

RESEARCH CENTRE

**Inria Centre at the University of  
Lille**

IN PARTNERSHIP WITH:  
CNRS, Université de Lille

2024  
**ACTIVITY REPORT**

Project-Team  
**LINKS**

## **Linking Dynamic Data**

IN COLLABORATION WITH: Centre de Recherche en Informatique, Signal  
et Automatique de Lille

### **DOMAIN**

**Perception, Cognition and Interaction**

### **THEME**

**Data and Knowledge Representation and  
Processing**

*Inria*

# Contents

|  |           |
|--|-----------|
| <b>Project-Team LINKS</b>  | <b>1</b>  |
| <b>1 Team members, visitors, external collaborators</b>  | <b>3</b>  |
| <b>2 Overall objectives</b>  | <b>4</b>  |
| 2.1 Presentation   | 4         |
| <b>3 Research program</b>  | <b>4</b>  |
| 3.1 Background   | 4         |
| 3.2 Research Axis: Querying Data Graphs  | 5         |
| 3.2.1 AI: Circuits for Data Analysis   | 5         |
| 3.2.2 Path Query Optimization  | 5         |
| 3.3 Research Axis: Monitoring Data Graphs  | 6         |
| 3.3.1 Functional Programming Languages for Data Graphs   | 6         |
| 3.3.2 Hyperstreaming Program Evaluation  | 6         |
| 3.4 Research Axis: Graph Data Integration  | 7         |
| 3.4.1 Data Quality with Schemas and Repairing with Inference   | 7         |
| 3.4.2 Integration and Graph Mappings with Schemas and Inference  | 7         |
| <b>4 Application domains</b>   | <b>8</b>  |
| 4.1 Linked data integration  | 8         |
| 4.2 Data cleaning  | 8         |
| 4.3 Real-time complex event processing   | 8         |
| <b>5 Social and environmental responsibility</b>   | <b>9</b>  |
| 5.1 Footprint of research activities   | 9         |
| 5.2 Women in science   | 9         |
| 5.3 Impact of research results   | 9         |
| <b>6 Highlights of the year</b>  | <b>9</b>  |
| 6.1 Awards   | 9         |
| 6.2 Evaluations  | 9         |
| <b>7 New software, platforms, open data</b>  | <b>9</b>  |
| 7.1 New software   | 9         |
| 7.1.1 NetworkDisk  | 9         |
| 7.1.2 Bibendum   | 9         |
| 7.1.3 XPath AutoBench  | 10        |
| 7.1.4 rsonpath   | 10        |
| 7.1.5 Coussinet  | 10        |
| 7.1.6 ShEx validator   | 11        |
| 7.1.7 gMark  | 11        |
| <b>8 New results</b>   | <b>11</b> |
| 8.1 Circuits for data manipulation   | 11        |
| 8.1.1 Circuits and knowledge compilation   | 11        |
| 8.1.2 Foundation of circuits: complexity, algebra and abstract machines  | 12        |
| 8.2 Logic and query evaluation   | 12        |
| 8.2.1 Provenance, explanation, aggregation, counting, uncertainty, probabilistic data, Shapley, approximation algorithms | 12        |
| 8.2.2 Efficient evaluation: streaming and parallelism  | 13        |
| 8.2.3 Enumerating the results of queries   | 13        |
| 8.3 Knowledge on Data  | 14        |
| 8.3.1 Optimization for schemas for graphs  | 14        |

|           |  |           |
|-----------|--|-----------|
| 8.3.2     | Static analysis with constraints (open-world query answering, query rewriting under constraints, consistency, certain answers) | 14        |
| 8.4       | Highlighted results  | 14        |
| <b>9</b>  | <b>Partnerships and cooperations</b>   | <b>15</b> |
| 9.1       | International research visitors  | 15        |
| 9.1.1     | Visits of international scientists   | 15        |
| 9.2       | National initiatives   | 16        |
| 9.3       | Regional initiatives   | 16        |
| <b>10</b> | <b>Dissemination</b>   | <b>17</b> |
| 10.1      | Promoting scientific activities  | 17        |
| 10.1.1    | Journal  | 17        |
| 10.1.2    | Invited talks  | 17        |
| 10.1.3    | Leadership within the scientific community   | 17        |
| 10.1.4    | Scientific expertise   | 17        |
| 10.1.5    | Research administration  | 17        |
| 10.2      | Teaching - Supervision - Juries  | 18        |
| 10.2.1    | Supervision  | 18        |
| 10.2.2    | PhD defended   | 18        |
| 10.2.3    | HDR defended   | 18        |
| 10.2.4    | Juries   | 18        |
| 10.2.5    | Teaching Responsibilities  | 18        |
| 10.2.6    | Teaching Activities  | 18        |
| 10.3      | Popularization   | 19        |
| 10.3.1    | Specific official responsibilities in science outreach structures  | 19        |
| <b>11</b> | <b>Scientific production</b>   | <b>19</b> |
| 11.1      | Major publications   | 19        |
| 11.2      | Publications of the year   | 20        |

## Project-Team LINKS

*Creation of the Project-Team: 2016 June 01*

### Keywords

#### Computer sciences and digital sciences

- A2.1. – Programming Languages
  - A2.1.1. – Semantics of programming languages
  - A2.1.4. – Functional programming
  - A2.1.6. – Concurrent programming
- A2.4. – Formal method for verification, reliability, certification
  - A2.4.1. – Analysis
  - A2.4.2. – Model-checking
  - A2.4.3. – Proofs
- A3.1. – Data
  - A3.1.1. – Modeling, representation
  - A3.1.2. – Data management, quering and storage
  - A3.1.3. – Distributed data
  - A3.1.4. – Uncertain data
  - A3.1.5. – Control access, privacy
  - A3.1.6. – Query optimization
  - A3.1.7. – Open data
  - A3.1.8. – Big data (production, storage, transfer)
  - A3.1.9. – Database
  - A3.2.1. – Knowledge bases
  - A3.2.2. – Knowledge extraction, cleaning
  - A3.2.3. – Inference
  - A3.2.4. – Semantic Web
- A4.7. – Access control
- A4.8. – Privacy-enhancing technologies
- A7. – Theory of computation
  - A7.2. – Logic in Computer Science
- A9.1. – Knowledge
- A9.2. – Machine learning
- A9.7. – AI algorithmics
- A9.8. – Reasoning

**Other research topics and application domains**

B6.1. – Software industry

B6.3.1. – Web

B6.3.4. – Social Networks

B6.5. – Information systems

B9.5.1. – Computer science

B9.5.6. – Data science

B9.10. – Privacy

# 1 Team members, visitors, external collaborators

## Research Scientists

- Antoine Amarilli [INRIA, Advanced Research Position, from Sep 2024]
- Mikael Monet [INRIA, Researcher]
- Joachim Niehren [INRIA, Senior Researcher]

## Faculty Members

- Sylvain Salvati [Team leader, UNIV LILLE, Professor Delegation]
- Iovka Boneva [UNIV LILLE, Associate Professor]
- Florent Capelli [UNIV LILLE, Associate Professor]
- Aurélien Lemay [UNIV LILLE, Associate Professor]
- Charles Paperman [UNIV LILLE, Associate Professor]
- Sophie Tison [UNIV LILLE, Emeritus]

## PhD Students

- Antonio Al Serhali [UNIV LILLE, Supervised by Niehren]
- Corentin Barloy [UNIV LILLE, until Aug 2024, Supervised by Paperman and Salvati]
- Bastien Degardins [UNIV LILLE, from Oct 2024, Supervised by Paperman and Marchet (CRISAL Bonsai)]
- Oliver Irwin [UNIV LILLE, Supervised by Capelli and Salvati]
- Joel Mba Kouhoue [UNIV LILLE, ATER, from Sep 2024]

## Interns and Apprentices

- Mathias Berry [INRIA, Intern, from Mar 2024 until Jul 2024, Supervised by Salvati]
- Djilide Mbenard Dabo [UNIV BORDEAUX, Intern, from Jun 2024 until Jul 2024, Supervised by Paperman]
- Bastien Degardins [UNIV LILLE, Intern, from Feb 2024 until Sep 2024]
- Margaux Mouton [UNIV LILLE, Intern, from Jun 2024 until Jul 2024, Supervised by Paperman]

## Administrative Assistants

- Nathalie Bonte [INRIA]
- Karine Lewandowski [INRIA]

## Visiting Scientists

- Sylvain Schmitz [UNIV PARIS XIII]
- Tatiana Starikovskaia [École normale supérieure, Paris]

## 2 Overall objectives

We develop algorithms for answering logical querying on heterogeneous linked data collections in hybrid formats, distributed programming languages for managing dynamic linked data collections and workflows based on queries and mappings, and symbolic machine learning algorithms that can link datasets by inferring appropriate queries and mappings.

### 2.1 Presentation

The following three items summarize our main research objectives.

**Querying Heterogeneous Linked Data.** We develop new kinds of schema mappings for semi-structured datasets in hybrid formats including graph databases, RDF collections, and relational databases. These induce recursive queries on linked data collections for which we will investigate evaluation algorithms, containment problems, and concrete applications.

**Managing Dynamic Linked Data.** In order to manage dynamic linked data collections and workflows, we will develop distributed data-centric programming languages with streams and parallelism, based on novel algorithms for incremental query answering, study the propagation of updates of dynamic data through schema mappings, and investigate static analysis methods for linked data workflows.

**Linking Data Graphs.** Finally, we will develop symbolic machine learning algorithms, for inferring queries and mappings between linked data collections in various graph formats from annotated examples.

## 3 Research program

### 3.1 Background

The main objective of LINKS is to develop methods for querying and managing linked data collections. Even though open linked data is the most prominent example, we will focus on hybrid linked data collections, which are collections of semi-structured datasets in hybrid formats: graph-based, RDF, relational, and NOSQL. The elements of these datasets may be linked, either by pointers or by additional relations between the elements of the different datasets, for instance the “same-as” or “member-of” relations as in RDF.

The advantage of traditional data models is that there exist powerful querying methods and technologies that one might want to preserve. In particular, they come with powerful schemas that constraint the possible manners in which knowledge is represented to a finite number of patterns. The exhaustiveness of these patterns is essential for writing queries that cover all possible cases. Pattern violations are excluded by schema validation. In contrast, RDF schema languages such as RDFS can only enrich the relations of a dataset by new relations, which also helps for query writing, but which cannot constraint the number of possible patterns, so that they do not come with any reasonable notion of schema validation.

The main weakness of traditional formats, however, is that they do not scale to large data collections as stored on the Web, while the RDF data models scale well to very big collections such as linked open data. Therefore, our objective is to study mixed data collections, some of which may be in RDF format, in which we can lift the advantages of smaller datasets in traditional formats to much larger linked data collections. Such data collections are typically distributed over the internet, where data sources may have rigid query facilities that cannot be easily adapted or extended.

The main assumption that we impose in order to enable the logical approach, is that the given linked data collection must be correct in most dimensions. This means that all datasets are well-formed with respect to their available constraints and schemas, and clean with respect to the data values in most of the components of the relations in the datasets. One of the challenges is to integrate good quality RDF datasets into this setting, another is to clean the incorrect data in those dimensions that are less proper. It remains to be investigated how far these assumptions can be maintained in realistic applications, and how much they can be weakened otherwise.

For querying linked data collections, the main problems are to resolve the heterogeneity of data formats and schemas, to understand the efficiency and expressiveness of recursive queries, that can follow links repeatedly, to answer queries under constraints, and to optimize query answering algorithms based on static analysis. When linked data is dynamically created, exchanged, or updated, the problems are how to process linked data incrementally, and how to manage linked data collections that change dynamically. In any case (static and dynamic) one needs to find appropriate schema mappings for linking semi-structured datasets. We will study how to automatize parts of this search process by developing symbolic machine learning techniques for linked data collections.

## 3.2 Research Axis: Querying Data Graphs

Linked data is often abstracted as datagraphs: nodes carry information and edges are labeled. Internet, the semantic web, open data, social networks and their connections, information streams such as twitter are examples of such datagraphs. An axis of LINKS is to propose methods and tools so as to extract information from datagraphs. We dwell in a wide spectrum of tools to construct these methods: circuits, compilation, optimization, logic, automata, machine learning. Our goal is to extend the kinds of information that can be extracted from datagraphs while improving the efficiency of existing ones.

This axis is split within two themes. The first one pertains to the use of low level representations by means of circuits to compute efficiently complex numerical aggregates that will find natural applications in AI. The second one proposes to explore path oriented query language and more particularly their efficient evaluation by means of efficient compilation and machine learning methods so as to have manageable statistics.

### 3.2.1 AI: Circuits for Data Analysis

Circuits are concise representations of data sets that recently found a unifying interest in various areas of artificial intelligence. A circuit may for instance represent the answer set of a database query as a dag whose operators are disjoint unions (for disjunction) and Cartesian products (for conjunction). Similarly, it may also represent the set of all matches of a pattern in a graph. The structure of the circuit may give rise to efficient algorithms to process large data sets based on representation that are often much smaller. Among others, such applications range from knowledge representation/compilation, counting the number of solutions of queries, efficient query answering, factorized databases.

In a first line of research, we want to study novel problems on circuits, in which database queries are relevant to data analysis tasks from artificial intelligence, in machine learning or data mining in particular. In particular we propose to study optimization problems on answer sets of database queries based on circuits, i.e. how to find optimal solutions in the answer set for a given set of conditions. Decompressing small circuits into large answer sets would make the optimization problem unfeasible in many cases. We believe that circuits can structure certain optimization problems in such a way that it can be phrased concisely and then solved efficiently.

Second, we propose to develop a tighter integration between circuits and databases. Indeed query-related circuits are generally produced from a database. This requires that the data is copied within the circuits. This memory cost is accompanied with the loss of the environment of the DBMS which allows many optimizations and uses many low level optimizations that are hard to implement. We propose then to encode circuits directly within the database using materialized views and index structures. We shall also develop the required computational tools for maintaining and exploiting these embedded circuits.

### 3.2.2 Path Query Optimization

Graph databases are easily queried using path descriptions. Most often these paths are described by means of regular expressions. This makes path queries difficult as the use of Kleene star makes them recursive. In relational DBMS, recursion is almost never used and it is not advised to use it. The natural theoretical tool that pertains to recursion in the context of relational data Datalog. There has been a wealth of optimization algorithms that have been proposed for queries written in Datalog. We propose to use Datalog as a low level language to which we will compile path queries of various kinds. The idea is that the compiler will try to obtain Datalog programs that will have low execution complexity taking advantage



of optimization techniques such as magic supplementary set rewriting, pre-computed indexes and also statistics computed from the graph. Our goal is to develop a compiler that will be able to efficiently evaluate path queries on large graphs which in particular will explore only a part of it.

### 3.3 Research Axis: Monitoring Data Graphs

Traditional database applications are programs that interact with database via updates and queries. We are interested in developing programming language techniques so as to interact with datagraphs rather than with traditional relational databases. Moreover, we shall take into account the dynamic aspects of datagraphs which shall evolve through updates. We will develop methods to monitor changes in datagraphs and react according to the modifications.

#### 3.3.1 Functional Programming Languages for Data Graphs

The first question is which kind of programming language to use to enable monitoring processes for data graphs based on query answering. While languages of path queries found quite some interest on data graphs, less attention has been given to the programming language tasks, that needed to be solved to produce structured output and to compose various queries with structured output into a pipeline. We believe that transferring the generalization of ideas developed for data trees in the context of XML to data graphs will allow to solve such problems in a systematic manner.

Our approach will consist in developing a functional programming language based on first principles (the lambda calculus, graph navigation, logical connective) that generalizes full XPath 3.0 to the context of graphs. Here we can rely on our previous work for data trees, such as the language X-Fun and  $\lambda$ -XP. After the language for data graphs is designed we shall study its behavior empirically by means of an implementation. This implementation will help us to design optimization methods so as to evaluate the queries in that language. This will allow us to use a wealth of techniques so as to optimize the computation. Indeed, we can try to compile data structures to imperative ones when possible and also exploit possibilities of parallel executions in certain cases. Functional programming comes with nice verification techniques that we are going to use in several contexts: (i) in optimizing queries (e.g. stop the evaluation when it is possible to know that no more data can contribute to the output) and (ii) to verify that the query behaves correctly. The verification methods we shall focus on will be mainly related to automata and transducers.

Finally we shall also develop a programming language that allows to describe services that use datagraphs as a backend for storing data. Here again, functional programming seems a good candidate, we would need however to orchestrate the concurrent executions of queries so as to ensure the correct behavior of services. This means that we should have concurrent constructs that are built in the language. The high level of concurrence enabled by the notion of *futures* seems an interesting candidate to adapt to the context of service orchestration.

#### 3.3.2 Hyperstreaming Program Evaluation

Complex-event processing requires to monitor data graphs that are produced on input streams and to write data graphs to some output stream, which can then be used as inputs again. A major problem here is to reduce the high risk of blocking, which arises when the writing of some of the output stream suspends on a data value that will become available only in the future on some input stream. In such cases, all monitoring processes reading the output stream may have to suspend as well. In order to reduce the risk of blocking, we propose to develop the hyperstreaming approach further, of which we laid the foundations in the evaluation period based on automata techniques. The idea is to generalize streams to hyperstreams, i.e. to add holes to streams that can be filled by some other stream in the future. In order to avoid suspension as possible, a monitoring process on hyperstream must then be able to jump over the holes, and to perform some speculative computation. The objectives for the next period are to develop tools for hyperstreaming query answering and to lift these to hyperstreaming program evaluation. Furthermore, on the conceptual side, the notion of certain query answers on hyperstreams needs to be lifted to certain program outputs on hyperstreams.

### 3.4 Research Axis: Graph Data Integration

We intend to continue to develop tools for integration of linked data with RDF being their principal format. Because from its conception the main credo of RDF has been “just publish your data”, the problem at hand faces two important challenges: data quality and data heterogeneity.

#### 3.4.1 Data Quality with Schemas and Repairing with Inference

The data quality of RDF may suffer due to a number of reasons. Impurities may arise due to data value errors (misspellings, errors during data entry etc.). Such data quality problems have been thoroughly investigated in literature for relational databases and solutions include dictionary methods etc. However, it remains to be seen if the challenges of adapting the existing solutions for relational databases can be easily addressed.

One particular challenge comes from the fact that RDF allows a higher degree of structural freedom in how information is represented as opposed to relational databases, where the choice is strongly limited to flat tables. We plan to investigate suitability of existing data cleaning methods to tackle the problems of data value impurities in RDF. The structural freedom of RDF is a source of data quality issues on its own. With the recent emergence of schema formalisms for RDF, it becomes evident that significant parts of existing RDF repositories do not necessarily satisfy schemas prepared by domain experts.

In the first place, we intend to investigate defining suitable measures of quality for RDF documents. Our approaches will be based on a schema language, such as ShEx and SHACL, and we shall explore suitable variants of graph alignment and graph edit distance to capture similarity between the existing RDF document and its possible repaired versions that satisfy the schema.

The central issue here is repairing an RDF document w.r.t. schema by identifying essential fragments of the RDF that fail to satisfy the schema. Once such fragments are identified, repairing actions can be applied however there might be a significant number of alternatives. We intend to explore enumeration approaches where the space of repairing alternatives is intelligently browsed by the user and the most suitable one is chosen. Furthermore, we intend to propose a rule language for choosing the most suitable repairing action and will investigate inference methods to derive from interactions with user the optimal order in which various repairing actions are presented to the user and derive the rules for the choice of the preferred repairing action for repeating types of fragments that do not satisfy the schema.

#### 3.4.2 Integration and Graph Mappings with Schemas and Inference

The second problem pertaining to integration of RDF data sources is their heterogeneity. We intend to continue to identify and study suitable classes of mappings between RDF documents conforming to potentially different and complementary schemas. We intend to assist the user in constructing such mappings by developing rich and expressive graphical languages for mappings. Also, we wish to investigate inference of RDF mappings with the active help of an expert user. We will need to define interactive protocols that allows the input to be sufficiently informative to guide the inference process while avoiding the pitfalls of user input being too ambiguous and causing combinatorial explosion. We intend to identify

RDF Data Quality. Approaches based on a schema language (ShEx or SHACL) are used to identify errors and to give a notion of a measure of quality of an RDF database. Impurities in RDF may come from data value errors (misspellings etc.) but also from the fact that RDF imposes fewer constraints on how data is structured which is a consequence of a significantly different use philosophy (just publish your data anyway you want). Repairing of RDF errors would be modeled with localized rules (transformations that operate within a small radius of an affected node) and if several rules apply, preferences are used to identify the most desirable one. Both the repairing rules and preferences can be inferred with the help of inference algorithms in an interactive setting. Smart tools for LOD integration. Assuming that the LOD sources are of good quality, we want to build tools that assist the user in constructing mappings that integrate data in the user database. For this, we want to define inference algorithms which are guided by schemas, and which are based on comprehensible interactions with the user. For this, we need to define interactions that are rich enough to inform the algorithm, while simple enough to be understandable by a non-expert user. In particular, that means that we need to present data (nodes in a graph for instance)

in a readable way. Also, we want to investigate how the - possibly inferred - schema can be used to guide the inference.

## 4 Application domains

### 4.1 Linked data integration

There are many contexts in which integrating linked data is interesting. We advocate here one possible scenario, namely that of integrating business linked data to feed what is called Business Intelligence. The latter consists of a set of theories and methodologies that transform raw data into meaningful and useful information for business purposes (from Wikipedia). In the past decade, most of the enterprise data was proprietary, thus residing within the enterprise repository, along with the knowledge derived from that data. Today's enterprises and businessmen need to face the problem of information explosion, due to the Internet's ability to rapidly convey large amounts of information throughout the world via end-user applications and tools. Although linked data collections exist by bridging the gap between enterprise data and external resources, they are not sufficient to support the various tasks of Business Intelligence. To make a concrete example, concepts in an enterprise repository need to be matched with concepts in Wikipedia and this can be done via pointers or equalities. However, more complex logical statements (i.e. mappings) need to be conceived to map a portion of a local database to a portion of an RDF graph, such as a subgraph in Wikipedia or in a social network, e.g. LinkedIn. Such mappings would then enrich the amount of knowledge shared within the enterprise and let more complex queries be evaluated. As an example, businessmen with the aid of business intelligence tools need to make complex sentimental analysis on the potential clients and for such a reason, such tools must be able to pose complex queries, that exploit the previous logical mappings to guide their analysis. Moreover, the external resources may be rapidly evolving thus leading to revisit the current state of business intelligence within the enterprise.

### 4.2 Data cleaning

The second example of application of our proposal concerns scientists who want to quickly inspect relevant literature and datasets. In such a case, local knowledge that comes from a local repository of publications belonging to a research institute (e.g. HAL) need to be integrated with other Web-based repositories, such as DBLP, Google Scholar, ResearchGate and even Wikipedia. Indeed, the local repository may be incomplete or contain semantic ambiguities, such as mistaken or missing conference venues, mistaken long names for the publication venues and journals, missing explanation of research keywords, and opaque keywords. We envision a publication management system that exploits both links between database elements, namely pointers to external resources and logical links. The latter can be complex relationships between local portions of data and remote resources, encoded as schema mappings. There are different tasks that such a scenario could entail such as (i) cleaning the errors with links to correct data e.g. via mappings from HAL to DBLP for the publications errors, and via mappings from HAL to Wikipedia for opaque keywords, (ii) thoroughly enrich the list of publications of a given research institute, and (iii) support complex queries on the corrected data combined with logical mappings.

### 4.3 Real-time complex event processing

Complex event processing serves for monitoring nested word streams in real time. Complex event streams are gaining popularity with social networks such as with Facebook and Twitter, and thus should be supported by distributed databases on the Web. Since this is not yet the case, there remains much space for future industrial transfer related to LINKS' second axis on dynamic linked data.

## 5 Social and environmental responsibility

### 5.1 Footprint of research activities

**Amarilli** is environmental chair in the steering committee of the informal conference “Highlights of Logic, Games, and Automata” (since 2024).

**Boneva** is the leader of the Sustainable Development Commission of the CRISAL research lab. Amarilli is a co-animator of the **TCS4F pledge** on sustainable research in theoretical computer science (since 2020).

### 5.2 Women in science

**Tison** Member of the steering committee of the *Programme de mentorat Femmes et Sciences de Lille*.

### 5.3 Impact of research results

Databases and methods from Artificial Intelligence are used in virtually all aspects of the modern digitalized world, from companies' web services to governments' institutions.

## 6 Highlights of the year

### 6.1 Awards

The paper [21] received the *Best Newcomer Paper Award* at the ICDT 2024 conference.

### 6.2 Evaluations

During the year 2024, Links has been evaluated both by INRIA and the HCERES during the evaluation of the laboratory CRISAL. We are proud that all these evaluations have been very positive.

## 7 New software, platforms, open data

### 7.1 New software

#### 7.1.1 NetworkDisk

**Name:** NetworkDisk

**Keywords:** Large graphs, Python, Databases

**Functional Description:** NetworkDisk provides a way to manipulate graphs on disk. The goal is to be as much as possible compatible with (Di)Graph objects of the NetworkX Python package but lifting memory requirement and providing persistence of the Graph.

**URL:** <https://networkdisk.inria.fr/>

**Contact:** Charles Paperman

#### 7.1.2 Bibendum

**Name:** Bibendum

**Keyword:** Bibliography

**Functional Description:** Small app to fetch bibtex from a short label with the format: LastName.Year.PublicationTerm where the . denote the concatenation. For instance Codd1970Relational. LastName is the last name of one of the authors. Year is the publication year. PublicationTerm is one meaningful word in the title of the publication.

In case of ambiguity, an extra integer is used. Ambiguous entries are resolved by sorting dois under lexicographical order. The api is idempotent, every decision taken is recorded and replayed.

**URL:** <https://bibendum.lille.inria.fr/>

**Contact:** Charles Paperman

### 7.1.3 XPath AutoBench

**Name:** A Benchmark Collection of Deterministic Automata for XPath Queries

**Functional Description:** We provide a benchmark collection of deterministic automata for regular XPath queries. For this, we select the subcollection of forward navigational XPath queries from a corpus that Lick and Schmitz extracted from real-world XSLT and XQuery programs, compile them to step-wise hedge automata (SHAs), and determinize them. Large blowups by automata determinization are avoided by using schema-based determinization. The schema captures the XML data model and the fact that any answer of a path query must return a single node. Our collection also provides deterministic nested word automata that we obtain by compilation from deterministic SHAs.

**URL:** [https://archive.softwareheritage.org/browse/origin/directory/?origin\\_url=https://gitlab.inria.fr/aalserha/xpath-benchmark](https://archive.softwareheritage.org/browse/origin/directory/?origin_url=https://gitlab.inria.fr/aalserha/xpath-benchmark)

**Contact:** Joachim Niehren

### 7.1.4 rsonpath

**Keywords:** JSON, Streaming, SIMD, Rust

**Functional Description:** The rsonpath crate provides a JSONPath parser and a query execution engine, which utilizes SIMD instructions to provide massive throughput improvements over conventional engines.

**URL:** <https://github.com/Voldek/rsonpath>

**Contact:** Charles Paperman

**Partner:** Warsaw University

### 7.1.5 Coussinet

**Name:** Coussinet

**Keywords:** Enumeration, Complexity

**Functional Description:** Coussinet is a demo illustrating a technique called geometric amortization for enumeration algorithms introduced in the paper Geometric Amortization for Enumeration Algorithms, Florent Capelli, Yann Strozecki. The result presented in this paper is about making the delay of enumeration algorithms more regular.

**URL:** <http://florent.capelli.me/coussinet/coussinet.html>

**Contact:** Florent Capelli

**Participants:** Florent Capelli, Yann Strozecki

### 7.1.6 ShEx validator

**Name:** Validation of Shape Expression schemas

**Keywords:** Data management, RDF

**Functional Description:** Shape Expression schemas is a formalism for defining constraints on RDF graphs. This software allows to check whether a graph satisfies a Shape Expressions schema.

**Release Contributions:** ShExJava now uses the Commons RDF API and so support RDF4J, Jena, JSON-LD-Java, OWL API and Apache Clerezza. It can parse ShEx schema in the ShEcC, ShEJ, ShExR formats and can serialize a schema in ShExJ.

To validate data against a ShExSchema using ShExJava, you have two different algorithms: - the refine algorithm: compute once and for all the typing for the whole graph - the recursive algorithm: compute only the typing required to answer a validate(node,ShapeLabel) call and forget the results.

**URL:** <https://github.com/iovka/shex-java>

**Contact:** Iovka Boneva

**Partner:** CRISTAL

### 7.1.7 gMark

**Name:** gMark: schema-driven graph and query generation

**Keywords:** Semantic Web, Data base

**Functional Description:** gMark allow the generation of graph databases and an associated set of query from a schema of the graph.gMark is based on the following principles: - great flexibility in the schema definition - ability to generate big size graphs - ability to generate recursive queries - ability to generate queries with a desired selectivity

**URL:** <https://github.com/graphMark/gmark>

**Contact:** Aurélien Lemay

## 8 New results

**Participants:** Antonio Al Serhali, Corentin Barloy, Iovka Boneva, Bastien Desgardin, Oliver Irwin, Aurélien Lemay, Mikael Monet, Margaux Mouton, Joachim Niehren, Charles Paperman, Sylvain Salvati, Sophie Tison.

### 8.1 Circuits for data manipulation

#### 8.1.1 Circuits and knowledge compilation

In their SIGMOD Record paper [30] Amarilli and Capelli review how database theory uses tractable circuit classes from knowledge compilation. They present relevant query evaluation tasks, and notions of tractable circuits. They then show how these tractable circuits can be used to address database tasks. They first focus on Boolean provenance and its applications for aggregation tasks, in particular probabilistic query evaluation. They study these for Monadic Second Order (MSO) queries on trees, and for safe Conjunctive Queries (CQs) and Union of Conjunctive Queries (UCQs). They also study circuit representations of query answers, and their applications to enumeration tasks: both in the Boolean setting (for MSO) and the multivalued setting (for CQs and UCQs).

In their paper [21] Capelli and Irwin study the direct access problem for conjunctive queries with negations. Given a conjunctive query  $Q$  and a database  $D$ , a direct access to the answers of  $Q$  over  $D$

is the operation of returning, given an index  $j$ , the  $j$ th answer for some order on its answers. While this problem is #P-hard in general with respect to combined complexity, many conjunctive queries have an underlying structure that allows for a direct access to their answers for some lexicographical ordering that takes polylogarithmic time in the size of the database after a polynomial time precomputation. Previous work has precisely characterised the tractable classes and given fine-grained lower bounds on the precomputation time needed depending on the structure of the query. In this paper, they generalise these tractability results to the case of signed conjunctive queries, that is, conjunctive queries that may contain negative atoms. Their technique is based on a class of circuits that can represent relational data. They first show that this class supports tractable direct access after a polynomial time preprocessing. We then give bounds on the size of the circuit needed to represent the answer set of signed conjunctive queries depending on their structure. Both results combined together allow them to prove the tractability of direct access for a large class of conjunctive queries. On the one hand, they recover the known tractable classes from the literature in the case of positive conjunctive queries. On the other hand, they generalise and unify known tractability results about negative conjunctive queries – that is, queries having only negated atoms. In particular, they show that the class of  $\beta$ -acyclic negative conjunctive queries and the class of bounded nest set width negative conjunctive queries admit tractable direct access.

The report [25] by Amarilli, Arenas, Choi, Monet, Van den Broeck and Wang is an introduction to two related formalisms to define Boolean functions: binary decision diagrams, and Boolean circuits. It presents these formalisms and several of their variants studied in the setting of knowledge compilation. Last, it explains how these formalisms can be connected to the notions of automata over words and trees.

The report [28] by Capelli, Irwin and Salvati presents an elementary branch and bound algorithm with a simple analysis of why it achieves worstcase optimality for join queries on classes of databases defined respectively by cardinality or acyclic degree constraints. It then shows that if one is given a reasonable way for recursively estimating upper bounds on the number of answers of the join queries, the algorithm can be turned into an algorithm for uniformly sampling answers with expected running time  $\tilde{O}(UP/OUT)$  where  $UP$  is the upper bound,  $OUT$  is the actual number of answers and  $\tilde{O}(\cdot)$  ignores polylogarithmic factors. The approach recovers recent results on worstcase optimal join algorithm and sampling in a modular, clean and elementary way.

### 8.1.2 Foundation of circuits: complexity, algebra and abstract machines

In the note [27], Amarilli, Monet and Suciu present a conjecture on intersections of set families, and a rephrasing of the conjecture in terms of principal downsets of Boolean lattices. The conjecture informally states that, whenever we can express the measure of a union of sets in terms of the measure of some of their intersections using the inclusion-exclusion formula, then we can express the union as a set from these same intersections via the set operations of disjoint union and subset complement. The note also presents a partial result towards establishing the conjecture.

## 8.2 Logic and query evaluation

### 8.2.1 Provenance, explanation, aggregation, counting, uncertainty, probabilistic data, Shapley, approximation algorithms

In their LMCS paper [18] Capelli, Corsetti, Niehren and Ramon study the problem of optimizing a linear program whose variables are the answers to a conjunctive query. For this they propose the language LP(CQ) for specifying linear programs whose constraints and objective functions depend on the answer sets of conjunctive queries. They contribute an efficient algorithm for solving programs in a fragment of LP(CQ). The natural approach constructs a linear program having as many variables as there are elements in the answer set of the queries. The approach constructs a linear program having the same optimal value but fewer variables. This is done by exploiting the structure of the conjunctive queries using generalized hypertree decompositions of small width to factorize elements of the answer set together. They illustrate the various applications of LP(CQ) programs on three examples: optimizing deliveries of resources, minimizing noise for differential privacy, and computing the  $s$ -measure of patterns in graphs as needed for data mining.

In their paper [19] Karmakar, Monet, Senellart and Bressan study Shapley-like scores in connection to probabilistic databases. Shapley values, originating in game theory and increasingly prominent in

explainable AI, have been proposed to assess the contribution of facts in query answering over databases, along with other similar power indices such as Banzhaf values. In this work they adapt these Shapley-like scores to probabilistic settings, the objective being to compute their expected value. They show that the computations of expected Shapley values and of the expected values of Boolean functions are interreducible in polynomial time, thus obtaining the same tractability landscape. They investigate the specific tractable case where Boolean functions are represented as deterministic decomposable circuits, designing a polynomial-time algorithm for this setting. They present applications to probabilistic databases through database provenance, and an effective implementation of this algorithm within the ProVSQL system, which experimentally validates its feasibility over a standard benchmark.

The report [26] by Amarilli, Gatterbauer, Makhija and Monet is about resilience for regular path queries. The resilience problem for a query and an input set or bag database is to compute the minimum number of facts to remove from the database to make the query false. In this paper, they study how to compute the resilience of Regular Path Queries (RPQs) over graph databases. Their goal is to characterize the regular languages  $L$  for which it is tractable to compute the resilience of the existentially-quantified RPQ built from  $L$ . They show that computing the resilience in this sense is tractable (even in combined complexity) for all RPQs defined from so-called local languages. By contrast, they show hardness in data complexity for RPQs defined from the following language classes (after reducing the languages to eliminate redundant words): all finite languages featuring a word containing a repeated letter, and all languages featuring a specific kind of counterexample to being local (which they call four-legged languages). The latter include in particular all languages that are not star-free. Their results also imply hardness for all non-local languages with a so-called neutral letter. They also highlight some remaining obstacles towards a full dichotomy. In particular, for the RPQ  $abc|be$ , resilience is tractable but the only PTIME algorithm that they know uses submodular function optimization.

### 8.2.2 Efficient evaluation: streaming and parallelism

In their paper in the Journal Algorithms [17] Al Serhali and Niehren demonstrate how to evaluate stepwise hedge automata (Shas) with subhedge projection while completely projecting irrelevant subhedges. Since this requires passing finite state information top-down, they introduce the notion of downward stepwise hedge automata. They use them to define in-memory and streaming evaluators with complete subhedge projection for Shas. They then tune the evaluators so that they can decide on membership at the earliest time point. They apply our algorithms to the problem of answering regular XPath queries on Xml streams. Their experiments show that complete subhedge projection of Shas can indeed speed up earliest query answering on Xml streams so that it becomes competitive with the best existing streaming tools for XPath queries.

In their paper [22] Gienieccko, Murlak and Paperman present a vectorized implementation for a fragment of JSONPath. Harnessing the power of SIMD can bring tremendous performance gains in data processing. In querying streamed JSON data, the state of the art leverages SIMD to fast forward significant portions of the document. However, it does not provide support for descendant, which excludes many real-life queries and makes formulating many others hard. In this work, they aim to change this: they consider the fragment of JSONPath that supports child, descendant, wildcard, and labels. They propose a modular approach based on novel depth-stack automata that process a stream of events produced by a state-driven classifier, allowing fast forwarding parts of the input document irrelevant at the current stage of the computation. They implement their solution in Rust and compare it with the state of the art, confirming that our approach allows supporting descendants without sacrificing performance, and that reformulating natural queries using descendants brings impressive performance gains in many cases.

### 8.2.3 Enumerating the results of queries

In their paper [20] Amarilli, Bourhis, Capelli and Monet study the problem of enumerating the satisfying assignments for circuit classes from knowledge compilation, where assignments are ranked in a specific order. In particular, they show how this problem can be used to efficiently perform ranked enumeration of the answers to MSO queries over trees, with the order being given by a ranking function satisfying a subset-monotonicity property. Assuming that the number of variables is constant, they show that they can enumerate the satisfying assignments in ranked order for so-called multivalued circuits that



are smooth, decomposable, and in negation normal form (smooth multivalued DNNF). There is no preprocessing and the enumeration delay is linear in the size of the circuit times the number of values, plus a logarithmic term in the number of assignments produced so far. If they further assume that the circuit is deterministic (smooth multivalued d-DNNF), they can achieve linear-time preprocessing in the circuit, and the delay only features the logarithmic term.

### 8.3 Knowledge on Data

#### 8.3.1 Optimization for schemas for graphs

The report [29] written by Degardins, Mouton, Guillon, Paperman and Marchet is about Vizitig, a novel visualization tool designed for the exploration of colored de Bruijn graphs, a structure increasingly used for comparative genomics, pangenomics, and metagenomics. Unlike existing tools such as Bandage, Vizitig supports the simultaneous visualization of multiple genomes or samples by leveraging color-coding schemes to highlight shared and unique sequences or meta-data. Vizitig allows users to efficiently manage, navigate, and render graphs annotated with various metadata, such as genomic features or sample-specific information. Developed as a cross-platform Python application, Vizitig integrates a user-friendly web interface and a command-line-free environment, making it accessible to a broader audience of genomic researchers.

#### 8.3.2 Static analysis with constraints (open-world query answering, query rewriting under constraints, consistency, certain answers)

In an ITCS'25 article [24], in collaboration with Benoît Groz (Université Paris-Saclay) and Nicole Wein (University of Michigan), Amarilli studies the question of finding paths in graphs that are edge-minimal and satisfy length modularity conditions. In other words, we want a path from a source vertex to a sink vertex that obeys a modularity constraint (i.e., having length  $p \bmod q$ ) and uses the least number of distinct edges. This is a first step towards understanding the same problem for regular path queries on graphs, i.e., finding which paths depend on the least number of distinct edges if the underlying graph is uncertain.

In their paper [23] Salvati and Tison study the containment problem of regular path queries under path constraints. Data integrity is ensured by expressing constraints it should satisfy. One can also view constraints as data properties and take advantage of them for several tasks such as reasoning about data or accelerating query processing. In the context of graph databases, simple constraints can be expressed by means of path constraints while simple queries are modeled as regular path queries (RPQs). In this paper, they investigate the containment of RPQs under path constraints. They focus on word constraints that can be viewed as tuple-generating dependencies (TGDs) of the form  $\forall x_1, x_2, \exists y_1 \dots y_{n-1}, a_1(x_1, y_1) \wedge \dots \wedge a_i(y_{i-1}, y_i) \wedge \dots \wedge a_n(y_{n-1}, x_2) \Rightarrow \exists z_1 \dots z_{m-1}, b_1(x_1, z_1) \wedge \dots \wedge b_i(z_{i-1}, z_i) \wedge \dots \wedge b_m(z_{m-1}, x_2)$ . Such a constraint means that whenever two nodes in a graph are connected by a path labeled  $a_1 \dots a_n$ , there is also a path labeled  $b_1 \dots b_m$  that connects them. Rewrite systems offer an abstract view of these TGDs: the rewrite rule  $a_1 \dots a_n \rightarrow b_1 \dots b_m$  represents the previous constraint. A set of constraints  $\mathcal{C}$  is then represented by a rewrite system  $R$  and, when dealing with possibly infinite databases, a path query  $p$  is contained in a path query  $q$  under the constraints  $\mathcal{C}$  iff  $p$  rewrites to  $q$  with  $R$ . Contrary to what has been claimed in the literature, they show that, when restricting to finite databases only, there are cases where a path query  $p$  is contained in a path query  $q$  under the constraints  $\mathcal{C}$  while  $p$  does not rewrite to  $q$  with  $R$ . More generally, they study the finite controllability of the containment of RPQs under word constraints, that is when this containment problem on unrestricted databases does coincide with the finite case. They give an exact characterisation of the cases where this equivalence holds. They then deduce the undecidability of the containment problem in the finite case even when RPQs are restricted to word queries. They prove several properties related to finite controllability, and in particular that it is undecidable. They also exhibit some classes of word constraints that ensure the finite controllability and the decidability of the containment problem.

### 8.4 Highlighted results

We highlight the following results as they constitute key advances in the overall project of Links.

- The paper [21] by Capelli and Irwin, *Direct Access for Conjunctive Queries with Negations*, shows how to use circuits to represent query results so as to enable direct access the nth result. Furthermore, this work allows for the use of negations in the query and contributes to the understanding of the treatment of negation in query resolution.
- The paper [22] by Gienieccko, Murlak and Paperman, *Supporting Descendants in SIMD-Accelerated JSONPath*, brings a deep theoretical result published earlier by the team to an impressive implementation. The software `rsonpath` solves extremely efficiently a fragment of JSONPath by combining deep algebraic analyses of queries and intense use of CPU parallelization capabilities, SIMD instructions. This article embodies the kind of applications the team is aiming.
- The paper [19] by Karmakar, Monet, Senellart and Bressan, *Expected Shapley-Like Scores of Boolean Functions: Complexity and Applications to Probabilistic Databases*. Shapley-like scores allow to understand the contribution of databases facts in the computation of expected values. This paper connects databases methods with explainable AI, uses circuits in the background and proposes an implementation.

## 9 Partnerships and cooperations

### 9.1 International research visitors

#### 9.1.1 Visits of international scientists

##### Other international visits to the team

###### **Michaël Cadilhac**

**Status** Researcher

**Institution of origin:** DePaul University

**Country:** United States

**Dates:** 24/06 to 06/07

**Context of the visit:** PhD defence of Corentin Barloy

**Mobility program/type of mobility:** research stay

###### **Howard Straubing**

**Status** Researcher

**Institution of origin:** Boston College

**Country:** United States

**Dates:** 24/06 to 08/07

**Context of the visit:** PhD defence of Corentin Barloy

**Mobility program/type of mobility:** research stay

## 9.2 National initiatives

### ANR JCJC KCODA

**Participants:** Florent Capelli (*correspondent*), Charles Paperman, Sylvain Salvati.

- **Duration:** 2021–2025
- **Objectives:** The aim of KCODA is to study how succinct representations can be used to efficiently solve modern optimization and AI problems that use a lot of data. We suggest using data structures from the field of compilation of knowledge that can represent large datasets succinctly by factoring certain parts while allowing efficient analysis of the represented data. The first goal of KCODA is to understand how one can efficiently solve optimization and training problems for data represented by these structures. The second goal of KCODA is to offer better integration of these techniques into the systems of database management by proposing new algorithms allowing to build factorized representations of the data responses to DB requests and by proposing encodings of these representations inside the DB.

### ANR SxS

**Participants:** Charles Paperman (*PI*), Sylvain Salvati.

- **Duration:** 2025–2030
- **Coordinator:** Charles Paperman
- **Scientific Partner:** LIP (Lyon) and LCIS (University Grenoble Alpes) and LABRI (Bordeaux University)
- **Objective:** Knowledge about handcrafted vectorization algorithms is scattered among various fields, system architectures, programming languages paradigms and industrial pieces of codes. The main objective of the project is to bring structure to this knowledge through novel conceptual and practical tools. The end goal is to facilitate the writing of SIMD programs and to pave the way to future auto-vectorization methods for stream processing. The main difficulty to achieve these goals is to understand the interplay of sequentiality and bit-level parallelism. Circuit complexity, as introduced by Shannon in the early years of computer science, offers a powerful framework to study such notion of parallelism. In this multidisciplinary project, we bring together methodologies from the fields of circuit complexity of automata, and programming language design for parallel architectures, with motivations from data processing and bioinformatics applications.

The project's results will be implemented into a compilation tool-chain. At the highest level, we will design and develop a domain specific language, dubbed Vectoid. Its goal is to allow more programmers to harness SIMD instructions. Vectoid will be compiled to a Vectorial Intermediate Representation (VIR) dedicated to stream processing. In turn VIR will be compiled to low level code for various SIMD architectures. The design of this compilation tool-chain will be informed by a theoretical investigation on the expressivity of vectorical circuits. It will be evaluated on data processing and bioinformatics benchmarks.

## 9.3 Regional initiatives

Amarili, Monet and Salvati have obtained a PhD grant from région Nord via the CPER Cornelia. Unfortunately, the PhD student that had been recruited finally resigned before starting the PhD.

## 10 Dissemination

**Participants:** Antoine Amarilli, Antonio Al Serhali, Iovka Boneva, Oliver Irwin, Aurélien Lemay, Charles Paperman, Mikaël Monet, Sylvain Salvati, Sophie Tison.

### 10.1 Promoting scientific activities

#### Member of the conference program committees

**Amarilli** Member of the program committee of ICDT 2025.

**Boneva** Member of the program committee of BDA 2024.

**Boneva** Member of the program committee of AMW 2024.

**Monet** Member of the program committee of PODS'2025.

**Salvati** Member of the program committee of Coling 2025.

#### 10.1.1 Journal

##### Member of the editorial boards

**Monet** Managing editor for the TheoretiCS journal.

#### 10.1.2 Invited talks

**Salvati** was invited to give a talk at the joint reunion of the working groups SCALP and VERiF of the GDR IFM.

#### 10.1.3 Leadership within the scientific community

**Amarilli** is vice-president of the EATCS (elected in 2024).

**Paperman** Elected member of EATCS (European Association for Theoretical Computer Sciences).

**Tison** Member of the scientific committee of the GDR IM.

**Tison** Member of the general assembly of the European Association for Theoretical Computer Science (EATCS) (2019-2024).

**Tison** Member of the SSAAL *Société des Sciences, de l'Agriculture et des Arts de Lille*.

#### 10.1.4 Scientific expertise

**Tison** Member of one HCERES committee (LIX).

**Tison** Expert for the price *L'Oréal/Unesco/Académie des Sciences Jeunes Talents France*

#### 10.1.5 Research administration

**Boneva** Member of the steering committee of BDA (French association for research in databases).

**Boneva** Member of Inria Lille CER (Commission des Emplois de Recherche).

**Paperman** Elected member of the research commission of FST (Université de Lille).

**Tison** Member of the Steering Committee of Highlights of Logic, Automata, and Games.

## 10.2 Teaching - Supervision - Juries

### 10.2.1 Supervision

**Antonio Al Serhali** PhD project started in 2020. On hyperstream programming. Supervised by Niehren. Defended in December 2024.

**Bastien Desgardin** PhD projet started in October 2024. On graph databases for DNA analysis. Supervised by Paperman and Marchet (CRIStAL Bonsai).

**Oliver Irwin** PhD project started in 2022. On compilation and aggregation in databases. Co-supervised by Capelli and Salvati.

### 10.2.2 PhD defended

**Corentin Barloy** has defended his PhD entitled *On the complexity of regular languages* on July 5 2024.

**Antonio Al Serhali** has defended his PhD entitled *Evaluation au plus tôt de requêtes régulières avec projection de sous-haies complète* on December 12 2024.

### 10.2.3 HDR defended

**Charles Paperman** has defended his Habilitation entitled *La théorie des semigroupes pour l'algorithmique, la complexité et la compilation* on October 10 2024.

### 10.2.4 Juries

**Paperman** Member of the jury for oral examination in Theoretical Computer Science at ENS Paris (2021-2024)

**Tison** Member of the PhD committees of Isa Vialard (Saclay, rewiever), C. Barloy (Lille), A. Al Sherahli (Lille)

### 10.2.5 Teaching Responsibilities

**Lemay** Responsible for computer science and numeric correspondent for his department.

**Paperman** Responsible of *alternance* for the department of Computer Science.

**Salvati** Head of Master of Computer Science at the University of Lille.

**Salvati** Member of the board for Computer Science of the doctoral school MADIS (Mathematics and Computer Science).

**Salvati** Member of the *collège doctoral* (doctoral college) of University of Lille: PhD students education and professional training.

**Salvati** Elected member of the Computer Science Department Council.

### 10.2.6 Teaching Activities

**Al Serhali** Taught 192 hours in Computer Science in the Computer Science Department of Université de Lille.

**Boneva** Teaches computer science in DUT Informatique of Université de Lille.

**Capelli** (Until September 2024) Teaches computer science in the LEA department of Université de Lille for around 200h per year (Licence and Master).

**Irwin** Taught 64 hours in Computer Science in the Computer Science Department of Université de Lille.

**Lemay** Teaches computer science in the LEA department of Université de Lille for around 200h per year (Licence and Master). He is also responsible for computer science and numeric correspondent for its department.

**Monet** Monet teaches computer science as a temporary lecturer at Université de Lille and at Centrale Lille. He teaches two databases courses, as well as a course on "algorithms and complexity".

**Niehren** M2 Machine Learning, Université de Lille 2022-2023, Fondaments Theorique des Bases de Données.

**Paperman** Teaches CS in master degrees of the computer science department, Miashs at bachelor level in the math department and in the data science master degree.

**Salvati** Teaches computer science for a total of around 230h per year in computer science department of Université de Lille.

## 10.3 Popularization

### 10.3.1 Specific official responsibilities in science outreach structures

**Tison** Member of the scientific committee of the *Maison Pour la Science*.

## 11 Scientific production

### 11.1 Major publications

- [1] A. Amarilli, L. Jachiet and C. Paperman. ‘Dynamic Membership for Regular Languages’. In: ICALP. Vol. 48. International Colloquium on Automata, Languages, and Programming (ICALP 2021). Glasgow, Scotland, France, 2nd July 2021, 116:1–116:17. DOI: [10.4230/LIPIcs.ICALP.2021.116](https://doi.org/10.4230/LIPIcs.ICALP.2021.116). URL: <https://hal.archives-ouvertes.fr/hal-03466453>.
- [2] M. Arenas, P. Barceló, L. Bertossi and M. Monet. ‘The Tractability of SHAP-Score-Based Explanations over Deterministic and Decomposable Boolean Circuits’. In: *Thirty-Fifth AAAI Conference on Artificial Intelligence*. Held online, France, Feb. 2021. URL: <https://hal.inria.fr/hal-03147623>.
- [3] C. Barloy, M. Cadilhac, C. Paperman and T. Zeume. ‘The Regular Languages of First-Order Logic with One Alternation’. In: LICS 2022 - 37th Annual ACM/IEEE Symposium on Logic in Computer Science. Haïfa, Israel, 2nd Aug. 2022, pp. 1–11. DOI: [10.1145/3531130.3533371](https://doi.org/10.1145/3531130.3533371). URL: <https://hal.science/hal-03934389>.
- [4] C. Barloy, F. Murlak and C. Paperman. ‘Stackless Processing of Streamed Trees’. In: *2021 PODS*. Xi’an, Shaanx, China, June 2021. DOI: [10.4230/LIPIcs](https://doi.org/10.4230/LIPIcs). URL: <https://hal.archives-ouvertes.fr/hal-03021960>.
- [5] M. Benedikt, P. Bourhis and M. V. Boom. ‘Characterizing Definability in Decidable Fixpoint Logics’. In: *ICALP 2017 - 44th International Colloquium on Automata, Languages, and Programming*. Ed. by I. Chatzigiannakis, P. Indyk, F. Kuhn and A. Muscholl. Vol. 107. ICALP-2017 Best paper award of Track B. Varsovie, Poland, July 2017, p. 14. DOI: [10.4230/LIPIcs.ICALP.2017.107](https://doi.org/10.4230/LIPIcs.ICALP.2017.107). URL: <https://hal.inria.fr/hal-01639015>.
- [6] A. Boiret, V. Hugot, J. Niehren and R. Treinen. ‘Logics for Unordered Trees with Data Constraints’. In: *Journal of Computer and System Sciences* (Dec. 2018), p. 40. URL: <https://hal.inria.fr/hal-01176763>.
- [7] I. Boneva, J. G. Labra Gayo and E. G. Prud’hommeaux. ‘Semantics and Validation of Shapes Schemas for RDF’. In: *ISWC2017 - 16th International semantic web conference*. Vienna, Austria, Oct. 2017. URL: <https://hal.archives-ouvertes.fr/hal-01590350>.

- [8] P. Bourhis, M. Leclère, M.-L. Mugnier, S. Tison, F. Ulliana and L. Gallois. ‘Oblivious and Semi-Oblivious Boundedness for Existential Rules’. In: *IJCAI 2019 - International Joint Conference on Artificial Intelligence*. Macao, China, Aug. 2019. URL: <https://hal-lirmm.ccsd.cnrs.fr/lirmm-02148142>.
- [9] F. Capelli, N. Crosetti, J. Niehren and J. Ramon. ‘Linear Programs with Conjunctive Queries’. In: 25th International Conference on Database Theory (ICDT 2022). Edinburgh, United Kingdom, 29th Mar. 2022. URL: <https://hal.archives-ouvertes.fr/hal-01981553>.
- [10] F. Capelli, J.-M. Lagniez and P. Marquis. ‘Certifying Top-Down Decision-DNNF Compilers’. In: *Thirty-Fifth AAAI Conference on Artificial Intelligence*. Online, France, Feb. 2021. URL: <https://hal.inria.fr/hal-03111679>.
- [11] D. Debarbieux, O. Gauwin, J. Niehren, T. Sebastian and M. Zergaoui. ‘Early Nested Word Automata for XPath Query Answering on XML Streams’. In: *Theoretical Computer Science* 578 (Mar. 2015), pp. 100–127. URL: <https://hal.inria.fr/hal-00966625>.
- [12] P. D. Gallot, A. Lemay and S. Salvati. ‘Linear high-order deterministic tree transducers with regular look-ahead’. In: *MFCS 2020 : The 45th International Symposium on Mathematical Foundations of Computer Science*. Andreas Feldmann, Michal Koucky and Anna Kotesovcova. Prague, Czech Republic, Aug. 2020. DOI: 10.4230/LIPIcs.MFCS.2020.34. URL: <https://hal.archives-ouvertes.fr/hal-02902853>.
- [13] V. Hugot, A. Boiret and J. Niehren. ‘Equivalence of Symbolic Tree Transducers’. In: *DLT 2017 - Developments in Language Theory*. Vol. 105. Liege, Belgium, Aug. 2017, p. 12. DOI: 10.1007/978-3-642-29709-0\_32. URL: <https://hal.inria.fr/hal-01517919>.
- [14] J. Niehren and M. Sakho. ‘Determinization and Minimization of Automata for Nested Words Revisited’. In: *Algorithms* (Feb. 2021). URL: <https://hal.inria.fr/hal-03134596>.
- [15] C. Paperman, S. Salvati and C. Soyez-Martin. *An algebraic approach to vectorial programs*. 27th Oct. 2022. DOI: 10.4230/LIPIcs.STACS.2023.14. URL: <https://hal.archives-ouvertes.fr/hal-03831752>.
- [16] S. Staworko and P. Wiecek. ‘Containment of Shape Expression Schemas for RDF’. In: *SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS)*. Amsterdam, Netherlands, June 2019. URL: <https://hal.inria.fr/hal-01959143>.

## 11.2 Publications of the year

### International journals

- [17] A. Al Serhali and J. Niehren. ‘Complete Subhedge Projection for Stepwise Hedge Automata’. In: *Algorithms*. Selected Algorithmic Papers From FCT 2023 17.8 (2nd Aug. 2024). DOI: 10.3390/a17080339. URL: <https://inria.hal.science/hal-04421323> (cit. on p. 13).
- [18] F. Capelli, N. Crosetti, J. Niehren and J. Ramon. ‘Linear Programs with Conjunctive Database Queries’. In: *Logical Methods in Computer Science* Volume 20, Issue 1 (3rd Jan. 2024). DOI: 10.46298/lmcs-20(1:9)2024. URL: <https://hal.science/hal-04317553>. In press (cit. on p. 12).
- [19] P. Karmakar, M. Monet, P. Senellart and S. Bressan. ‘Expected Shapley-Like Scores of Boolean Functions: Complexity and Applications to Probabilistic Databases’. In: *Proceedings of the ACM on Management of Data* 2.2 (PODS) (12th Jan. 2024). DOI: 10.1145/3651593. URL: <https://inria.hal.science/hal-04393781> (cit. on pp. 12, 15).

### International peer-reviewed conferences

- [20] A. Amarilli, P. Bourhis, F. Capelli and M. Monet. ‘Ranked Enumeration for MSO on Trees via Knowledge Compilation’. In: *27th International Conference on Database Theory (ICDT 2024)*. International Conference on Database Theory (ICDT 2024). Vol. 290. 25. Paestum, Italy, 14th Mar. 2024, 5:1–25:18. DOI: 10.4230/LIPIcs.ICDT.2024.25. URL: <https://inria.hal.science/hal-04377344> (cit. on p. 13).

- [21] F. Capelli and O. Irwin. ‘Direct Access for Conjunctive Queries with Negations’. In: International Conference on Database Theory. Vol. 27th International Conference on Database Theory (ICDT 2024). Paestum, Italy: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024, 13:1–13:20. DOI: [10.4230/LIPICs.ICDT.2024.13](https://hal.science/hal-04504243). URL: <https://hal.science/hal-04504243> (cit. on pp. 9, 11, 15).
- [22] M. Gienieccko, F. Murlak and C. Paperman. ‘Supporting Descendants in SIMD-Accelerated JSON-Path’. In: International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2024). San Diego (California), United States: ACM, 2024, pp. 338–361. DOI: [10.4230/LIPICs](https://hal.science/hal-04398350). URL: <https://hal.science/hal-04398350> (cit. on pp. 13, 15).
- [23] S. Salvati and S. Tison. ‘Containment of Regular Path Queries Under Path Constraints’. In: 27th International Conference on Database Theory (ICDT 2024). Vol. 27th International Conference on Database Theory (ICDT 2024). Leibniz International Proceedings in Informatics (LIPIcs). Paestum, Italy: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. DOI: [10.4230/LIPICs.ICDT.2024.17](https://inria.hal.science/hal-04520222). URL: <https://inria.hal.science/hal-04520222> (cit. on p. 14).

### Conferences without proceedings

- [24] A. Amarilli, B. Groz and N. Wein. ‘Edge-Minimum Walk of Modular Length in Polynomial Time’. In: 16th Innovations in Theoretical Computer Science - ITCS 2025. New York City, United States, 2nd Dec. 2024. URL: <https://hal.science/hal-04871489> (cit. on p. 14).

### Reports & preprints

- [25] A. Amarilli, M. Arenas, Y. Choi, M. Monet, G. v. D. Broeck and B. Wang. *A Circus of Circuits: Connections Between Decision Diagrams, Circuits, and Automata*. 15th Apr. 2024. URL: <https://hal.science/hal-04871512> (cit. on p. 12).
- [26] A. Amarilli, W. Gatterbauer, N. Makhija and M. Monet. *Resilience for Regular Path Queries: Towards a Complexity Classification*. 12th Dec. 2024. URL: <https://hal.science/hal-04871464> (cit. on p. 13).
- [27] A. Amarilli, M. Monet and D. Suciu. *The Non-Cancelling Intersections Conjecture*. 29th Jan. 2024. URL: <https://inria.hal.science/hal-04603239> (cit. on p. 12).
- [28] F. Capelli, O. Irwin and S. Salvati. *A Simple Algorithm for Worst Case Optimal Join and Sampling*. 20th Sept. 2024. URL: <https://hal.science/hal-04707428> (cit. on p. 12).
- [29] B. Degardins, M. Mouton, B. Guillon, C. Paperman and C. Marchet. *Vizitig: A visual tool for colored de Bruijn graphs exploration*. 2024. DOI: [10.5281/zenodo.13929021](https://zenodo.org/record/13929021). URL: <https://hal.science/hal-04735326> (cit. on p. 14).

### Other scientific publications

- [30] A. Amarilli and F. Capelli. ‘Tractable Circuits in Database Theory’. In: *SIGMOD record* 53.2 (31st July 2024), pp. 6–20. DOI: [10.1145/3685980.3685982](https://hal.science/hal-04871509). URL: <https://hal.science/hal-04871509> (cit. on p. 11).