cases as possible**.

They first quickly show an upper bound for each problem, and then they consider each problem individually. For each one, they **find smaller upper bound** for some particular cases, and **show completeness** for other cases. A very impressive thing in this paper is that the authors try to find the exact complexity **for as much cases as possible**.

Membership and Uniqueness (Tables 7, 8) They start with the *membership and uniqueness problems* for which proof of completeness or PTIME algorithms are not too hard. They prove the exact complexity for every possible case of the *membership problem* (see Table 7). The only case which is PTIME is when the query is **the identity** and the input is a `table`.

> To prove it, they show an intuitive reduction to the *bipartite graph matching problem* in which one try to **link every row of the table to a row of the instance**.

For every other cases, they find a reduction of the *graph 3-coloring problem*. Indeed, as soon we introduce relations between variables($=$ or $\neq$), we can "implement" the condition that *no two adjacent nodes of a colored graph have the same color*.

The theorem for the *uniqueness problem* does not cover every possible case (see Table 8), but it shows **a perfect dichotomy in the case** $q = Id$. They use different coNP-complete problems for the two hardness results.

> They show a reduction of the *3DNF tautology problem* for `c-table` which is really beautiful: if you match each clause of the 3DNF formula Ψ to a row of the `c-table`, then Ψ is not a tautology **if and only if** there is an instance of Ψ variables such that every clause is FALSE, i.e. no row appears in the possible world associated to this instance.

Containment (Table 9) As it takes two inputs and two parameters, the *containment problem* complexity is explored in depth and for even more cases than other problems (see Table 9). Indeed, for some inputs, the problem is in PTIME, for some other cases, the problem is in NP or coNP (they use the *3DNF tautology problem* for hardness results), and in most cases, the problem is in Π_2^p (they use the $\forall\exists 3CNF$ *problem* for hardness results).

Proofs of hardness become really complicated at this point, even if they are all pretty well explained and illustrated with tables and examples.

Possibility and Certainty (Tables 10, 11) These two problems are probably the problems of this paper which **were the most used in following research**. Intuitively, the *possibility problem* is NP because one has to find the possible world **in which all facts are True** (they use a reduction of the $3CNF$ *satisfiability problem* for hardness), and the *certainty problem* is $coNP$ because you have to find the possible world **in which all facts are False** (they use a reduction of the $3DNF$ *tautology problem* for hardness).

Every single case is covered for the *possibility problem* :

- If k is not specified, then the problem is PTIME **if and only if** $q = Id$ and the input is a `table`. It is NP-complete otherwise.
- If k is specified, then the problem is PTIME **if and only if** q is a positive existential query. It is NP-complete otherwise.

For the *certainty problem*, they complete the work started in [3, 9] to obtain a perfect dichotomy:

- The problem is PTIME **if and only if** the input is at most a `g-table` and the query can be expressed in $DATALOG$. It is coNP-complete otherwise.

2 Interest and limitations

First of all, the paper is a **perfectly structured theoretical paper**: After a short introduction, the authors define the formalism used, then they look at each problem one by one and try to cover as most cases as possible. The proofs are beautiful, even if some are a bit complicated, and they contain examples which enable to understand them more easily.

The work here is purely theoretical, and there is no consideration of how to deal with incomplete database in practice. In particular, *what is the link between* `null` *values and variables of a* `table`*? How can we represent certain and possible answer in a practical query language? How well are actually performing PTIME algorithm? Can we use a SAT-solver to efficiently solve NP or coNP problem?* Anyway, this is not what theoretical computer science papers are about (and SAT-solvers were not as efficient as today in 1991).

3 Further work

As it is one of the first papers to talk about *possible worlds* and how to deal with *null values*, it is a reference in incomplete database research and is quoted in various surveys and general books on the subject [1, 6, 7, 4, 2], even if **they all focus around more recent theoretical results and practical algorithms.** In the literature, this definition of incomplete databases is often **compared to probabilistic databases** as they share the notion of *possible world*. It also led to a lot of research on **certain and possible answers**, sometimes extended to *ranking* or *voting* with incomplete data [8, 5].

As a conclusion, this paper from 1991 open the way to the research on databases with missing information and *null* values. It is also a paper with really nice theoretical results.

References

[1] Aggarwal, C. C., and Yu, P. S. A survey of uncertain data algorithms and applications. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING 21*, 5 (2009), 609–623.

[2] Atzeni, P., and De Antonellis, V. *Relational database theory*. Benjamin/Cummings Redwood City, CA, 1993.

[3] Imieliundefinedski, T., and Lipski, W. Incomplete information in relational databases. *J. ACM 31*, 4 (Sept. 1984), 761–791.

[4] Kanellakis, P. C. Elements of relational database theory. In *Formal models and semantics*. Elsevier, 1990, pp. 1073–1156.

[5] Kimelfeld, B., Kolaitis, P. G., and Stoyanovich, J. Computational social choice meets databases. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18* (7 2018), International Joint Conferences on Artificial Intelligence Organization, pp. 317–323.

[6] Li, Y., Chen, J., and Feng, L. Dealing with uncertainty: A survey of theories and practices. *IEEE Transactions on Knowledge and Data Engineering 25*, 11 (2012), 2463–2482.

[7] Parsons, S. Current approaches to handling imperfect information in data and knowledge bases. *IEEE Transactions on knowledge and data engineering 8*, 3 (1996), 353–372.

[8] Soliman, M. A., and Ilyas, I. F. Ranking with uncertain scores. In *2009 IEEE 25th International Conference on Data Engineering* (2009), IEEE, pp. 317–328.

[9] Vardi, M. Y. Querying logical databases. *Journal of Computer and System Sciences 33*, 2 (1986), 142 – 160.

Annex

Query	Input	Complexity	Proof
$q = Id$	$\leq$ `table`	PTIME	Reduc. to *bipartite graph matching*
$q = Id$	$\geq$ `e-table`	NP-COMPLETE	Reduc. of *graph 3-coloriability*
$q = Id$	$\geq$ `i-table`	NP-COMPLETE	Reduc. of *graph 3-coloriability*
$q \in \mathcal{Q}_\exists^+$	$\geq$ `table`	NP-COMPLETE	Reduc. of *graph 3-coloriability*

Table 7: Complexities of *Membership* (NP)

Query	Input	Complexity	Proof
$q = Id$	$\leq$ `g-table`	PTIME	Check if all variables can be replaced by a constant
$q = Id$	$\geq$ `c-table`	coNP-COMPLETE	Reduc. of *3DNF tautology*
$q \in \mathcal{Q}_\exists^+$	$\leq$ `e-table`	PTIME	Shown an algorithm
$q \in FOL$	$\geq$ `table`	coNP-COMPLETE	Reduc. of *graph non 3-colorability*

Table 8: Complexities of *Uniqueness* (coNP)

Left query	Right query	Left input	Right input	Complexity	Proof
$q_l = Id$	$q_r = Id$	$\leq$ `g-table`	$\leq$ `table`	PITME	
$q_l = Id$	$q_r = Id$	$\geq$ `table`	$\geq$ `i-table`	Π_2^p-COMPLETE	Reduc. of $\forall\exists 3CNF$
$q_l = Id$	$q_r = Id$	$\geq$ `c-table`	$\geq$ `e-table`	Π_2^p-COMPLETE	Reduc. of $\forall\exists 3CNF$
$q_l = Id$	$q_r = Id$	$\leq$ `g-table`	$\leq$ `e-table`	NP	
$q_l = Id$	$q_r \in \mathcal{Q}_\exists^+$	$\geq$ `table`	$\geq$ `table`	Π_2^p-COMPLETE	Reduc. of $\forall\exists 3CNF$
$q_l \in \mathcal{Q}_\exists^+$	$q_r = Id$	$\geq$ `table`	$\geq$ `table`	coNP-COMPLETE	Reduc. of 3DNF tautology
$q_l \in \mathcal{Q}_\exists^+$	$q_r = Id$	$\leq$ `g-table`	$\leq$ `table`	coNP	
$q_l \in \mathcal{Q}_\exists^+$	$q_r = Id$	$\geq$ `table`	$\geq$ `e-table`	Π_2^p-COMPLETE	Reduc. of $\forall\exists 3CNF$

Table 9: Complexities of *Containment* (Π_2^p)

Query	#facts	Input	Complexity	Proof
$q = Id$	*	$\leq$ `table`	PTIME	
$q = Id$	*	$\geq$ `e-table`	NP-COMPLETE	Reduc. of *3CNF sat*
$q = Id$	*	$\geq$ `i-table`	NP-COMPLETE	Reduc. of *3CNF sat*
$q \in \mathcal{Q}_\exists^+$	*	$\geq$ `table`	NP-COMPLETE	Reduc. of *graph 3-coloriability*
$q \in \mathcal{Q}_\exists^+$	k	$\leq$ `c-table`	PTIME	Transform the table
$q \in DATALOG$	k	$\geq$ `table`	NP-COMPLETE	Reduc. of *3CNF sat*
$q \in FOL$	k	$\geq$ `table`	NP-COMPLETE	Reduc. of *3DNF nontautology*

Table 10: Complexities of *Possibility* (NP)

Query	**#facts**	**Input**	**Complexity**	**Proof**
$q \in DATALOG$	$*$	$\leq$ `g-table`	PTIME	Known results
$q \in FOL$	k	$\geq$ `table`	coNP-complete	Reduc. of *3DNF tautology*
$q = Id$	k	$\geq$ `c-table`	coNP-complete	Reduc. of *3DNF tautology*

Table 11: Complexities of *Certainty* (coNP)