QUID: Efficient Querying for Incomplete and Inconsistent Data

Acronym	QUID
Title of the project	Efficient Querying for Incomplete and Inconsistent Data
Funding	Projet de recherche collaborative
Type of research	Recherche fondamentale
Défi 7 Axe 4	Société de l'information et de la communication Données, Connaissances, Big Data, Contenus multimédias - Intelligence Artificielle
Duration of the project	48 months

Contents

Summary

Data management systems nowadays need to handle large data sets often containing inconsistent or incomplete information, and yet provide meaningful answers while guaranteeing data privacy requirements. This project aims at advancing the foundations of query answering in the presence of incomplete or inconsistent data. For various query languages and data integrity constraints, we will develop new techniques for making access control and privacy more reliable and develop new tools for making data querying more efficient. These developments hinge upon the key notion of *certain answers* of a query: answers found in all possible plausible interpretations of an inconsistent

or incomplete data set. A special case of incomplete data arises from concealing information to certain users, with the purpose of preserving data privacy. The question of whether this concealment indeed leaks private data is then a delicate matter, known as the *determinacy problem*, and it is still not well understood.

Our research program focuses on the problems of certain answers and determinacy in relation to three scenarios: ensuring data privacy, querying inconsistent data, and querying incomplete data. We will develop formal methods for analysing data management protocols as well as efficient algorithms for computing relevant answers. These new methods will make access control and privacy more reliable, and they will enable the exploitation of data containing inconsistencies or incomplete information. Further, we will investigate formal connections with certain Constraint Satisfaction Problems (CSP), which shall bring new insights both to the database and CSP communities.

Project QUID is a 48-month collaborative project involving research groups working on different areas of database theory. The consortium brings together 8 researchers from 4 laboratories: Laboratoire d'Informatique Gaspard-Monge (LIGM, Université Marne-la-Vallée), Institut de Recherche en Informatique Fondamentale (IRIF, Paris Diderot), École Normale Supérieure de Paris (ENS Ulm) and Laboratoire Bordelais de Recherche en Informatique (LaBRI, Université de Bordeaux). It will also benefit from existing international collaborations. The project also includes the hiring of a PhD student who will be jointly supervised between LaBRI and IRIF, and two 1-year post-doctoral fellows.

Partner	**Name**	**Last name**	**Position**	**p.m.**	**Responsibility**
LIGM	Claire Nadime	David Francis	MCF MCF		Project coordinator
IRIF	Cristina Amélie	Sirangelo Gheerbrant	PR MCF		Local coordinator
ENS Ulm	Luc Pierre	Segoufin Senellart	DR Inria PR		Local coordinator
LaBRI	Diego Gabriele	Figueira Puppis	CR CNRS CR CNRS		Local coordinator

1 Context, positioning and objectives

1.1 Objectives and scientific hypotheses

Data management systems nowadays need to cope with large data sets, often integrated from many heterogeneous sources, containing redundant, inconsistent and incomplete data. Moreover, data is often not available in its whole, due to prohibitively large data volumes or access restrictions. In this scenario, data management becomes error-prone and vulnerable to data leakage.

At the same time, we witness an increasing need for reliable and efficient systems, providing more privacy, more security, and more relevant answers to user queries.

Our proposal aims at narrowing the gap between current capabilities of data management systems on the one hand, and user requirements on the other hand. This will be done by developing formal methods for analysing data management protocols as well as efficient algorithms for computing relevant answers. These new methods will make access control and privacy more reliable, and they will enable the exploitation of data containing inconsistencies or incomplete information.

Our research program is targeted towards three different but related scenarios. In the first scenario we are interested in querying data views, and analysing the information they may leak on the original data, impacting privacy and data leakage issues. The second one concerns repairs of inconsistent data sets, with a direct impact on efficient algorithms for retrieving relevant answers. In the third scenario we deal with the presence of missing data, and aim at providing effective methods to answer user queries efficiently.

Despite the specificity of each of these scenarios, they have one main aspect in common. In each of them one lacks full information on the data that needs to be queried, but the system still needs to answer user queries with certainty, and provide security guarantees. The *certainty* aspect is the main technical difficulty in finding good solutions and this has been extensively studied in the literature, e.g. [GLS14; Lib15; Wij12]. Despite all the attention attracted, effective and efficient methods are still missing in all three scenarios above, and the development of these methods are the main goal of this proposal.

Moreover, recent studies [FSS15; Fon15; LW15] have independently shown evidence that certainty in query answering is intimately related to *Constraint Satisfaction Problems* (CSP), a very active area of research at the frontier of mathematics and graph theory. So far, these relationships with CSP have been shown in an *ad-hoc* way, each tailored to a particular problem. However, we think that there is ground for a unified approach which can also feedback the CSP area with novel interesting questions. This constitutes the most prospective goal of this proposal.

We remark that answering user queries with certainty is also the main focus of ontology-based query processing [Cal+11], mainly studied in artificial intelligence, which has also been shown to be related to CSP [Bie+14]. Although we do not exclude potential relationships with our approach, ontologies are not in the scope of this proposal.

We now present our research plan in more details, starting with the description of our three scenarios, followed by the prospective connection with CSP.

1.1.1 Determinacy, privacy and data leakage

The problem of determinacy arises in a classical scenario for database management systems. In this scenario the system presents a set of *views* of the data set to the users. These views are defined as answers to queries $Q_1, \ldots, Q_k$. Each user only has access to the information of a database D provided by the views $V_1 = Q_1(D), \ldots, V_k = Q_k(D)$. The idea is that the views return data to which the user should have access, while filtering out all private data that should remain hidden. Let Q be a query that would return sensitive information, that is, a query whose results over D specifically contains the private data that should be kept from the user. Then it should be ensured that the user cannot compute $Q(D)$ from the views $V_1, \ldots, V_k$ at her disposal; otherwise, the views leak private data and thus are not secure.

In fact while views are in general lossy, in that they lose some information contained in the original data, they may still provide enough information to answer *some* queries. One wants to make sure that the private queries of interest are not in such disclosed queries.

For instance on a network a view could only provide information about network nodes connected by many hops (say 100 hops) which hides information about the actual connections. Queries asking for nodes connected by a multiple of 100 hops can still be answered only using the view.

Consider a user with the views $V_1 = Q_1(D), \ldots, V_k = Q_k(D)$ of the database D at her disposal. In theory she can always compute an underapproximation of $Q(D)$ by computing the answers to Q that can be found in *all* databases whose view is what she currently sees. In other words, the

views $V_1, \ldots, V_k$ allow her to compute:

$$\bigcap_{D', \forall i\, Q_i(D')=V_i} Q(D')$$

This is known as the *certain answers* to Q. When the certain answers to Q using $V_1, \ldots V_k$ coincide with the real answers $Q(D)$ then the user can use her view to compute the answers to Q. If Q would return private information, then the views leak that information, certainly an undesirable feature.

There are many approaches to check whether the views disclose undesired information. One of them is to check the absence of leakage of private data at run time, by computing some "knowledge" of each user from the query answers she was already given. Another possibility is to perform an analysis of the logs of the system and try to determine whether some leakage has occurred. We will focus on a third approach which is the static analysis version of the problem, that is performed offline and beforehand.

In database theory this problem is known as *the determinacy problem* [NSV10]: given a specification of the view in some language (e.g., as the answer to a conjunctive query) and a target query, one has to decide whether the user's view always carries enough information to evaluate the query, no matter the content of the database. In other words, we want to test whether for all databases D and D' whenever $Q_i(D) = Q_i(D')$ for every view Q_i, we have $Q(D) = Q(D')$. This is equivalent to saying that there exists another query R that, evaluated on $Q_1(D), \ldots Q_k(D)$ returns $Q(D)$, no matter D.

For instance in the example above, the 100-hops view determines the 500-hops query, in fact it suffices to issue a 5-hops query R on the view to know which nodes are connected by 500 hops in the original network.

However it is not always so simple to figure out determinacy. Consider for instance a view V_1 providing the pairs of network nodes connected by 3 hops —which can be specified by a conjunctive query $Q_3(x, y) := \exists z_1 z_2\ E(x, z_1) \wedge E(z_1, z_2) \wedge E(z_2, y)$—, a view V_2 providing the pairs of nodes connected by 4 hops —specified by a conjunctive query $Q_4(x, y) := \exists z_1 z_2 z_3\ E(x, z_1) \wedge E(z_1, z_2) \wedge E(z_2, z_3) \wedge E(z_3, y)$— , and a query Q asking for the pairs or nodes connected by 5 hops. Then one can show that V_1 and V_2 determine Q. Indeed, one can prove that, for every database D, $Q(D)$ can be computed from the views $V_1 = Q_3(D)$ and $V_2 = Q_4(D)$ by issuing the following first-order query $\exists z\ V_2(x, z) \wedge (\forall z'\ V_1(z', z) \rightarrow V_2(z', y))$. This shows that the question is not trivial, albeit conjunctive queries – corresponding to the existential positive fragment of first order logic – are the most basic and commonly used queries on classical relational databases.

There remain various open questions around the determinacy problem, as we detail next. We aim at solving them for some of the most natural query and view languages.

Deciding determinacy In general, already for conjunctive queries and conjunctive views, the determinacy problem is undecidable [GM15]. However there exist tractable cases [Afr11; FSS15], as well as simple fragments for which the question is open.

The picture is still wide open for queries and views that have access to a form of recursion. One case of high interest is that of *regular path queries*, which are the most basic and widespread form of queries over graph databases. These queries ask for pairs of nodes in a labelled graph connected via a path whose sequence of labels belong to a specified regular language.

It is still unknown whether the determinacy problem for regular path queries is decidable [Fra15]

We aim at finding out the decidability status of the determinacy problem for most common fragments of query and view languages.

Members of this proposal based in Marne-la-Vallée and ENS Ulm have already progressed on the identification of decidable fragments. [NSV10; Fra17].

Efficient determinacy While the problem of determinacy asks whether the set of views carry enough information to compute a query, it does not say *how* the query can be computed over such views. In other words determinacy only points out *when* it is possible to infer private data from the view, that is, when there exists a new query R over the views that returns private data. It does not say how R can be expressed, nor how R can be found.

When it exists, such a query is called a *rewriting* because it reformulates the original query on a new vocabulary: the one associated to the views. In our examples above, for a 100-hops view, and a 500-hops query there exists a simple conjunctive rewriting (the 5-hops query). With 3-hops and 4-hops views, the 5-hops query can be rewritten on the views as the simple first-order formula presented above. In both cases the rewriting can be computed efficiently on the views, with the current database systems.

Unfortunately this is not always so simple. Indeed, even if the views leak enough information for computing a sensitive query, the system may still be regarded as safe if actually answering this query requires too much computational power to be feasible. We believe that in fact the right question to be asked in this scenario is whether the query can be *efficiently* computed from the views. This is a setting that we aim to develop with this project, which thus far has not been explored. We measure efficiency in terms of *data complexity*, i.e., we consider the answers to the views as input parameters, while the sizes of queries are regarded as constants. The *efficient determinacy problem* then asks, given a specification of the views and a target query, whether a user may be able to evaluate the query efficiently (say in polynomial time) using only her views, no matter the content of the database.

For conjunctive queries and views, as well as for regular path queries and views, when the views determine the query then the user can always evaluate the query using a rewriting with NP $\cap$ co-NP data complexity [NSV10]. However, we don't know whether this bound can be improved. Most importantly we do not know whether a PTIME rewriting can always be found. The only known lower bounds come from the existence of views and queries requiring a computational power beyond AC^0 [GM16]. The gap is considerable, which calls for investigation.

> We will study the efficient determinacy problem for various query and view specification languages.

- The first approach in this direction will be to fix a target PTIME language for rewritings and ask when this language is sufficient.

  Members of this proposal have already started addressing the the case of rewritings expressible in Datalog, a query language corresponding to the existential positive fragment of fixed point logic. In fact Datalog is a very natural language for expressing queries with a form of recursion over graphs, and Datalog queries can be evaluated in PTIME in the size of the view.

  It turns out that Datalog is a sufficiently powerful language for monotone rewritings for regular path queries and views: whenever a monotone rewriting exists, it is expressible in Datalog. This is the main result obtained during Nadime Francis's PhD [FSS15], which settles an open question.

  However there is currently no evidence that the full expressive power of Datalog is needed to express monotone rewritings, we will then look for tighter upper bounds inside Datalog.

- We will then investigate the possibility of rewriting queries over views using non-monotone languages. In fact although monotone query languages seem to be natural for expressing structural properties of the database (e.g. the existence of paths, cycles or patterns in a graph), we know that (rather surprisingly) it may be required to use a non-monotone query language to express rewritings, even if the rewritten query is itself monotone [Afr11].

  Languages we should look at need of course to extend the ones needed for monotone rewritings. For regular path queries and views there are several natural candidates: several variants of Datalog with negation, least fixed-point logic or even first order logic with transitive closure.

  However non-monotone rewritings have been largely unexplored, and the few cases that have been studied have proved to be technically much more difficult. Even if views and queries contain no recursion at all, it is still an open question to know whether a first order rewriting can always be found, whenever the query is determined by the views.

1.1.2 Repairs of inconsistent databases

In this scenario the data set may have inconsistencies relative to integrity constraints (e.g., key and foreign key constraints, inclusion constraints). For instance the data set may be a binary relation E relating employees to supervisors. An instance of E may indicate multiple supervisors per employee while the integrity constraints state for instance that each employee has only one supervisor. This corresponds to the very common case of *key constraints*.

To cope with such situations a general notion of *database repair* has been defined which intuitively consists in a database satisfying the integrity constraints and whose *distance* from the original data set is *minimal*. By tuning the definitions of distance and of minimality, one obtains several concrete definitions of what a repair should be, whose relevance depends on the context, and in particular on the set of integrity constraints [ABC03].

For instance, when the integrity constraints are only key dependencies, the commonly accepted notion of repair keeps only one tuple per key value. It is important to note that an inconsistent database may have several repairs. In our example, each repair would "choose" only one supervisor per employee. Therefore there are exponentially many repairs, where the exponent is the number of employees having multiple supervisors in the inconsistent dataset.

As many repairs may exist for a given database, it is not clear what the answers to a query should be. The widely accepted notion is again the one of *certain answers*, i.e. answers that can be found in **all** repairs. The challenge is then to provide efficient algorithms for computing the certain answers directly from the inconsistent database, as going through all possible repairs is clearly not an efficient strategy. This problem is usually studied from the data complexity point of view. For each query, the associated problem takes as input an inconsistent database and asks for the certain answers. When the query is Boolean, this becomes the decision problem of whether the input query is true on all the repairs of the database.

Consider for instance the query asking whether there is an employee having at least two levels of hierarchy above his. If the database is consistent then this is a distance 2 query that can be expressed with the conjunctive query $Q_2 := \exists x, y, z\ E(x, y) \wedge E(y, z)$. In the presence of inconsistency, testing that all repairs satisfy Q_2 requires a slightly more complicated formulation, but it is still expressible in first order logic: $\exists x, y, z\ E(x, y) \wedge E(y, z) \wedge \big(\forall y'\ E(x, y') \rightarrow \exists z'\ E(y', z')\big)$.

The complexity of the problem of computing certain answers is illustrated in the following two examples. Assume E and F are binary relations and the integrity constraints require that for both relation the first attribute is a key, as in our initial example.

As above we can see that the certain answers to the query $\exists x, y, z\ E(x, y) \wedge F(y, z)$ can be expressed in first-order logic and therefore computed efficiently. However for a simple modification

of the query, $\exists x, y\ E(x, y) \wedge F(y, x)$, it can be shown that there is no first-order formulation of its certain answers [Wij10]. Even worse, the certain answers to the query $\exists x, y, z\ E(x, y) \wedge E(z, y)$ are coNP hard to compute [FM07].

For key dependencies and conjunctive queries, the problem of computing the certain answers is always in coNP (for more involved integrity constraints, the problem associated to some conjunctive queries could be even more complicated), it is coNP-complete for some conjunctive queries and polynomial for some others.

Dichotomy conjecture When the query is conjunctive and the integrity constraints are functional dependencies, a dichotomy conjecture states that computing certain answers can either be done in polynomial time or is coNP-complete. This conjecture has only been solved for self-join free queries [KW15].

> We will investigate the dichotomy conjecture in the general case, for various classes of integrity constraints.

- We will start with the case which has a richer literature : key constraints and self-join free conjunctive queries. The current dichotomy proof for this case is unsatisfactory as it does not say how to compute the certain answers in the polynomial case. We would like to have a new proof of this result with an explicit construction of the query formulation. Hopefully this will open up new perspectives on the problem which will allow us to solve the general case of CQs with key dependencies.
- We will then move to larger classes of constraints for which the dichotomy conjecture is currently open. These include functional dependencies (which are a natural extension of key dependencies), GAV constraints (i.e. constraints of the form $\forall \bar{x}\ \alpha(\bar{x}) \rightarrow E(\bar{x})$, where α is a conjunction of atoms, and E is a single atom), and *equality-generating dependencies* - EGDs (i.e. constraints of the form $\forall \bar{x}\ \alpha(\bar{x}) \rightarrow x_i = x_j$, where x_i, x_j are in $\bar{x}$).

Repairs of graph-structured data Another interesting scenario deals with graph databases — arising from many novel applications that store their data in a graph structure. In this context, the problem of querying certain answers has not been completely explored yet. For instance, the decidability status of computing certain answers of conjunctive regular path queries (akin to conjunctive queries for relational databases) under guarded tuple generating dependencies (TGDs, i.e. constraints of the form $\forall \bar{x}\ \alpha(\bar{x}) \rightarrow \exists \bar{y}\ \beta(\bar{x}, \bar{y})$, where α and β are quantifier free conjunctions of atoms) or inclusion dependencies is unknown.

> We will study the decidability frontier for computing certain answers to regular path queries, and identify tractable cases.

PhD topic. The recruited PhD student will work on repairs of inconsistent databases. (S)he will start attacking the dichotomy conjecture for simple constraint languages, and then move to repairs of graph data. The PhD will also contribute to develop connections with CSP (described in section 1.1.4), as this objective spans through all the tasks of this proposal.

1.1.3 Data incompleteness

In this setting, queries are issued on an *incomplete* database, with some missing or unknown data values. Incompleteness arises naturally in many settings, going from simple typing errors to massive

data integration processes, that put together data from different sources; to this end, the addition of new data fields, which are not available in all sources, is often necessary.

This scenario makes query processing hard since, in this setting as well, one usually wants to obtain answers which are consistent with **all** possible completions of the available data. As before, this corresponds to the *certain answers*. The *semantics* of incompleteness specifies what an allowed completion is.

Assume for instance a query asks whether there exists a product order in the database which has not been shipped. Assume in the database we have $Orders = \{1, 2, \dots n\}$ and $Shipments = \{x_1, \dots x_{n-1}\}$, with $x_1, \dots x_{n-1}$ unknown order ids. We can answer with certainty that the set difference $Orders \setminus Shipments$ is non-empty, no matter the values of $x_1, \dots x_{n-1}$ – under the assumption that these values are the only information missing from the database, i.e. no additional elements are possibly missing from the *Shipments* table. This is usually referred to as the *closed world assumption* semantics of incompleteness, as opposed to the *open world assumption* allowing any possible extension of the database instance as a completion.

Clearly the query evaluation task in the presence of incompleteness may be more or less hard depending on the semantics of incompleteness. Moreover different fragments of query languages allow different query processing solutions, and have different computational properties over incomplete data. As one can expect, negation in queries is the main source of difficulty, as it asks for what does *not* belong to an incomplete database.

Although many tractability results have been found in many settings, a systematic approach offering generic tools to understand what guarantees tractability of querying incomplete data is still missing.

The objective of this research axis is to relate tractability of querying incomplete data to syntactic properties of queries

Characterising tractability of query evaluation based on syntactic/structural properties of queries is in general an important trend in database research. In fact we have similar objectives in our previous axis, concerning repairs of inconsistent databases.

However in this setting we are constrained by what is already in place in database systems. Classical database systems allow the presence of NULL values, representing missing data. Unfortunately it is known that even for the SQL standard, some of the rules for evaluating queries over databases with nulls result in plain wrong answers (i.e. provide answers which are not certain) [Lib16b]. This is done in order to achieve scalable query evaluation. However other efficient procedures, easily implementable in existing database systems can be devised. In fact while in general hard to compute, sometimes certain answers can be found by what is called *naïve evaluation* techniques. These essentially say: use the standard query evaluation engine provided by the DBMS, as if the database were complete, i.e. treat nulls as if they where new fresh data values, and evaluate the query as usual.

We have already started investigating when such naïve evaluation give correct answers, and obtained promising initial results [GLS14]. In fact, we established a general methodology for finding classes of queries for which naïve evaluation works under different semantics of incompleteness, and related it to classical notions in logic. In particular we have found out a close relationship with efficient testing of homomorphism preservation properties of classes of formulas over finite models. In this setting, tools from finite model theory turn out to be essential, as they provide a means to understand the relationship between syntactic and semantic properties of logics.

We plan to expand the scope of this investigation and make it more applicable in practical scenarios. We mention three directions here.

Integrity constraints The first direction is about answering queries under integrity constraints. All real-life databases come with a set of integrity rules (e.g., to ensure that two different people do not have the same passport number), and such constraints affect the complexity of query evaluation significantly in the presence of incompleteness. We need to know how they affect naïve evaluation techniques. Techniques, in the relational case, will be based on solving various implication problems in the finite, while for other data models they will combine algorithmic, logic, and automata techniques.

> We will explore efficient techniques for query answering under integrity constraints.

Naïve evaluation for graph-structured data Moving beyond the classical relational model of data, where data is structured into rigid tables, one finds several models which are more and more in use today. These include semi-structured data (such as *XML* and *json*) or graph data (such ad RDF, and *triple stores* of the noSQL family). The difficulty in dealing with such models, is not only the absence of a well established model of incompleteness, but also the different expressiveness of query languages, that often need to go beyond first order logic. The relationship between syntactic and semantic properties of queries has not received much attention beyond first order logics, with a few exceptions.

Using and possibly enriching the connection with finite model theory tools, we aim at studying efficient querying of incomplete data in other data models, in particular in the so called ***property graph model***. This is a more and more widespread data model, which is advocated by the Linked Data Benchmark Council (LDBC) and implemented for instance by the leader in the graph database market, Neo Technology Inc, in their data management system Neo4j [Ang+17]. Our work in collaboration with Neo Technology Inc has already produced formal semantics for **Cypher**, the query language used in Neo4j [Fra+18]. This provides a necessary starting point for applying formal methods to a real-life system. It turns out that many features of incompleteness can be expressed in Cypher, but not much is known concerning naïve evaluation of Cypher queries. There is a real opportunity here for foundational research to have an impact on commercial products, as Cypher has a potential for becoming a new standard, much like SQL for relational databases.

> We will study naïve evaluation of query languages akin to Cypher on the property graph model.

Approximation of certain answers Finally, finding certain answers being often a computationally hard problem, approximation algorithms have recently been proposed in the literature [CGL16]. There is still a lot to explore in that direction. We are interested in particular in applying this setting to new data models such as the property graph model, or in designing procedures that would permit to evaluate and compare the quality of various approximations.

> We will investigate various notions of efficient approximations for certain answering of incomplete data.

1.1.4 Connections with CSP

The three scenarios discussed earlier relate to one another in that they all amount to computing certain answers with various notions of *certainty*. It turns out that in many cases [Cal+00a; FSS15; LW15], the task of finding *certain answers* can be restated as a constraint satisfaction problem

(CSP). Furthermore, we believe that there may be a strong connection between CSPs and finding certain answers in the contexts described. Uncovering such relationships is a very worthwhile endeavour, as it can benefit both communities. On the one hand, it provides us with the very powerful tools that have been developed in the CSP area, while on the other hand it also opens new questions and provides new leads towards a finer understanding of CSPs themselves. This is the most prospective part of our research proposal. The connection between certain answers and CSP being still vague and not yet fully formalised.

We aim at finding formal connections between CSP and computing certain answers.

CSP and determinacy The case of determinacy, described in our first scenario, provides a compelling example. When the view determines a query, the query can be computed from the view. This amounts to saying that the query result is the same in all databases having the same view, thus in this case all query answers are certain (i.e. true no matter the missing information, not provided by the views). When the views are regular path queries, in the case of monotone determinacy, the certain answer computation problem can be casted as a CSP [Cal+00a]. However, even though the derived CSP instance is computationally hard, we have shown that it is still possible for the query to be computed efficiently [FSS15], and in particular in Datalog. The mismatch comes from the fact that CSP has no domain restriction while the computation of the query only needs to be performed on view images, and those may have specific properties that can be exploited to get a polynomial time algorithm [FSS15].

Datalog definability of CSPs has attracted much attention in the CSP literature, since it captures solvability of the CSP by the so called *local consistency checking game* (a two-player game on a graph), considered one of the most natural tractable approximations of a CSP [BK09].

To start with, this connection can possibly give us important insights on finding monotone rewritings with lower complexity than full Datalog. In fact there exist refinements of the local consistency checking game characterising definability of CSPs in fragments of Datalog. It is possible that some of these refinements can be successfully applied in our context. Among these fragments particularly interesting to us is *linear Datalog* [Dal05], a syntactic restriction of Datalog limiting the use of recursion, which we plan to consider first. Linear Datalog has NLOGSPACE complexity, the same as regular path queries. It thus represents a natural extension of basic graph query languages, which does not increase their complexity. Moreover complexity-wise it is in a sense optimal, if we want to be able to express reachability properties on graphs (which is a natural requirement for a graph query language).

Another direction where CSP results can possibly be exploited is the study of fixed parameter tractability of rewritings on views. Universal algebra approaches [BK09] have in fact shown that Datalog programs with fixed-parameter tractable data complexity are enough to express all Datalog-definable CSPs on graphs.

This result does not apply immediately to monotone rewritings on views, however we plan to investigate whether the techniques are applicable directly to our restricted setting, thus initiating the study of fixed parameter tractable determinacy.

CSP and repairs The database repair setting can also be linked with CSP. For instance if the query is a union of conjunctive queries, and not just a single conjunctive query, then a reduction from computing certain answers to CSP has been exhibited [Fon15]. This opens the

way for transferring complexity results from CSP to consistent query answering. Very recent progress on the complexity of CSP can thus have an important impact on the study of consistent query answering. To the best of our knowledge, this has not been investigated yet.

In the case of single conjunctive queries, from our initial investigation we conjecture that there is also a link between cases where certain answers can be computed in PTime and cases where CSPs have *bounded width*, i.e. can be solved using a Datalog program. This requires further investigations.

Feedback on CSP The determinacy setting provides a very strong motivation for us to study CSPs under a restricted input domain, a question that remains largely unexplored even in the CSP community.

We will initiate a study of CSP when the domain is not the full set of structures.

As a starting point, we plan to exploit the games underlying various classes of CSPs. In fact while general algebraic properties of CSPs are difficult to exploit, games characterising classes of CSPs have a clear interplay with the structural properties of the input data.

For instance in the case of monotone determinacy we were able to prove that there is a nice interplay between the local consistency checking game and the "regular" graph structure of view instances [FSS15]. It follows that, on such view instances, a large class of CSPs which are hard on arbitrary structures is actually solvable by local consistency checking. This approach has the potential to go beyond the specific results obtained in determinacy, and can possibly trigger a new study on restricted-domain CSP.

Other meaningful restrictions on the input structures are likely to arise from the repair setting as well.

1.2 Originality and relevance in relation to the state of the art

This proposal intends to attack several open problems, highly relevant in database theory and systems. This is ambitious, however our preliminary results, described in the previous section, are novel and very promising, as we explain next.

On the determinacy problem, as detailed in the previous section we have recently progressed significantly [FSS15; Fra17]. These results represent the first progress on the question after a number of years (the last results dating back to 2011 [Afr11]). This demonstrates that we have identified new techniques, worth being pursued, that could make a breach into notoriously difficult problems. In particular previous work exhibiting a connection between determinacy and CSP [Cal+00b] was in a sense limited to using CSP techniques as a black box. The strength of our new approach, started in [FSS15], is to have opened this black box and revealed a real interplay between the two problems. This has much more potential than we have exploited so far; we believe it can be the source of a fruitful exchange from CSP to database questions and back.

The management of data incompleteness, which has been thoroughly studied in the 80's [IL84], had reached a standstill, until attracting renewed interest in connection with data integration and exchange problems at the beginning of 2000's [Lib06]. However tools from the 80's were in a sense outdated at this point, as data models and query languages had significantly evolved. Our work [GLS14] was the first foundational study on data incompleteness after the seminal work from the 80's. The peculiarity of our framework is that it studies incompleteness in an algebraic domain, independent of the data model. This gives our approach the potential to be applicable beyond

the classical relational model, and in particular in the context of modern graph data, such as the property graphs. We thus believe that our objectives are at reach.

On this topic we plan to continue our tight collaboration with Leonid Libkin who is a leader in this area. Some of his more recent work has strong connections with the objectives of this research axis [Lib16a; CGL16]), in particular with approximation of certain answers, which we have already started looking at, in collaboration with Libkin.

Querying database repairs has a rather rich state of the art. We have witnessed many attempts to prove the dichotomy conjecture in the case of conjunctive queries, starting in 2005 [FM05] and culminating in a partial solution only in 2015 [KW15], obtaining a dichotomy for self-join free conjunctive queries. This restriction on queries is very ad-hoc, only motivated by technical reasons and not corresponding to any realistic scenario. This demonstrates that the problem needs to be looked at from a new angle.

Although we have not yet published in this area, we have identified many similarities between querying database repairs and the other two axes of this project. Surprisingly the three areas have always been studied independently, while they are actually strongly tied – as witnessed by the fact that both determinacy and repairs have been independently related to CSP [Cal+00a; FSS15; LW15]. We believe that generalising the relationship with CSP is the key that will allow us to transfer techniques from one problem to the other, and thus open new investigation avenues for the repair problem as well.

1.3 Methodology and risk management

This project concerns foundational research, which always has the risk of coming up with unexpected and undesired results: we cannot *choose* the truth of theorems, we only can *find* the true statements. Nevertheless, the research program is based upon 3 distinct scenarios, each of which can be completed with the others failing. That being said, we do not foresee big risks for the research related to the three scenarios mentioned in Section 1.1. All members of our consortium have already worked in related areas and have accumulated enough experience for being confident about the success of this part.

The connection with CSP is certainly of higher risk (but also of higher impact). As mentioned earlier, we have already used results from the CSP community and have accumulated enough experience to have the intuition that the connections mentioned in Section 1.1 exist. We do not have an expert from the CSP community in our group, but such an expert is probably not necessary for *formalising the connections* between certain answers and CSP as only superficial knowledge about CSP is necessary here. However, we stress that the study of CSP with restricted domain may require a much deeper understanding of CSP. In that respect, we are already working with Benoît Larose (Université du Québec, Montreal), expert in CSP, about related issues. Luc Segoufin has already made two recent visits to Montreal in order to discuss these issues with him. Furthermore we are planning several visits of Benoît in Paris during the project in order to check with him that we are on the right track.

We are aware of the fact that we cannot foresee all risks. We are nevertheless confident that, due to the established partnership between the participating sites, we will be able to deal with any reasonable challenges.

2 Project organisation and means implemented

2.1 Consortium

2.1.1 Scientific coordinator

The project will be coordinated by Claire David, maître de conférences (MCF) at Laboratoire d'Informatique Gaspard Monge (LIGM) and she will be involved at 60% in the project. She did her PhD at LIAFA (Université Paris-Diderot) in a research team focusing on formal methods for verification, automata and games theory. She spent two years in the database team at LFCS (University of Edinburgh) before joining the Modèles et Algorithmes team at LIGM (Université Paris-Est Marne-la-Vallée). Her main research interests are database theory, logic and automata; she focuses on new models of data such as trees and graphs. In particular, she has worked on questions related to incomplete information [DFM14; Ama+14], certain answers [DLM10] and integrity constraints [Cze+16] for tree structured data.

Her results appear in international journals such as JCSS, ACM-TOCL, ACM-TODS, JACM and in the major database theory international conferences PODS and ICDT. She also published in other international theory conferences such as FOSSACS, FSTTCS, LICS, MFCS, LPAR, GANDALF. She received the best paper award at ICDT'2011 and together with Luc Segoufin, they received the best paper award at PODS'2006 which was followed by the test of time award at PODS'2016. She regularly serves on program committees for major database conferences (ICDT in 2018, 2014 and 2012, PODS in 2016 and 2014, ICDT test of time award in 2016 and PODS PhD Symposium in 2015) but also for smaller events (Highlights'2016, JFLA'2016, BDA'2013 and BDA'2011, LID'2011).

She has international research collaborations (e.g. with F. Murlak and his colleagues in Warsaw, L. Libkin in Edinburgh, W. Martens in Bayreuth) and has been involved in the writing of the recent Dagstuhl Manifesto "Research Directions for Principles of Data Management" also published in ACM SIGMOD Record [Abi+16].

She also has experience in organising international meetings and events. For example, she was involved in the organisation of FLOC'10 (over 1000 participants) and one of the main organisers of Highlights'2014 (180 participants).

On top of her research contribution to this project, she will coordinate the project and lead the organisation of the annual meetings and international workshops.

2.1.2 Consortium

The consortium involves members of Computer Science departments from four academic institutions: Université Paris-Est Marne-la-Vallée, Université Paris-Diderot, ENS Ulm - Inria and Université de Bordeaux. Each site contributes with two permanent researchers. The consortium includes three full-time researchers (CNRS and Inria), three Maîtres de Conférences and two professors, and plans on hiring two 1-year post-doc fellows and one PhD student.

Claire David and Nadime Francis are both Maître de Conférences at LIGM (Université Paris-Est Marne-la-Vallée). Cristina Sirangelo is a Professor and Amélie Gheerbrant is a Maître de Conférences at IRIF (Université Paris-Diderot). Luc Segoufin is a Directeur de Recherche at Inria and Pierre Senellart is Professor in the Département Informatique of ENS Ulm. Diego Figueira and Gabriele Puppis are both CNRS Chargé de Recherche from LaBRI (Université de Bordeaux).

All members share a strong expertise and international visibility in database theory, which is the core of the project. They also have strong knowledge in formal methods for verification, logic, automata and finite model theory, whose tools will be central for the research program. Some of their previous work relate to the main axes of the project as follows:

Nadime Francis, Cristina Sirangelo and Luc Segoufin already worked together on the view determinacy problem [FSS15; Fra17]. This work has stressed the link between determinacy, certain answer and CSP for restricted domain.

The problem of certain answers has been studied by Claire David [DLM10; Ama+14], Amélie Gheerbrant, Cristina Sirangelo [GLS14; LS11] and Nadime Francis [FL17] for relational, semi-structured or graph databases, in the context of incomplete information settings such as data exchange.

Diego Figueira and Gabriele Puppis have worked on static analysis and efficient querying over semi-structured data [Bou+16; Ben+15; FS17; Fig16]. Diego Figueira also recently started some work on privacy in the context of querying relational databases [AFG16].

Some of the problems addressed in the project are difficult long standing problems. We will make use of the strong combined expertise of the consortium in order to achieve significant progress in these areas. It is important to note that each member of the consortium brings together with them strong international collaborations with experts who will be able to provide insight or feedback on the project outputs. The ongoing collaboration with Benoit Larose has been developed to make progress on the connection between certain answers and CSP. We can also mention Leonid Libkin for his interest in incompleteness and his strong expertise in finite model theory, Pablo Barcelo regarding approximations of query answering, and with Myrto Arapanis regarding privacy questions, among others.

Finally, several members of the consortium have already successfully worked together in the past and are willing to strengthen or revive these collaborations.

2.2 Scientific programme

The outline of our scientific programme directly follows the various research directions detailed in Section 1. We give here a brief overview of the task breakdown that additionally specifies which members are mainly involved in each task. Note that the following outline does not mention workshop organisation tasks, which will be carried out by the coordinator. We plan to work on all tasks during the whole duration of the project, with an emphasis of the first three at the beginning of the project, and an emphasis on task four towards the end of the project.

Task 1 Determinacy
Members involved: CD, NF, LS, CS, post-doc.

Task 1.1 Deciding determinacy
Find out the decidability status of the determinacy problem for various common query languages. Find meaningful query language fragments for which the determinacy problem is decidable.

Task 1.2 Efficient determinacy
Determine when a given query can be rewritten using given views in a specific target language that has efficient query evaluation. Determine when a given query can be efficiently evaluated using given views.

Task 2 Repairs
Members involved: DF, AG, GP, LS, PS, CS, PhD student.

Task 2.1 Dichotomy conjecture
Solve the coNP vs. PTime conjecture for conjunctive queries under key constraints. Establish similar results for larger integrity constraints, such as functional dependencies and GAV.

Task 2.2 Repairs of graph-structured data

Study the case of graph databases and regular path queries under various cases of integrity constraints.

Task 3 Incompleteness

Members involved: CD, DF, NF, AG, PS, CS, post-doc.

Task 3.1 Query answering under integrity constraints

Design and study efficient techniques for answering queries under integrity constraints in the presence of incomplete data.

Task 3.2 Naïve evaluation for graph-structured data

Find syntactic fragments of both real-life query languages over graph data and their theoretical abstractions for which naïve evaluation provides the correct answers.

Task 3.3 Approximation of certain answers

Design efficient techniques for computing approximations of certain answers over both relational and graph data. Design ways of evaluating and comparing the quality of such techniques.

Task 4 Constraint Satisfaction Problem

Members involved: CD, DF, NF, AG, GP, LS, CS, PhD student.

Task 4.1 CSP and determinacy

Refine the use of CSP techniques in determinacy, so as to lower the complexity of rewritings.

Task 4.2 CSP and repairs

Investigate the connection between computing certain answers of conjunctive queries under key constraints and CSP.

Task 4.3 Feedback on CSP

Design a theoretical framework for studying CSP under restricted input domain.

2.3 Means requested

Members (in p.m.)

	LIGM	**IRIF**	**Inria Paris**	**LaBRI**	Total
Permanents					
PhD					
Postdocs					
Total					

Funding requests (in K€)

	LIGM	**IRIF**	**Inria Paris**	**LaBRI**	Total
Post-docs & PhD					
Travel & project meetings					
Workshops					
Other Supplies					

Total Cost and Funding request (in K€, including 8% overhead)

	LIGM	**IRIF**	**Inria Paris**	**LaBRI**	Total
Total Cost					
Funding request					

3 Impact and benefits

3.1 Expected impact

Views, repairs and incomplete information are the sources of recurrent challenges for database systems. These challenges have been addressed by the scientific community since the beginning of databases and there is a long history of tentative or partial solutions. QUID aims at unifying all these approaches.

The main impact of our research will therefore be on the state of the art in database theory, and many of our results will be published in the best conferences in the field such as PODS or ICDT.

Our results will hopefully exhibit links between all our three scenarios and other fields of computer science such as the CSP community. This part is more prospective but, should it succeed, its impact would be broadened to all fields that are connected with the CSP community, such as universal algebra and proof theory among others.

These results will be disseminated to conferences of broader audiences such as STACS, LICS, ICALP, in order to impact all the desired communities. Moreover, many application domains can benefit from the results we aim at. These include – besides the core of database systems – data privacy, data integration and ontological reasoning.

3.2 Within the current ANR call

As explained in the beginning of the document, we aim to develop formal methods for analysing data management protocols as well as efficient algorithms for computing relevant answers in the

context of incomplete or inconsistent information. The issues we address in this project have been identified as research directions of high relevance to society by the database community [Abi+16]. Our research will impact the reliability of systems manipulating data. This includes **data security**, **data leakage** and **management of inconsistent or incomplete data**.

In this sense, the proposed research project lies perfectly in the challenge **B7, Axe 4**:

"Données, Connaissances, Big Data, Contenus multimédias - Intelligence Artificielle".

In particular the following sentences of the Plan d'Action 2018, page 57 in the themes "Des données aux connaissances" and "Big data", fits completely within our research agenda:

"*prise en compte de la protection des données individuelles et de la sécurité des données et de leurs traitements*",

"*Un point clé consiste à produire des connaissances vérifiées, en assurant la robustesse des processus face aux données incomplètes, incertaines ou imprécises [...] et des connaissances vérifiables, en proposant des processus favorisant la transparence du raisonnement, et des analyses améliorant la compréhension.*"

3.3 Dissemination Strategy

The results of QUID will be mainly disseminated by publications in major conferences and in major journals. Moreover, the project will have a dedicated website advertising our results. Finally, another effective means to disseminate the results will be through the organisation of the two international workshops, targeting the most prominent researchers in the area.

4 Bibliography

[Abi+16] S. Abiteboul, M. Arenas, P. Barceló, M. Bienvenu, D. Calvanese, C. David, R. Hull, E. Hüllermeier, B. Kimelfeld, L. Libkin, W. Martens, T. Milo, F. Murlak, F. Neven, M. Ortiz, T. Schwentick, J. Stoyanovich, J. Su, D. Suciu, V. Vianu, and K. Yi. "Research Directions for Principles of Data Management (Abridged)". In: *SIGMOD Record* 45.4 (2016), pp. 5–17. DOI: 10.1145/3092931.3092933. URL: http://doi.acm.org/10.1145/3092931.3092933.

[Afr11] F. Afrati. "Determinacy and query rewriting for conjunctive queries and views". In: *Theoretical Computer Science* 412.11 (2011), pp. 1005–1021.

[Ama+14] S. Amano, C. David, L. Libkin, and F. Murlak. "XML Schema Mappings: Data Exchange and Metadata Management". In: *J. ACM* 61.2 (2014), 12:1–12:48. DOI: 10.1145/2590773. URL: http://doi.acm.org/10.1145/2590773.

[Ang+17] R. Angles, M. Arenas, P. Barceló, P. A. Boncz, G. H. L. Fletcher, C. Gutierrez, T. Lindaaker, M. Paradies, S. Plantikow, J. F. Sequeda, O. van Rest, and H. Voigt. "G-CORE: A Core for Future Graph Query Languages". In: *CoRR* abs/1712.01550 (2017). arXiv: 1712.01550. URL: http://arxiv.org/abs/1712.01550.

[AFG16] M. Arapinis, D. Figueira, and M. Gaboardi. "Sensitivity of Counting Queries". In: *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*. Ed. by I. Chatzigiannakis, M. Mitzenmacher, Y. Rabani, and D. Sangiorgi. Vol. 55. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016, 120:1–120:13. ISBN: 978-3-95977-013-2. DOI: 10.4230/LIPIcs.ICALP.2016.120. URL: https://doi.org/10.4230/LIPIcs.ICALP.2016.120.

[ABC03] M. Arenas, L. E. Bertossi, and J. Chomicki. "Answer sets for consistent query answering in inconsistent databases". In: *TPLP* 3.4-5 (2003), pp. 393–424. DOI: 10.1017/S1471068403001832. URL: https://doi.org/10.1017/S1471068403001832.

[BK09] L. Barto and M. Kozik. "Constraint Satisfaction Problems of Bounded Width". In: *FOCS*. 2009, pp. 595–603.

[Ben+15] M. Benedikt, P. Bourhis, B. ten Cate, and G. Puppis. "Querying Visible and Invisible Tables in the Presence of Integrity Constraints". In: *CoRR* abs/1509.01683 (2015). arXiv: 1509.01683. URL: http://arxiv.org/abs/1509.01683.

[Bie+14] M. Bienvenu, B. ten Cate, C. Lutz, and F. Wolter. "Ontology-Based Data Access: A Study through Disjunctive Datalog, CSP, and MMSNP". In: *ACM Trans. Database Syst.* 39.4 (2014), 33:1–33:44. DOI: 10.1145/2661643. URL: http://doi.acm.org/10.1145/2661643.

[Bou+16] P. Bourhis, G. Puppis, C. Riveros, and S. Staworko. "Bounded Repairability for Regular Tree Languages". In: *ACM Trans. Database Syst.* 41.3 (2016), 18:1–18:45. DOI: 10.1145/2898995. URL: http://doi.acm.org/10.1145/2898995.

[Cal+00a] D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Y. Vardi. "View-Based Query Processing and Constraint Satisfaction". In: *LICS*. IEEE, 2000, pp. 361–371. DOI: 10.1109/LICS.2000.855784. URL: https://doi.org/10.1109/LICS.2000.855784.

[Cal+00b] D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Y. Vardi. "View-based query processing and constraint satisfaction". In: *LICS*. IEEE. 2000, pp. 361–371.

[Cal+11] A. Calì, G. Gottlob, T. Lukasiewicz, and A. Pieris. "A logical toolbox for ontological reasoning". In: *SIGMOD Record* 40.3 (2011), pp. 5–14. DOI: 10.1145/2070736.2070738. URL: http://doi.acm.org/10.1145/2070736.2070738.

[CGL16] M. Console, P. Guagliardo, and L. Libkin. "Approximations and Refinements of Certain Answers via Many-Valued Logics". In: *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, April 25-29, 2016*. 2016, pp. 349–358. URL: http://www.aaai.org/ocs/index.php/KR/KR16/paper/view/12813.

[Cze+16] W. Czerwinski, C. David, F. Murlak, and P. Parys. "Reasoning About Integrity Constraints for Tree-Structured Data". In: *19th International Conference on Database Theory, ICDT 2016, Bordeaux, France, March 15-18, 2016*. 2016, 20:1–20:18. DOI: 10.4230/LIPIcs.ICDT.2016.20. URL: https://doi.org/10.4230/LIPIcs.ICDT.2016.20.

[Dal05] V. Dalmau. "Linear datalog and bounded path duality of relational structures". In: *Logical Methods in Computer Science* 1.1 (2005).

[DFM14] C. David, N. Francis, and F. Murlak. "Consistency of Injective Tree Patterns". In: *34th International Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS 2014, December 15-17, 2014, New Delhi, India*. 2014, pp. 279–290. DOI: 10.4230/LIPIcs.FSTTCS.2014.279. URL: https://doi.org/10.4230/LIPIcs.FSTTCS.2014.279.

[DLM10] C. David, L. Libkin, and F. Murlak. "Certain answers for XML queries". In: *Proceedings of the Twenty-Ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2010, June 6-11, 2010, Indianapolis, Indiana, USA*. 2010, pp. 191–202. DOI: 10.1145/1807085.1807112. URL: http://doi.acm.org/10.1145/1807085.1807112.

[Fig16] D. Figueira. "Semantically Acyclic Conjunctive Queries under Functional Dependencies". In: *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*. Ed. by M. Grohe, E. Koskinen, and N. Shankar. ACM, 2016, pp. 847–856. ISBN: 978-1-4503-4391-6. DOI: 10.1145/2933575.2933580. URL: http://doi.acm.org/10.1145/2933575.2933580.

[FS17] D. Figueira and L. Segoufin. "Bottom-up automata on data trees and vertical XPath". In: *Logical Methods in Computer Science* 13.4 (2017). DOI: 10.23638/LMCS-13(4:5)2017. URL: https://doi.org/10.23638/LMCS-13(4:5)2017.

[Fon15] G. Fontaine. "Why Is It Hard to Obtain a Dichotomy for Consistent Query Answering?" In: *ACM Trans. Comput. Log.* 16.1 (2015), 7:1–7:24. DOI: 10.1145/2699912. URL: http://doi.acm.org/10.1145/2699912.

[Fra15] N. Francis. "View-based query determinacy and rewritings over graph databases. (Vues et requêtes sur les graphes de données : déterminabilité et réécritures)". PhD thesis. University of Paris-Saclay, Versailles, Saint-Quentin-en-Yvelines, Saint-Aubin, Essonne, France, 2015. URL: https://tel.archives-ouvertes.fr/tel-01247115.

[Fra17] N. Francis. "Asymptotic Determinacy of Path Queries Using Union-of-Paths Views". In: *Theory Comput. Syst.* 61.1 (2017), pp. 156–190. DOI: 10.1007/s00224-016-9697-x. URL: https://doi.org/10.1007/s00224-016-9697-x.

[FL17] N. Francis and L. Libkin. "Schema Mappings for Data Graphs". In: *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2017, Chicago, IL, USA, May 14-19, 2017*. 2017, pp. 389–401. DOI: 10.1145/3034786.3056113. URL: http://doi.acm.org/10.1145/3034786.3056113.

[FSS15] N. Francis, L. Segoufin, and C. Sirangelo. "Datalog Rewritings of Regular Path Queries using Views". In: *Logical Methods in Computer Science* 11.4 (2015). DOI: 10.2168/LMCS-11(4:14)2015. URL: https://doi.org/10.2168/LMCS-11(4:14)2015.

[Fra+18] N. Francis, A. Green, P. Guagliardo, L. Libkin, T. Lindaaker, V. Marsault, S. Plantikow, M. Rydberg, P. Selmer, and A. Taylor. "Cypher: An Evolving Query Language for Property Graphs". In: *Proceedings of the 2018 ACM International Conference on Management of Data (SIGMOD)*. To appear. 2018.

[FM05] A. Fuxman and R. J. Miller. "First-Order Query Rewriting for Inconsistent Databases". In: *Database Theory - ICDT 2005, 10th International Conference, Edinburgh, UK, January 5-7, 2005, Proceedings*. 2005, pp. 337–351. DOI: 10.1007/978-3-540-30570-5_23. URL: https://doi.org/10.1007/978-3-540-30570-5_23.

[FM07] A. Fuxman and R. J. Miller. "First-order query rewriting for inconsistent databases". In: *J. Comput. Syst. Sci.* 73.4 (2007), pp. 610–635. DOI: 10.1016/j.jcss.2006.10.013. URL: https://doi.org/10.1016/j.jcss.2006.10.013.

[GLS14] A. Gheerbrant, L. Libkin, and C. Sirangelo. "NaÏVe Evaluation of Queries over Incomplete Databases". In: *ACM Trans. Database Syst.* 39.4 (Dec. 2014), 31:1–31:42. ISSN: 0362-5915. DOI: 10.1145/2691190.2691194. URL: http://doi.acm.org/10.1145/2691190.2691194.

[GM15] T. Gogacz and J. Marcinkowski. "The Hunt for a Red Spider: Conjunctive Query Determinacy Is Undecidable". In: *LICS*. IEEE, 2015, pp. 281–292. DOI: 10.1109/LICS.2015.35. URL: https://doi.org/10.1109/LICS.2015.35.

[GM16] T. Gogacz and J. Marcinkowski. "Red Spider Meets a Rainworm: Conjunctive Query Finite Determinacy Is Undecidable". In: *PODS*. 2016, pp. 121–134. DOI: 10.1145/2902251.2902288. URL: http://doi.acm.org/10.1145/2902251.2902288.

[IL84] T. Imielinski and W. Lipski. "Incomplete Information in Relational Databases". In: *JACM* 31.4 (1984), pp. 761–791.

[KW15] P. Koutris and J. Wijsen. "The Data Complexity of Consistent Query Answering for Self-Join-Free Conjunctive Queries Under Primary Key Constraints". In: *PODS*. ACM, 2015, pp. 17–29. DOI: 10.1145/2745754.2745769. URL: http://doi.acm.org/10.1145/2745754.2745769.

[Lib06] L. Libkin. "Data exchange and incomplete information". In: *PODS*. 2006, pp. 60–69.

[Lib15] L. Libkin. "How to Define Certain Answers". In: *Proceedings of the 24th International Conference on Artificial Intelligence*. IJCAI'15. Buenos Aires, Argentina: AAAI Press, 2015, pp. 4282–4288. ISBN: 978-1-57735-738-4. URL: http://dl.acm.org/citation.cfm?id=2832747.2832854.

[Lib16a] L. Libkin. "Certain answers as objects and knowledge". In: *Artif. Intell.* 232 (2016), pp. 1–19. DOI: 10.1016/j.artint.2015.11.004. URL: https://doi.org/10.1016/j.artint.2015.11.004.

[Lib16b] L. Libkin. "SQL's Three-Valued Logic and Certain Answers". In: *ACM Trans. Database Syst.* 41.1 (2016), 1:1–1:28. DOI: 10.1145/2877206. URL: http://doi.acm.org/10.1145/2877206.

[LS11] L. Libkin and C. Sirangelo. "Data exchange and schema mappings in open and closed worlds". In: *J. Comput. Syst. Sci.* 77.3 (2011), pp. 542–571. DOI: 10.1016/j.jcss.2010.04.010. URL: https://doi.org/10.1016/j.jcss.2010.04.010.

[LW15] C. Lutz and F. Wolter. "On the Relationship between Consistent Query Answering and Constraint Satisfaction Problems". In: *18th International Conference on Database Theory, ICDT 2015, March 23-27, 2015, Brussels, Belgium*. 2015, pp. 363–379. DOI: 10.4230/LIPIcs.ICDT.2015.363. URL: https://doi.org/10.4230/LIPIcs.ICDT.2015.363.

[NSV10] A. Nash, L. Segoufin, and V. Vianu. "Views and queries: Determinacy and rewriting". In: *ACM Trans. Database Syst.* 35.3 (2010), 21:1–21:41. DOI: 10.1145/1806907.1806913. URL: http://doi.acm.org/10.1145/1806907.1806913.

[Wij10] J. Wijsen. "A remark on the complexity of consistent conjunctive query answering under primary key violations". In: *Inf. Process. Lett.* 110.21 (2010), pp. 950–955. DOI: 10.1016/j.ipl.2010.07.021. URL: https://doi.org/10.1016/j.ipl.2010.07.021.

[Wij12] J. Wijsen. "Certain Conjunctive Query Answering in First-order Logic". In: *ACM Trans. Database Syst.* 37.2 (June 2012), 9:1–9:35. ISSN: 0362-5915. DOI: 10.1145/2188349.2188351. URL: http://doi.acm.org/10.1145/2188349.2188351.