The Chilean Database Group

Marcelo Arenas[1], Pablo Barceló[2], Loreto Bravo[3], Claudio Gutiérrez[2],
Jorge Pérez[2], Juan Reutter[1], M. Andrea Rodríguez[3]

[1] Department of Computer Science, Pontificia Universidad Católica de Chile
{marenas,jlreutter}@ing.puc.cl
[2] Department of Computer Science, Universidad de Chile
{pbarcelo,cgutierr,jperez}@dcc.uchile.cl
[3] Department of Computer Science, Universidad de Concepción
{lbravo,andrea}@udec.cl

Abstract. During the last 15 years, the chilean researchers on databases have built a strong and cohesive group with wide international visiblity. In the present article, we briefly survey the history of the group and describe the research done by the team in five big areas: Semantic web databases, graph databases, data exchange, access control policies, and spatial databases. We also describe the international collaboration networks of the group and the participation of the group members in the organization of AMW, the most important regional event in data management. We finish the article by explaining what are the next steps for the group and what are the plans to achieve them.

Categories and Subject Descriptors: H.2 [**Database Management**]: Miscellaneous

Keywords: Chile, Computer Science, Databases

1. INTRODUCTION

The Chilean Database Group (CDG) consists of several Professors, postdocs and students interested in different aspects of databases, that interact in a more or less cohesive way. The group is split across three Computer Science departments in the country. Two of them are located in Santiago – in particular, at the University of Chile (UCH) and at the Pontifical Catholic University of Chile (PUC) – and one in Concepción, 500 km south from Santiago, at the University of Concepción (UdeC).

The Professors that lead the CDG are listed below, including their affiliations and research interests:

—Marcelo Arenas (PUC). Database theory, semantic Web.
—Pablo Barceló (UCH). Database theory, in particular, graph databases and interoperability.
—Loreto Bravo (UdeC). Data consistency, access control.
—Claudio Gutiérrez (UCH). Semantic Web, linked data, open data.
—Jorge Pérez (UCH). Database theory, semantic Web.
—Juan Reutter (PUC). Database theory.
—M. Andrea Rodríguez (UdeC). Spatial databases.

By the end of 2013 a new Professor will be joining the group: Cristián Riveros (PUC). His research interests are in the areas of database theory and query languages.

Several other faculty members of chilean institutions have worked with our group and coauthored a good amount of our articles. This includes Carlos Hurtado (Univ. Adolfo Ibañez and Microsystem S.A.), Mónica Caniupan (Univ. of Bío-Bío), Renzo Angles (Univ. of Talca), Mauro San Martín (Univ.

de la Serena) and Marcela Varas (UdeC). We are indebted to all of them. We are sure that the quality of our research has been vastly enriched by their contributions.

The group also consists of two postdocs – Carlos Buil-Aranda (PUC) and Gaelle Fontaine (UCH) – two Ph.D. students and more than 10 master students.

The CDG has obtained high visibility worldwide in the last years. This is witnessed by the international recognition of our work (with more than 6000 citations in Google Scholar), several best paper awards in the most important conferences in the area (WWW 2012, PODS 2003, PODS 2005, PODS 2011, ISWC 2006, ESWC 2005, ESWC 2007, ESWC 2011), invitations to give plenary talks in premier international conferences (PODS 2011, PODS 2013, ESWC 2007) and international courses in the most important summer schools in the area, participation in the program committees of the most renowned venues (WWW, SIGMOD, PODS, ICDT, ISWC, AAAI, STACS, GIS, etc.), and a publication record with a notorious number of articles in top conferences (WWW, VLDB, SIGMOD, PODS, ICDT, LICS, ICALP, ISWC, ESWC, IJCAI, GIS, etc.) and journals (JACM, SICOMP, TODS, VLDB Journal, J. of Web Semantics, IEEE Transactions on KDE, etc.).

1.1 A bit of History

A big part of our group shares a common academic ancestor, who is Professor Leopoldo Bertossi from Carleton University (Ottawa, Canada). Prof. Bertossi held a faculty position from 1992 until 2001 in the Department of Computer Science of PUC. Although being initially trained in the formal aspects of logic and probabilities, Bertossi quickly became interested in the foundations of databases. His early research on the topic produced many articles about consistency of data, the most influential of which is his PODS 1999 article "Consistent query answering over inconsistent databases" [Arenas et al. 1999], with more than 500 citations in Google Scholar.

Notably, one of the coauthors of Prof. Bertossi in the article "Consistent query answering over inconsistent databases" (and in many others) was a young student who is now a member of our group: Marcelo Arenas. Marcelo finished his Master studies under the supervision of L. Bertossi, and then moved in 2000 to the University of Toronto, Canada, to start his PhD studies under the supervision of Prof. Leonid Libkin. This group was joined two years later by another member of the CDG, Pablo Barceló, who had also finished his Master studies under the supervision of L. Bertossi.

Thus, Marcelo Arenas and Pablo Barceló have a strong academic link with Prof. Bertossi. This is also the case for another member of the group, Loreto Bravo, who completed his PhD studies under his supervision in Carleton University. In addition, Marcelo Arenas supervised the PhD studies of Jorge Pérez, and the Master studies of Juan Reutter and Cristián Riveros. All of them are thus academic "grandchildren" of Prof. Bertossi.

As we mentioned above, Prof. Bertossi left PUC in 2001. He moved to Carleton University, Canada, where he has been until today. Fortunately, his departure did not mean the end of database research in Chile, because two members of our group, Claudio Gutiérrez and M. Andrea Rodríguez, returned to Chile after finishing their PhDs in the early 2000s. Claudio established as faculty member of UCH after finishing a PhD in Wesleyan University, USA, and M. Andrea as faculty member of UdeC after finishing a PhD in the University of Maine, USA. Andrea had been working on spatial aspects of databases, area that she continued researching after her return to Chile [Rodríguez and Egenhofer 2003; Rodríguez et al. 2013]. On the other hand, Claudio's interests had focused on more theoretical aspects, but after his return to Chile he quickly switched area to databases, and, in particular, to the Foundations of Semantic Web data [Gutiérrez et al. 2004; Gutiérrez et al. 2005]. He was joined in this work by a former member of our group, Carlos Hurtado (currently in the industry), who had also recently returned to Chile after finishing a PhD in the University of Toronto. He had worked there under the supervision of one of the most prominent Latin American database researchers of their time, Alberto Mendelzon, who passed away in June 2005. M. Andrea Rodríguez and Claudio Gutiérrez have

worked together in the area of qualitative reasoning of networks [Rodríguez and Gutierrez 2006].

Arenas and Barceló produced several joint articles while doing their PhD in Toronto. Marcelo finished his PhD in 2005, and returned to Chile to continue his academic career as a member of PUC. Pablo Barceló returned to Chile in 2006, and since 2007 he is a faculty member of UCH. Both have continued working actively in the area of databases, and jointly in the area of data exchange [Arenas et al. 2011; Arenas et al. 2010]. Professor Arenas also established a close collaboration with UCH working with Claudio Gutiérrez in the area of Foundations of Semantic Web Databases [Pérez et al. 2009; 2010]. They were joined in this work by Jorge Pérez, by the time a PhD student of Marcelo Arenas. After finishing his PhD in 2011, Jorge became faculty member of UCH.

After finishing her PhD in 2007 and a postdoc in 2008 in the University of Edinburgh, Loreto Bravo joined the faculty of UdeC. She has collaborated since then with M. Andrea Rodríguez, particularly in topics related to data consistency over spatial databases [Bravo and Rodríguez 2012]. Juan Reutter joined the faculty of PUC during January, 2013, after finishing his PhD at the University of Edinburgh. He has a vast collaboration record with Marcelo Arenas and Jorge Pérez on topics related to data interoperability and schema mapping management [Arenas et al. 2009b; Arenas et al. 2011; Arenas et al. 2012]. He has also extensively worked with Pablo Barceló and Jorge Pérez in the areas of data exchange and graph databases [Arenas et al. 2011; Barceló et al. 2013].

1.2 Organization of the Article

In Section 2 we present the main research topics that our team has studied over the last years. This includes semantic Web data, graph databases, data exchange and schema mapping management, access control policies and its consistency, and spatial databases. In Section 3 we describe in detail about the extensive collaboration network of our group around the world. Later, in Section 4, we present details about the Alberto Mendelzon Workshop (AMW), which is an initiative of the Latin America community of researchers on data management and the Web that has received permanent support from the CDG. Finally, in Section 5, we present some concluding remarks about our past achievements, but also about how we project our group in the future.

2. MAIN RESEARCH TOPICS

The CDG has pursued different research directions in the last decade. We present here the five themes that have concentrated most of our efforts and that are most representative of the research done by the team in terms of international visibility: (1) Semantic Web databases, (2) graph databases, (3) data exchange and schema mapping management, (4) access control policies and its consistency, and (5) spatial databases.

2.1 Semantic Web Databases

The Semantic Web is a proposal to build an infrastructure of machine-readable data on the Web [Tim Berners-Lee and Lassila 2001]. In 1999, the World Wide Web Consortium (W3C) issued a recommendation of a metadata model and language to serve as the basis for such infrastructure, the *Resource Description Framework (RDF)*. Motivated by the theoretical development of the Semantic Web, our group has done research in several aspects of Semantic Web data management in the last 10 years. This includes the study of the foundations of RDF [Gutiérrez et al. 2004], the study of the semantics, complexity and expressiveness of the standard query language for RDF, SPARQL [Pérez et al. 2009], the study of several proposals to enhance the expressiveness of SPARQL [Pérez et al. 2010; Buil-Aranda et al. 2011; Arenas et al. 2012] and, more recently, the foundations of the optimization of Semantic Web queries [Letelier et al. 2012]. Several surveys and tutorials on Semantic Web Databases published by members of our group summarize some of the results described in this section [Arenas et al. 2009; Arenas and Pérez 2011; 2012; Arenas et al. 2012].

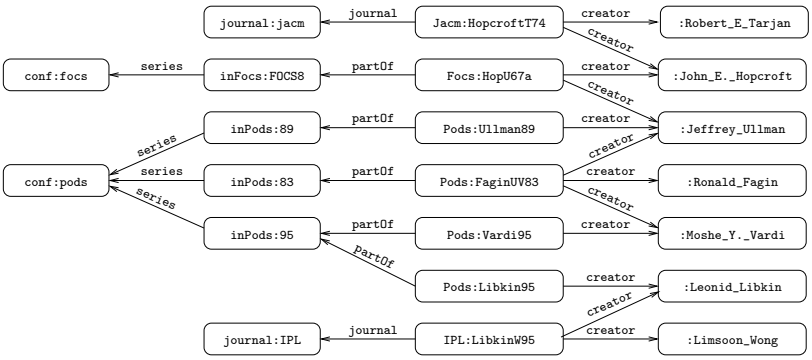


Fig. 1. An abstraction of a fragment of the RDF representation of DBLP available at `http://dblp.l3s.de/d2r/`

In the RDF model, the universe to be modeled is a set of *resources*, essentially anything that can have a *universal resource identifier*, URI. The language to describe them is a set of *properties*, technically, binary predicates. Descriptions are *statements* very much in the subject-predicate-object structure. Both subject and object can be anonymous objects, known as *blank nodes.* A subject-predicate-object piece of data is called an RDF *triple* and sets of RDF triples conform what is called an *RDF graph* in which subject and predicate of triples are the nodes and properties are labels of directed edges between nodes. Figure 1 shows an RDF graph that represents bibliographic data. In addition to the aforementioned features, the RDF specification includes a built-in vocabulary with a normative semantics, the *RDF Schema* (RDFS). This vocabulary deals with inheritance of classes and properties, as well as typing, among other features.

The RDF data model allows several representations for the same information, which raises the question about the existence of normal forms and testing of equivalence among them. On the same lines, query language features deserve a systematic and integrated study. Traditional database notions of query containment do not translate directly to the RDF setting. They need to be reformulated to take into account the fact that RDF queries process logical specifications rather than plain data. Regarding query processing, the presence of predefined semantics given by the RDFS vocabulary, and also blank nodes that act as existentially quantified variables in RDF, introduce new problems.

One of the first articles from members of our group on the topic of Semantic Web was "Foundations of Semantic Web Databases" published in PODS 2004 [Gutiérrez et al. 2004], and it rapidly became a standard citation for the subsequent works on the foundations of Semantic Web data management[1]. The group made the first step towards the systematic study of the theoretical aspects of RDF [Gutiérrez et al. 2004]. The authors presented an integrated analysis of fundamental database problems in the realm of RDF, a simple and abstract version of RDF which captured the core aspects of the language, as well as a simple query language in a streamlined form to have a basic core to focus on the central aspects of the problems mentioned in the previous paragraph. The abstract model was not intended for practical use, but designed to be simple enough to make it easy to formalize and prove results about its properties. The query language design addressed the basic features that arise in querying RDF graphs as opposed to standard databases: the presence of blank nodes, premises in queries, and the role played in this scenario by RDFS vocabulary with predefined semantics.

Jointly with the release of RDF by the W3C, the natural problem of querying RDF data was raised. Since then, several designs and implementations of RDF query languages had been proposed. In 2004, the RDF Data Access Working Group (DAWG), part of the W3C Semantic Web Activity, released a first public working draft of a query language for RDF, called SPARQL[2]. Since then, SPARQL has

[1]A full, adapted and updated version of this article was published some years later in JCSS 2011 [Gutierrez et al. 2011].
[2]The name SPARQL is a recursive acronym that stands for *SPARQL Protocol and RDF Query Language.*

been adopted as the standard for querying Semantic Web data. Four years later (January 2008), SPARQL became a W3C Recommendation. DAWG has recently released (March 2013) the second version of SPARQL called SPARQL 1.1.

RDF is a directed labeled graph data format and, thus, SPARQL is essentially a graph-matching query language. SPARQL queries are composed of three parts: (a) *pattern matching*, which includes several interesting features of pattern matching for graphs, like conjunctions, optional parts, union of patterns, nesting, filtering values of possible matchings, and the possibility of choosing the data source to be matched by a pattern; (b) *solution modifiers*, which once the output of the pattern has been computed (in the form of a table of values of variables), allow to modify these values applying classical operators like projection, distinct, order and limit; (c) the *output* of a SPARQL query can be of different types: yes/no queries, selections of values of variables which match the patterns, and construction of new RDF data.

The definition of a formal semantics for SPARQL has played a key role in the standardization process of this query language. Although taken one by one the features of SPARQL are intuitive and simple to describe and understand, it turns out that the combination of them makes SPARQL a complex language, and reaching a consensus in the W3C standardization process about a formal semantics for SPARQL, was not an easy task. In the article "Semantics and Complexity of SPARQL" [Pérez et al. 2009], members of our group presented one of the first formalizations of a semantics for a fragment of the language. Currently, the official specification of SPARQL, endorsed by the W3C, formalizes a semantics based on the work from our group.

A fundamental issue in every query language is the complexity of query evaluation and, in particular, what is the influence of each component of the language in this complexity. In another contribution, Pérez et al. [2009] studied the complexity of the evaluation of SPARQL graph patterns. The authors considered several fragments of SPARQL built incrementally, and presented complexity results for each such fragment. Among other results, they showed that the complexity of the evaluation problem for general SPARQL graph patterns is PSPACE-complete. It should be noticed that one of the delicate issues in the formalization of SPARQL is the treatment of *optional matching* [Pérez et al. 2009]. Pérez et al. [2009] proved that the high (PSPACE) complexity is obtained as a consequence of unlimited use of nested optional parts. Thus, another contribution of Pérez et al. [2009] was the identification of a large fragment of the language where the query evaluation problem can be solved more efficiently. This fragment of the language, called *well-designed patterns*, is obtained by forbidding a special form of interaction between variables appearing in optional parts.

Well-designed patterns form a natural fragment of SPARQL that is very common in practice. Furthermore, well-designed patterns have several interesting features. We show that the complexity of the query evaluation problem for well-designed patterns is considerably lower, namely coNP-complete. We also prove that the property of being well designed has important consequences for the optimization of SPARQL queries. We present several rewriting rules for well-designed patterns whose application may have a considerable impact in the cost of evaluating SPARQL queries, and prove the existence of a normal form for well-designed patterns based on the application of these rewriting rules.

Recently, members of our group have contributed to the formalization of SPARQL 1.1 which includes two main new features: *Federation* and *Navigation*. Since the release of the first version of SPARQL, the Web has witnessed a constant growth in the amount of RDF data publicly available on-line. This has led the W3C to standardize a set of protocols – plus some language constructs – that allow to query RDF repositories via SPARQL. All these constructs are part of the *federation extensions* of SPARQL [Prud'hommeaux and Buil-Aranda 2013], included in the new version of the standard. Members of our group discovered some issues in the preliminary version of the SPARQL 1.1 standard, specifically, on the impossibility of answering some unbounded federated queries [Buil-Aranda et al. 2011; Buil-Aranda et al. 2013]. Somewhat orthogonally, we have the issue of navigating data. It has been largely recognized that navigational capabilities are of fundamental importance for data

models with explicit tree or graph structure, like XML and RDF. Nevertheless, the first release of SPARQL included very limited navigational capabilities. This is one of the motivations of the W3C to include the *property-path* feature in SPARQL 1.1. Property paths are essentially regular expressions, that retrieve pairs of nodes from an RDF graph that are connected by paths conforming to those expressions, and provide a very powerful formalism to query RDF data. Members of our group raised the problem of the high computational complexity of the evaluation problem for queries including navigation functionalities as defined in the preliminary versions of the SPARQL 1.1 specification, providing a proposal on how to deal with these problems [Arenas et al. 2012] . The results [Buil-Aranda et al. 2011; Buil-Aranda et al. 2013; Arenas et al. 2012] were considered to develop the final specification of SPARQL 1.1 released in March 2013 by the W3C.

2.2 Graph Databases

Graph databases are crucial for many applications in which the topology of the data is as important as the data itself. While early interest in graph databases could be explained by their applications in hypertext systems, or their connections with semistructured data and object databases, new application domains have taken the field by storm in the last decade, including the Semantic Web, social networks analysis, biological networks, data provenance, and several others. Graph databases have been surveyed in full detail by a member of Gutiérrez [Angles and Gutiérrez 2008].

Our group has done research in several aspects of graph databases in the last three years. This includes the design and study of expressive languages for path queries, a deep analysis of the notion of incompleteness for graph databases, and the study of restricted languages that allow tractable evaluation in combined complexity. We delve into these issues below, but before we provide a quick introduction to graph databases and their query languages. We expect this will help the reader obtaining a better intuitive understanding of our work. Several aspects of navigational query languages for graph databases have been surveyed by Barceló [Barceló 2013].

In their simplest form, graph databases are finite, directed, edge-labeled graphs. Let Σ be a finite alphabet. A graph database $\mathcal{G}$ over Σ is a pair (V, E), where V is a finite set of nodes and $E \subseteq V \times \Sigma \times V$. Thus, each edge in $\mathcal{G}$ is a triple $(v, a, v') \in V \times \Sigma \times V$, whose interpretation is an a-labeled edge from v to v' in $\mathcal{G}$. A *path* ρ in $\mathcal{G}$ is a sequence $v_0 a_0 v_1 \cdots v_{k-1} a_k v_k$, such that $(v_{i-1}, a_{i-1}, v_i) \in E$, for each i with $1 \leq i \leq k$. The *label* of ρ, denoted $\lambda(\rho)$, is the string $a_0 \cdots a_{k-1} \in \Sigma^*$.

Many query languages for graph databases are *navigational*, which means that they allow recursively traversing the edges of the graph database (in both directions) looking for a path satisfying certain condition. This immediately precludes the application of relational database technology for answering queries over graph databases, as most relational languages, including SQL, allow limited recursion.

One of the simplest examples of a navigational language for graph databases is the class of *regular path queries with inverse*, or 2RPQs, that are regular expressions over the aphabet $\Sigma^{\pm}$ that extends Σ with the *inverse* a^- of each symbol $a \in \Sigma$. In order to define the evaluation of a 2RPQ over a graph database $\mathcal{G} = (V, E)$ over Σ, we need the notion of the *completion* $\mathcal{G}^{\pm}$ of $\mathcal{G}$, which is the graph database over $\Sigma^{\pm}$ that is obtained from $\mathcal{G}$ by adding the *inverse* edge (u, a^-, v), for each edge $(v, a, u) \in E$. The interpretation $r(\mathcal{G})$ of the 2RPQ r over the graph database $\mathcal{G} = (V, E)$ consists of all pairs (v, v') of nodes in V such that there is a path ρ from v to v' in $\mathcal{G}$ for which it is the case that $\lambda(\rho)$ belongs to the regular language defined by r (this is well-defined since r is a regular expression over $\Sigma^{\pm}$).

Example: Let $\mathcal{G}$ be the graph database over $\Sigma = \{\texttt{creator}, \texttt{partOf}, \texttt{series}\}$ in Figure 1. This graph contains an abstraction of a fragment of the *RDF Linked Data* representation of DBLP (and it is based on an example by Arenas and Pérez in a PODS tutorial [Arenas and Pérez 2011]). The 2RPQ $r = \texttt{creator}^- \cdot \texttt{partOf} \cdot \texttt{series}$ matches all pairs (x, y) such that x is an author that published a paper in conference y. For example, the pairs $(:\texttt{Jeffrey\_D.\_Ullman}, \texttt{conf} : \texttt{focs})$ and $(:\texttt{Ronald\_Fagin}, \texttt{conf} : \texttt{pods})$ are in $r(\mathcal{G})$. □

Extending 2RPQs with joins and projection yields the class of *conjunctive* 2RPQs, or C2RPQs. This class permits to check the existence of interesting patterns over the data, in the same way than the class of *conjunctive queries* over relational databases does. For instance, consider the C2RPQ given by the rule $Ans(x, y) \leftarrow (x, \texttt{creator}^-, \texttt{u}) \wedge (\texttt{u}, \texttt{partOf}, \texttt{v}) \wedge (\texttt{v}, \texttt{series}, \texttt{w}) \wedge (\texttt{u}, \texttt{creator}, \texttt{y})$, where $Ans(x, y)$ is a distinguished binary symbol that defines the output of this query to contain variables x and y only (that is, variables u, v, w are *existentially quantified*). Over the graph database $\mathcal{G}$ shown in Figure 1 this C2RPQ defines the set of pairs (x, y) of authors that have a joint conference paper, e.g., the pairs (: `Jeffrey_D._Ullman`, : `Ronald_Fagin`) and (: `Ronald_Fagin`, : `Moshe_Y._Vardi`) belong to the output of the query. This query cannot be expressed as a 2RPQ over $\mathcal{G}$, and, thus, that C2RPQs increase the expressiveness of the language of 2RPQs. Evaluation of C2RPQs is tractable in data complexity (that is, assuming the query to be fixed), but NP-complete in combined complexity (that is, when both data and query are part of the input).

In our work we have noticed that the class of C2RPQs falls short of expressive power for several modern applications of graph databases. In many of these applications, such as the semantic Web, biological netwoks, or provenance, a minimal requirement for sufficiently expressive queries are: (a) the ability to define complex semantic relationships among paths, and (b) the ability to include paths in the output of a query. None of these functionalities is provided by the C2RPQs.

In order to overcome this lack of expressiveness, Barceló, together with international collaborators, introduced a class of queries, called *extended* C2RPQs (EC2RPQs), that increase the expressiveness of C2RPQs by allowing to output and compare paths [Barceló et al. 2012]. Paths are compared by its conformance to a *regular relation*, which is a class that naturally extends the class of regular languages to relations of arbitrary arity over words. Examples of binary regular relations are the prefix relation, i.e. the set of pairs (w_1, w_2) of words such that w_1 is a prefix of w_2, and the relation that contains all pairs of words that are at *edit distance* at most k, for $k \geq 0$. EC2RPQs preserve the data complexity of query evaluation of C2RPQs, that is, they can be evaluated efficiently assuming the query to be fixed [Barceló et al. 2012]. Several important properties of the class of EC2RPQs, and of some of its extensions, are also studied in the article. Since the output of an EC2RPQ may contain paths, it is potentially infinite (e.g. in the case when the graph database $\mathcal{G}$ contains a cycle and the query matches all paths in $\mathcal{G}$). Interestingly enough, it is shown [Barceló et al. 2012] that it is possible in such cases to compute a compact (in particular, finite) representation of the output, in the form of an automaton that accepts precisely the paths that belong to it.

Several relations on words of practical interest are not regular, e.g., the subsequence or subword relations, that find applications in the semantic Web or in biological networks. Because of this, Barceló, together with international collaborators, studied a class of EC2RPQs that allows to compare paths in an extended class that contains these relations [Barceló et al. 2012]. This is the class of *rational* relations, as defined by asynchronous n-head automata. It is easy to prove that the query evaluation problem for this extended language is undecidable. This result is strengthened in our article, as it is shown that it is undecidable to evaluate these queries even if the only rational relation allowed is the suffix relation (which is of practical importance). On the other hand, only allowing the subsequence relation yields decidability but at a prohibitively high complexity. These are negative results, that rule out the possibility of using these languages in practical applications. On the positive side, the article identifies several syntactic restrictions of the language that are motivated by practical applications and allow tractable evaluation in data complexity.

Incompleteness is an important problem for modern applications that integrate and exchange data. Barceló and Reutter studied, together with Prof. Libkin from the Univ. of Ednburgh, the notion of incompleteness in graph databases [Barceló et al. 2010]. Intuitively, an incomplete graph database is a graph database in which some nodes are replaced by node variables and some edges are labeled by regular expressions. Such incomplete graph database *represents* each graph database that can be obtained by replacing node variables by nodes and each edge labeled with regular expression r by a

path labeled with a word in r. Query answering over incomplete databases is commonly understood in terms of *certain* answers, which are those that hold in each graph database that is represented by the incomplete one. Several important problems of incomplete graph databases are studied in our article, including query answering for many of the languages we have seen before.

Evaluation of C2RPQs is of order $O(|\mathcal{G}|^{|Q|})$, for $\mathcal{G}$ a graph database and Q a C2RPQ. This is infeasible for modern applications of graph databases that store massive amounts of data (i.e., $|\mathcal{G}|$ is very big), even if Q is small. Barceló, Pérez and Reutter identified and studied several query languages for graph databases that are useful in practice and allow tractable evaluation in combined complexity [Barceló et al. 2012]. In particular, it is shown that a syntactic restriction of the class of C2RPQs, namely, the acyclic C2RPQs, can be evaluated in linear time $O(|\mathcal{G}| \cdot |Q|)$. Intuitively, a C2RPQ is acyclic if its *underlying undirected graph* is acyclic. An example of an acyclic C2RPQ is the query $Ans(x, y) \leftarrow (x, \texttt{creator}^-, \texttt{u}) \wedge (\texttt{u}, \texttt{partOf}, \texttt{v}) \wedge (\texttt{v}, \texttt{series}, \texttt{w}) \wedge (\texttt{u}, \texttt{creator}, \texttt{y})$ presented earlier. On the other hand, the C2RPQ $Ans(x, y, z) \leftarrow (x, a, y), (y, b, z), (z, c, x)$ over alphabet $\Sigma = \{a, b, c\}$ is not acyclic, since its underlying undirected graph consists of a single cycle on nodes $\{x, y, z\}$.

We have also investigated an expressive extension of the class of 2RPQs with an existential nesting operator (*á la* XPath) that allows linear time evaluation. This is known as the class of *nested regular expressions* (NREs). Acyclic C2RPQs and NREs are compared in terms of their expressive power [Barceló et al. 2012]. It is shown that they are incomparable, but NREs properly extend the class of acyclic C2RPQs assuming a mild syntactic restriction on the latter.

Acyclicity is a *syntactic* property of C2RPQs that yields tractability of query evaluation. The class of *unions* of acyclic C2RPQs retains this good behavior. It is natural then to ask what happens with the class of unions of C2RPQs that have this property at a *semantic* level, that is, the class of unions of C2RPQs that are equivalent to a union of acyclic C2RPQs. We call this the class of *semantically acyclic* unions of C2RPQs. Barceló, together with PhD student Miguel Romero and Prof. M. Vardi from Rice University, have recently shown that it is decidable to check whether a union of C2RPQs belongs to this class [Barceló et al. 2013]. This implies that the class of semantically acyclic unions of C2RPQs has better evaluation properties than arbitrary unions of C2RPQs. Our results yield a strong theory of *approximations* for unions of C2RPQs in the class of unions of acyclic C2RPQs. Intuitively, the approximation of a union of C2RPQs Q is the union of acyclic C2RPQs Q' that differs as "little as possible" from Q. In the case when it is infeasible to run Q on a graph database, one may prefer to run its approximation Q' that can be evaluated efficiently. Our results extend recent results of Barceló, Libkin and Romero on the theory of approximations for conjuntive queries over relational databases [Barceló et al. 2012].

2.3 Data exchange and schema mapping management

A schema mapping is a specification that describes how data from a source schema is to be mapped to a target schema. Schema mappings are of fundamental importance in data management today. In particular, they have proved to be the essential building block for several data-interoperability tasks such as data exchange, data integration and peer data management.

The research of our group has focused both on the fundamentals of schema mappings, as well as in the problem of data exchange, one of the main applications of schema mappings. A comprehensive book on this subject has been compiled by two members of our group (together with two international collaborators) [Arenas et al. 2010], and we have also published two surveys covering the recent results in these topics (see [Arenas et al. 2009a] for schema mapping management and [Barceló 2009] for data exchange). We now briefly survey our main contributions to these areas, but before we need to introduce some terminology.

Given two relational schemas $\mathbf{R}_1$ and $\mathbf{R}_2$, with no relation symbols in common, a *schema mapping* $\mathcal{M}$ (or just *mapping*, from now on) from $\mathbf{R}_1$ to $\mathbf{R}_2$ is a set of pairs (I, J), where I is an instance of

$\mathbf{R}_1$, and J is an instance of $\mathbf{R}_2$. Further, we say that J is a *solution for I under $\mathcal{M}$* if $(I, J) \in \mathcal{M}$. Usually one assumes that mappings are *specified* by a special class of sentences in first-order logic, that we here call *dependencies*. Formally, a *dependency* from $\mathbf{R}_1$ to $\mathbf{R}_2$ is a sentence of the form

$$\forall \bar{x}\,(\varphi(\bar{x}) \rightarrow \psi(\bar{x})),$$

where $\varphi(\bar{x})$ and $\psi(\bar{x})$ are first order queries over $\mathbf{R}_1$ and $\mathbf{R}_2$, respectively. If both $\varphi(\bar{x})$ and $\psi(\bar{x})$ are conjunctive queries, then our dependency is a *tuple-generating dependency* (tgd), a special class of dependencies that has taken a prominent role in data exchange applications. We then say that a mapping $\mathcal{M}$ is *specified* by a set Σ of dependencies, if for every pair (I, J) of instances of $\mathbf{R}_1$ and $\mathbf{R}_2$, respectively, we have that $(I, J) \in \mathcal{M}$ if and only if (I, J) satisfies Σ.

Example: Consider relational schemas $\mathbf{R}_1$ and $\mathbf{R}_2$ containing relations $\texttt{Emp}(\texttt{name}, \texttt{lives\_in}, \texttt{works\_in})$ and $\texttt{Shuttle}(\texttt{name}, \texttt{dest})$, respectively. Relation `Emp` in schema $\mathbf{R}_1$ is used to store employees names and the places where they live and work. Relation `Shuttle` in schema $\mathbf{R}_2$ is intended to store names of employees that must take the shuttle bus to reach their workplaces (destination). A possible way of relating schemas $\mathbf{R}_1$ and $\mathbf{R}_2$ is by using the following dependency:

$$\forall x \forall z \big(\exists y(\texttt{Emp}(x, y, z) \wedge y \neq z) \rightarrow \texttt{Shuttle}(x, z) \big). \tag{1}$$

The above formula states that if relation `Emp` stores an employee that lives in a place different from which she/he works in, then the employee and the place where she/he works in should be stored in relation `Shuttle`. In this case, the mapping $\mathcal{M}$ specified with dependency (1) contains all pairs of instances that are consistent with the semantic relationship we have just described. □

Many information-system problems involve not only the design and integration of artifacts such as schema mappings, but also their subsequent manipulation. This led to the proposal of a framework for managing schema mappings, called *model management*, in which high-level algebraic operators such as composition, merging and inversion are used to manipulate schema mappings.

Our group has conducted extensive research on this area, particularly with respect to the inverse operator for schema mappings. Intuitively, given a mapping $\mathcal{M}$ from a schema $\mathbf{R}_1$ to a schema $\mathbf{R}_2$, *an inverse* of $\mathcal{M}_1$ is a new mapping that describes the *reverse* relationship from $\mathbf{R}_2$ to $\mathbf{R}_1$, and is semantically consistent with $\mathcal{M}$. Computing this inverse has several applications in practical scenarios. For example, in a data exchange context, if a mapping $\mathcal{M}$ is used to exchange data from a source to a target schema, an inverse of $\mathcal{M}$ can be used to exchange the data back to the source, thus *reversing* the application of $\mathcal{M}$.

The process of inverting schema mappings turned out to be a nontrivial task, and even the fundamental problem defining a meaningful and practical semantics for the inverse operator has been the subject of several research projects. Our group has made several notable contributions in this direction. Arenas, Pérez and Riveros proposed the notions of recovery and maximum recovery [Arenas et al. 2009]: a general framework for inverting schema mappings that captured all previous attempts to define inverse operators. Our group has then extended this framework in favor for other weaker notions of inverses that may be more suitable for some practical applications [Arenas et al. 2009b; 2012].

Another useful operator is composition, which can be described as follows. Given mappings $\mathcal{M}_{12}$ from $\mathbf{R}_1$ to $\mathbf{R}_2$, and $\mathcal{M}_{23}$ from $\mathbf{R}_2$ to a schema $\mathbf{R}_3$, *the composition* of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ is a new mapping that describes the relationship between schemas $\mathbf{R}_1$ and $\mathbf{R}_3$. This new mapping must be *semantically equivalent* with the relationships previously established by $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$, i.e., equivalent to the successive applications of these mappings. Arenas worked with international collaborators to study the composition of mappings that are specified with more complex dependencies [Arenas et al. 2011], and again with Pérez, Reutter and Riveros to study different notions of composition of mappings [Arenas et al. 2013]. Our group has also compiled a survey on the main results about composition and inversion of schema mappings [Arenas et al. 2009a].

There are other, more complex operators in model management that could not be studied without first developing formal tools to compare schema mappings, in terms of their ability to transfer data and avoid storing redundant information. Our group has also addressed this issue, and Arenas, Pérez, Reutter and Riveros developed an order to compare the amount of information transferred by schema mappings [Arenas et al. 2010], together with several other foundational tools for comparing mappings. Using this machinery, they formalized several complex mapping operators such as *extract* and *merge*, as well as the problem of *schema evolution*. This order has also been used by Pérez and international collaborators to define two new operators for schema mappings: union and intersection [Pérez et al. 2012].

To problem of *data exchange* is one of the main applications of schema mappings. It can be defined as follows. One is given a source schema and a target schema, a schema mapping $\mathcal{M}$ that specifies the relationship between the source and the target, and an instance I of the source schema. The basic problem then that one wants to address is how to materialize an instance of the target schema that reflects the source data as accurately as possible [Arenas et al. 2010]. In data exchange terms, the problem is how to materialize the *best solution* for I under $\mathcal{M}$.

In a traditional data exchange setting, source instances are restricted to be *complete*: every fact in them is either true or false. However natural this setting may be, this restriction is gradually becoming an impediment to a wide range of applications that need to exchange objects that admit several interpretations. Two of these applications are exchanging *incomplete information* and exchanging *knowledge bases*.

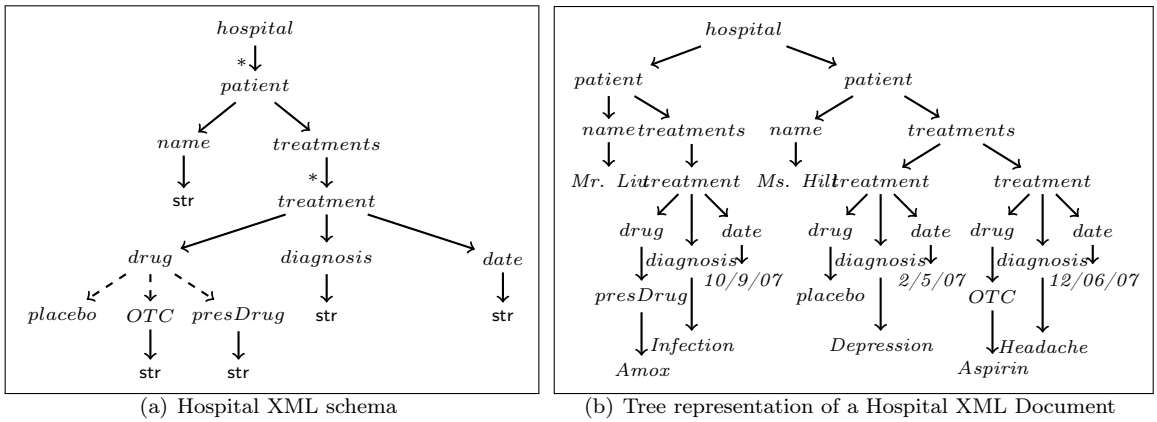
Our group is now studying how to extend this traditional setting to cope with these applications. Arenas, Pérez and Reutter studied the general problem of exchanging information given by *representation systems* [Arenas et al. 2011], essentially finite descriptions of (possibly infinite) sets of complete instances. They proposed a general framework, and then showed that it could cope with both the exchange of incomplete information and the exchange of knowledge bases. Arenas continued to work on the problem of exchanging knowledge bases [Arenas et al. 2012]. The results of our group have also been presented in tutorials in both the RR conference and the Description Logics workshop [Arenas 2011a; 2011b].

At the same time, we are also conducting research on the traditional data exchange setting. This includes, for example, our proposal to use a special type of DATALOG programs as a query language for data exchange [Arenas et al. 2011; 2010]. We have also started to combine our knowledge of graph databases with the topics of schema mappings and data exchange, specifically working on the definition of suitable mapping languages for graph databases [Barceló et al. 2013].

2.4 Access Control Policies and its Consistency

When a database contains sensitive information, access control techniques are needed to determine who can read or modify the data contained in it. This problem has been widely studied in relational databases, but is relatively recent in new data models such as XML and RDF, where there are new challenges that are not present in the relational case. For every data model, one wants to define access control policies that allow fine grained control while being able to efficiently enforce the permissions and ensure that there are no security flaws.

XML access control has received much attention as the amount of sensitive XML data exchanged between applications is increasing. Access control techniques for XML data have been considered extensively for *read-only queries*. However, the problem of controlling *write access* has not received much attention. Bravo and other researchers from the University of Edinburgh, realized that an important problem in this context is the presence of certain type of vulnerabilities, here called *inconsistencies*, that allow *one explicitly forbidden update operation to be simulated by a sequence of allowed ones*. In general, an XML write-access control policy (ACP) specifies the update actions a user can perform



(a) Hospital XML schema (b) Tree representation of a Hospital XML Document

Fig. 2. Hospital Example

based on the *syntax* of the update and *not* its actual *behavior*. Thus, it is possible that a single update request which is explicitly forbidden by the policy can nevertheless be simulated by a sequence of more than one allowed update requests.

Consider for example the XML DTD in Fig. 2(a) that describes patient data. A *patient* has a *name* and is associated with zero or more treatments. A *treatment* consists of a *drug* that was prescribed to the patient and that can be one of *placebo*, *presDrug* (prescription) and *OTC* (over-the-counter) drug, a *diagnosis* and the *date* of a patient's visit. The XML document shown in Fig. 2(b) is an instance of the hospital schema shown in Fig. 2(a). The document can be updated and queried by different users, e.g., doctors, nurses, administrators. A user is allowed to perform certain updates or access only part of the data. For example, a nurse is allowed to *insert* and *delete* patients, but she *cannot modify* a patient's diagnosis or change a prescription drug to an over-the-counter drug. It is easy to see that the diagnosis of a patient can be changed by deleting a patient record and then inserting it back again with a modified value of diagnosis. Thus, a forbidden update request can be achieved by a sequence of allowed ones. Thus, this access control policy is *inconsistent*.

It is important to be able to detect inconsistencies and suggest possible ways of repairing policies in order to ensure their consistency. Bravo and her collaborators made an initial step towards addressing this problem [Bravo et al. 2007; 2008]. They provided a formal definition of consistency and defined access control policies for documents valid against schemas of the type *structured DTDs*. Update permissions over XML documents are defined by *access types* that group updates by the operation they perform: either insert, delete, replace or replaceVal. For example $(hospital, \mathsf{delete}(patient))$ and $(hospital, \mathsf{insert}(patient))$ control the permissions for updates that respectively insert or delete a patient below an element of type hospital. An access type $(drug, \mathsf{replace}(presDrug, OTC))$ controls the replacement of an *OTC* drug by a *presDrug* below element *drug*. If we want to control the updates that replace the text within element *diagnosis*, the access type $(diagnosis, \mathsf{replaceVal})$ can be used. A *total policy* is then defined as a set of allowed and forbidden update access types such that the union of both corresponds to all possible permissions that can be defined over the given schema. For our nurse example, both $(hospital, \mathsf{delete}(patient))$ and $(hospital, \mathsf{insert}(patient))$ would be in the allowed types and $(diagnosis, \mathsf{replaceVal})$ in the forbidden ones.

For these access control policies checking consistency was shown to be in PTIME, and the repair problem based on deleting a minimal set of privileges to restore consistency was shown to be NP-complete. Bravo et al. showed that these results do not change if we extend the definition to policies over *chain extended DTDs* [Bravo et al. 2012], which account for over 90% of the schemas used in practice. Since the repair problem is not tractable, approximate algorithms are provided and experimentally evaluated by comparing them with an exact approach based on answer-set programming. It is shown that the heuristics yield reasonable results in practice.

Bravo and her co-authors also studied *partial policies* that are more convenient to write since many privileges may be left unspecified. They showed that consistent partial policies can be extended to unique least-privilege consistent total policies. The algorithms provided to repair total policies are shown to be useful also for partial ones [Bravo et al. 2012].

Bravo and her master's student Ricardo Segovia, simplified the definition of policies by only assigning insert/delete permissions and computing the replace permissions from them [Bravo and Segovia 2012]. For example, if $(drug, \mathsf{delete}(pres\ Drug))$ and $(drug, \mathsf{insert}(OTC))$ are allowed, then also the access type $(drug, \mathsf{replace}(presSrug, OTC))$ is allowed. For these simplified policies the repair problem becomes tractable with a very small loss in the expressivity of the access control policy.

2.5 Spatial Databases

Spatial database systems are important components of diverse applications such as Geographic Information Systems (GIS), Computer-Aided Design (CAD), multimedia information systems, data warehousing, mobile computing, location-based services, and NASA's Earth Observing System (EOS). Current models of spatial database systems are typically seen as extensions of the relational data model (object-relational models), with the definition of abstract data types to specify spatial attributes.

In the last three years, Rodríguez's work has concentrated on consistency of spatial databases, which offer new challenges due, particularly, to the complex nature of spatial attributes, the derivation of implicit relations between spatial attributes, and the combination of spatial and non-spatial attributes (called thematic attributes). Formally [Bravo and Rodríguez 2012; Rodríguez et al. 2013], a spatio-relational database schema is a tuple $\Sigma = (\mathcal{U}, \mathcal{A}, \mathcal{S}, \mathcal{R}, \mathcal{T})$, such that: (a) $\mathcal{U}$ is the possibly infinite database domain that includes $\mathbb{R}$; (b) $\mathcal{A} = \{A_1, \ldots, A_n\}$, where each A_i is a thematic attribute which takes values in $\mathcal{U}$; (c) $\mathcal{S} = \{S_1, \ldots, S_m\}$ where each S_i is a spatial attribute[3]; (d) $\mathcal{R}$ is a finite set of predicates, each of them with a finite and ordered set of attributes belonging to $\mathcal{A}$ or $\mathcal{S}$. When predicate R contains only one spatial attribute, we refer to it as a single-geometry relational predicate; (e) $\mathcal{T}$ is a set of binary topological predicates.

Bravo and Rodríguez presented a formalization of integrity constraints that combines thematic with spatial attributes and that extends classical notions of functional and inclusion dependencies by imposing topological relations between spatial attributes of spatial databases [Bravo and Rodríguez 2012]. A topological dependency constraint (TD) for single-geometry relational predicates is of the form $\forall \bar{x}_1 \bar{x}_2 g_1 g_2 (P(\bar{x}_1, g_1) \wedge R(\bar{x}_2, g_2) \wedge y \neq z \rightarrow T(g_1, g_2))$, where g_1, g_2 are variables of spatial attributes, y, z are variables and $\bar{x_1}, \bar{x_2}$ are sequences of variables denoting thematic attributes, and T is a topological relation (e.g., touch, overlap and disjoint) between geometries. Bravo and Rodríguez analyzed the database satisfiability problem on this and other spatial semantic constraints (i.e., referential topological constraints and check constraints) and found that it is not tractable in general. However, they showed algorithms for subsets of functional dependencies, topological dependencies and check constraints, where the later one imposes constraints on thematic attributes in function of spatial attributes.

In a related work [del Mondo et al. 2013], Bravo, Rodríguez, and their collaborators formalized a graph-based spatio-temporal model to represent evolving regions in terms of spatial and filiation (i.e., continuation and derivation) relationships. Figure 3 illustrates the components of this model, where R_1 and R_2 denotes regions, $\mathsf{EQ}, \mathsf{EC}, \mathsf{PO}$ refer to topological relations equal, touch, and partial overlap, respectively, and ρ_c, ρ_d refer to continuation and derivation relations, respectively. Based on this model, they also extended functional dependency constraints to topological-filiation dependency constraints (TFD), which impose topological relations between geometries at different time instants and holding particular filiation relationships. Similar to TD, a TFD is of the form $\forall \bar{u_1} \bar{u_2} t_1 t_2 \bar{x}_1 \bar{x}_2 g_1 g_2 (P(\bar{u}_1, t_1, \bar{x}_1, g_1) \wedge R(\bar{u_2}, t_2, \bar{x}_2, g_2) \wedge F(\bar{u_1}, t_1, \bar{u_2}, t_2) \rightarrow T(g_1, g_2))$, where $\bar{u_1}, t_1$ and

[3]Our worof k concentrates until now on spatial attributes in a 2D space

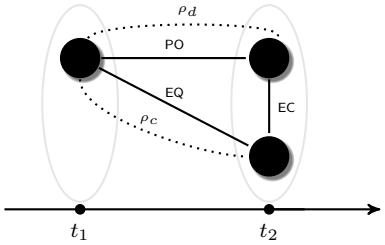


Fig. 3. A graph-based spatio-temporal model for evolving regions

$\bar{u_2}, t_2$ are keys of spatio-temporal relational predicates P and R, and predicate F defines a filiation relationship between tuples with keys $\langle \bar{u_1}, t_1 \rangle$ and $\langle \bar{u_2}, t_2 \rangle$. Checking these constraints can be done in polynomial time and different algorithms were provided [del Mondo et al. 2013].

Focusing on topological integrity constraints expressed as denial constraints (DSICs), Rodríguez and her collaborators [Rodríguez et al. 2013] proposed a repair semantics. DSICs are of the form $\forall \bar{g} \bar{x}\ \neg(\bigwedge_{i=1}^{m} R_i(\bar{x}_i, g_i) \wedge \bigwedge_i \mathsf{NonEmpty}(g_i) \wedge \varphi \wedge \bigwedge_{j=1}^{n} T_j(v_j, w_j))$, where NonEmpty is a predicate that is true when g_i is the non-empty geometry, $v_j, w_j \in \bar{g}$, φ is an optional formula that is a conjunction of built-in atoms over thematic attributes, and $T_j, \ldots, T_n$ are predicates in $\mathcal{T}$. A database D violates a DSIC with two spatial relational predicates R_1 and R_2 when there are data values $\bar{a}_1, \bar{a}_2, s_1, s_2$ in the database, with s_1, s_2 non-empty geometries, such that $(R_1(\bar{a}_1 s_1) \wedge R_2(\bar{a}_2, s_2) \wedge \varphi \wedge T(s_1, s_2))$ is true. When this is the case, it is possible to restore consistency of D by modifying s_1 or s_2, to make $T(s_1, s_2)$ false. This is done by updates that shrink geometries of objects, even at the point of deleting geometries for some exceptional cases, as for topological relation *disjoint*.

The repair semantics is used as an auxiliary concept for handling inconsistency tolerance and computing consistent answers (CQA) to spatial range and join queries. Complexity analysis showed that CQA is intractable. In particular, for a set Ψ of DSICs, deciding if an instance D' is a minimal repair of an input database instance D is CO-NP-*complete* in data (complexity). With the purpose of avoiding to compute and query all repairs, they identified cases of DSICs (IDSICs) and conjunctive (basic range and join) queries where the consistent answers can be obtained by posing a standard query to a single view of the original instance. This view is equivalent to the intersection of all possible minimal repairs, what we called the *core* of a database instance, which for IDSICs can be computed in polynomial time without determining each repair.

Recently, Rodríguez and her collaborators have studied inconsistency measures of spatial databases with respect to topological dependency constraints [Brisaboa et al. 2013]. These measures compare the topological relations between geometries with respect to their expected relations as expressed by topological dependency constraints. These measures exploit the idea that consistency for spatial information in not a binary decision: spatial objects can be partially inconsistency. They are based on intrinsic properties of the objects' geometry (length, area, etc.) and also on properties of the relationship between them (distance, overlapping area, etc.). We expect to use these measures to compare spatial databases and define strategies for data cleaning.

In a broader context, Bravo and Rodríguez presented a model, query language, and integrity constraints for multi-granular databases [Rodríguez and Bravo 2012]. This is done from a general perspective, where data is not necessarily stored at the finest level of granularity upon which aggregation functions derive data at other coarser levels. In addition, categories are not necessarily made explicit, but they can be created on the fly depending on the domain, and granules relates not only in a hierarchy, which is the classical approach in data warehousing. In this model, a granular functional dependency extends classic functional dependency with the concept of multi-granular attributes. It enforces that tuples with the same values in a subset of attributes, at specific granularities, should have the same values in another set of attributes, at given granularities. This can be further extended

to filter the tuples over which the dependencies are checked in the same way as conditional functional dependencies extend classic functional dependencies to add conditions.

3. INTERNATIONAL COLLABORATION NETWORKS

The researchers in the group have a dense network of collaborators in several universities and research centers around the world, which is witnessed by the large number of articles we have co-authored with international partners. Among others, our collaboration network includes: IBM Almaden (Ron Fagin), Microsoft Research (Phil Bernstein), Oxford Univ. (Prof. Georg Gottlob), Univ. of Edinburgh (Profs. Leonid Libkin, Wenfei Fan and James Cheney), Rice Univ. (Prof. Moshe Vardi), Birbeck College Univ. of London (Profs. Peter Wood and Andrea Calì), TU Wien (Prof. Reinhard Pichler), Univ. of Bolzano (Prof. Diego Calvanese), Univ. of Austin (Prof. Daniel Miranker), FORTH (Prof. Irini Fundulaki), DERI (Aidan Hogan), Siemens - Austria (Axel Polleres), Univ. Politécnica Madrid (Prof. Oscar Corcho), IBM Watson (Anastasios Kementsietsidis), Universidade da Coruña (Prof. Nieves Brisaboa) and Univ. of Warsaw (Prof. Filip Murlak).

4. ALBERTO MENDELZON WORKSHOP (AMW)

AMW – the Alberto Mendelzon International Workshop on Foundations of Data Management – is an initiative of the Latin American community of researchers in data management, to which our friend, colleague and mentor Alberto so greatly contributed. In its 7th edition, AMW has been a periodical Latin America-based venue for high level research in the fundamental aspects of the area. This is a way to honor the memory of Alberto, and to boost and solidify the research in the region. The event encourages the participation of Latin American graduate students and presents some activities specially designed for them, such as grad schools, tutorials and panels. The previous editions of AMW have been held in Laguna San Rafael, Chile, November 2006; Punta del Este, Uruguay, November 2007; Arequipa, Peru, May 2009; Buenos Aires, Argentina, May 2010; Santiago, Chile, May 2011; Ouro Preto, Brazil, June 2012; and Puebla, Mexico, May 2013.

The CDG and other chilean researchers have permanently supported the event and helped in its organization. One member of our group, P. Barceló, and a former member of it, L. Bertossi, belong to the steering committee of AMW. The first edition of the worskhop was organized by two chilean reseachers: Ricardo Baeza-Yates (UCH, Yahoo!) and Leopoldo Bertossi. The chairs of the 3rd edition of AMW held in Arequipa, Peru, were Marcelo Arenas and Leopoldo Bertossi. Pablo Barceló was co-chair in 2011 and Loreto Bravo in 2013. Pablo Barceló acted as general chair of the workshop in 2009, Jorge Pérez in 2011 and Leopoldo Bertossi in 2013.

5. LOOKING BACK, LOOKING FORWARD

The individual efforts of the members of our team have positioned us as a strong data management group with wide international recognition. This is witnessed, among others, by the number of our citations in Google Scholar (more than 6000) and a record of publications with a notorious number of articles in the main data management conferences (PODS, SIGMOD, VLDB, ICDT, ISWC, etc) and journals (JACM, TODS, VLDB Journal, Journal of Web Semantics, etc).

In order to consolidate the individual efforts of our members, and build a more cohesive team that tackles the big problems that will dominate the data management discussion in the next years, we require to build an even stronger group, with more professors, and, in particular, more graduate students and postdocs. We are in the process of applying to different national funding schemes that would allow us to achieve this goal.

Another goal of the group is to attract more students from the region (South America) to come pursue a graduate degree with us. This can only be done by augmenting our visibility among re-

gional prospective graduate students, and we plan to achieve this by visiting universities abroad and organizing yearly schools for them.

REFERENCES

Angles, R. and Gutiérrez, C. Survey of Graph Database Models. *ACM Computing Surveys* 40 (1): 1:1 –1:39, 2008.

Arenas, M. Exchanging More than Complete Data. In *Proceedings of the International Conference on Web Reasoning and Rule Systems*. Galway, Ireland, pp. 1–1, 2011a.

Arenas, M. Exchanging More than Complete Data. In *Description Logics Workshop*. Barcelona, Spain, pp. 1–1, 2011b.

Arenas, M., Barceló, P., Libkin, L., and Murlak, F. *Relational and XML Data Exchange*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2010.

Arenas, M., Barceló, P., and Reutter, J. L. Datalog as a Query Language for Data Exchange Systems. In *Proceedings of Datalog Reloaded*. Oxford, UK, pp. 302–320, 2010.

Arenas, M., Barceló, P., and Reutter, J. L. Query Languages for Data Exchange: beyond unions of conjunctive queries. *Theory of Computing Systems* 49 (2): 489–564, 2011.

Arenas, M., Bertossi, L. E., and Chomicki, J. Consistent Query Answers in Inconsistent Databases. In *Proceedings of the ACM Symposium on Principles of Database Systems*. Philadelphia, USA, pp. 68–79, 1999.

Arenas, M., Botoeva, E., Calvanese, D., Ryzhikov, V., and Sherkhonov, E. Exchanging Description Logic Knowledge Bases. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*. Rome, Italy, pp. 563–567, 2012.

Arenas, M., Conca, S., and Pérez, J. Counting Beyond a Yottabyte, or how SPARQL 1.1 Property Paths will Prevent Adoption of the Standard. In *Proceedings of the International World Wide Web Conferences*. Lyon, France, pp. 629–638, 2012.

Arenas, M., Fagin, R., and Nash, A. Composition with Target Constraints. *Logical Methods in Computer Science* 7 (3): 1–38, 2011.

Arenas, M., Gutierrez, C., Miranker, D. P., Pérez, J., and Sequeda, J. Querying Semantic Data on the Web. *SIGMOD Record* 41 (4): 6–17, 2012.

Arenas, M., Gutierrez, C., and Pérez, J. Foundations of RDF Databases. In *Proceedings of Reasoning Web*. Bolzano, Italy, pp. 158–204, 2009.

Arenas, M. and Pérez, J. Querying Semantic Web Data with SPARQL. In *Proceedings of the ACM Symposium on Principles of Database Systems*. Athens, Greece, pp. 305–316, 2011.

Arenas, M. and Pérez, J. Federation and Navigation in SPARQL 1.1. In *Proceedings of Reasoning Web*. Vienna, Austria, pp. 78–111, 2012.

Arenas, M., Pérez, J., and Reutter, J. L. Data Exchange beyond Complete Data. In *Proceedings of the ACM Symposium on Principles of Database Systems*. Athens, Greece, pp. 83–94, 2011.

Arenas, M., Pérez, J., Reutter, J. L., and Riveros, C. Composition and Inversion of Schema Mappings. *SIGMOD Record* 38 (3): 17–28, 2009a.

Arenas, M., Pérez, J., Reutter, J. L., and Riveros, C. Inverting Schema Mappings: bridging the gap between theory and practice. *Proceedings of the VLDB Endowment* 2 (1): 1018–1029, 2009b.

Arenas, M., Pérez, J., Reutter, J. L., and Riveros, C. Foundations of Schema Mapping Management. In *Proceedings of the ACM Symposium on Principles of Database Systems*. Indianapolis, USA, pp. 227–238, 2010.

Arenas, M., Pérez, J., Reutter, J. L., and Riveros, C. Query Language-based Inverses of Schema Mappings: semantics, computation, and closure properties. *VLDB Journal* 21 (6): 823–842, 2012.

Arenas, M., Pérez, J., Reutter, J. L., and Riveros, C. The Language of Plain SO-tgds: composition, inversion and structural properties. *Journal of Computer and System Sciences* 79 (6): 763–784, 2013.

Arenas, M., Pérez, J., and Riveros, C. The Recovery of a Schema Mapping: bringing exchanged data back. *ACM Transactions on Database Systems* 34 (4), 2009.

Barceló, P. Logical Foundations of Relational Data Exchange. *SIGMOD Record* 38 (1): 49–58, 2009.

Barceló, P. Querying Graph Databases. In *Proceedings of the ACM Symposium on Principles of Database Systems*. New York, USA, pp. 175–188, 2013.

Barceló, P., Figueira, D., and Libkin, L. Graph-logics with Rational Relations and the Generalized Intersection Problem. In *Proceedings of the IEEE/ACM Symposium on Logic in Computer Science*. Dubrovnik, Croatia, pp. 115–124, 2012.

Barceló, P., Libkin, L., Lin, A. W., and Wood, P. T. Expressive Languages for Path Queries over Graph-Structured Data. *ACM Transactions on Database Systems* 37 (4): 31, 2012.

Barceló, P., Pérez, J., and Reutter, J. L. Relative Expressiveness of Nested Regular Expressions. In *Proceedings of the Alberto Mendelzon International Workshop on Foundations of Data Management*. Ouro Preto, Brazil, pp. 180–195, 2012.

Barceló, P., Pérez, J., and Reutter, J. L. Schema Mappings and Data Exchange for Graph Databases. In *Proceedings of the International Conference on Database Theory*. Genoa, Italy, pp. 189–200, 2013.

Barceló, P., Romero, M., and Vardi, M. Y. Semantic Acyclicity on Graph Databases. In *Proceedings of the ACM Symposium on Principles of Database Systems*. New York, USA, pp. 237–248, 2013.

Barceló, P., Libkin, L., and Reutter, J. Querying Graph Patterns. In *Proceedings of the ACM Symposium on Principles of Database Systems*. Athens, Greece, pp. 199–210, 2010.

Barceló, P., Libkin, L., and Romero, M. Efficient Approximations of Conjunctive Queries. In *Proceedings of the ACM Symposium on Principles of Database Systems*. Scottsdale, USA, pp. 249–260, 2012.

Bravo, L., Cheney, J., and Fundulaki, I. Repairing Inconsistent XML Write-Access Control Policies. In *Proceedings of the International Workshop on Database Programming Languages*. Vienna, Austria, pp. 97–111, 2007.

Bravo, L., Cheney, J., and Fundulaki, I. ACCOn: checking consistency of XML write-access control policies. In *Proceedings of the International Conference on Extending Database Technology*. Nantes, France, 2008.

Bravo, L., Cheney, J., Fundulaki, I., and Segovia, R. Consistency and Repair for XML Write-Access Control Policies. *VLDB Journal* 21 (6): 843–867, 2012.

Bravo, L. and Rodríguez, M. A. Formalization and Reasoning about Spatial Semantic Integrity Constraints. *Data Knowledge and Engineering* vol. 72, pp. 63–82, 2012.

Bravo, L. and Segovia, R. Simplified Access Control Policies for XML Databases. In *Alberto Mendelzon International Workshop on Foundations of Data Management*. Ouro Preto, Brazil, pp. 20–34, 2012.

Brisaboa, N., Luaces, M., Rodríguez, M. A., and Seco, D. An Inconsistency Measure of Spatial Data Sets with respect to Topological Constraints. *International Journal of Geographic Information Science (accepted)*, 2013.

Buil-Aranda, C., Arenas, M., and Corcho, Ó. Semantics and Optimization of the SPARQL 1.1 Federation Extension. In *Proceedings of the European Semantic Web Conference*. Heraklion, Greece, pp. 1–15, 2011.

Buil-Aranda, C., Arenas, M., Corcho, Ó., and Polleres, A. Federating Queries in SPARQL 1.1: syntax, semantics and evaluation. *Journal of Web Semantics* 18 (1): 1–17, 2013.

del Mondo, G., Rodríguez, M. A., Claramunt, C., and Bravo, L. Modeling Consistency of Spatio-Temporal Graphs. *Data Knowledge and Engineering* vol. 84, pp. 58–80, 2013.

Gutiérrez, C., Hurtado, C. A., and Mendelzon, A. O. Foundations of Semantic Web Databases. In *Proceedings of the ACM Symposium on Principles of Database Systems*. Paris, France, pp. 95–106, 2004.

Gutierrez, C., Hurtado, C. A., Mendelzon, A. O., and Pérez, J. Foundations of Semantic Web databases. *Journal of Computer and System Sciences* 77 (3): 520–541, 2011.

Gutiérrez, C., Hurtado, C. A., and Vaisman, A. A. Temporal RDF. In *Proceedings of the European Semantic Web Conference*. Heraklion, Greece, pp. 93–107, 2005.

Letelier, A., Pérez, J., Pichler, R., and Skritek, S. Static Analysis and Optimization of Semantic Web Queries. In *Proceedings of the ACM Symposium on Principles of Database Systems*. Scottsdale, USA, pp. 89–100, 2012.

Pérez, J., Arenas, M., and Gutierrez, C. Semantics and Complexity of SPARQL. *ACM Transactions on Database Systems* 34 (3), 2009.

Pérez, J., Arenas, M., and Gutierrez, C. nSPARQL: a navigational language for RDF. *Journal of Web Semantics* 8 (4): 255–270, 2010.

Pérez, J., Pichler, R., Sallinger, E., and Savenkov, V. Union and Intersection of Schema Mappings. In *Proceedings of the Alberto Mendelzon International Workshop on Foundations of Data Management*. Ouro Preto, Brazil, pp. 129–141, 2012.

Prud'hommeaux, E. and Buil-Aranda, C. SPARQL 1.1 Federated Query, W3C Recommendation. `http://www.w3.org/TR/sparql11-federated-query/`, 2013.

Rodríguez, A. and Gutierrez, C. A Formal Approach to Qualitative Reasoning on Topological Properties of Networks. In *Proceedings of the International Conference on Managing Knowledge in a World of Networks*. Podebrady, Czech Republic, pp. 358–365, 2006.

Rodríguez, M. A., Bertossi, L. E., and Marileo, M. C. Consistent Query Answering under Spatial Semantic Constraints. *Information Systems* 38 (2): 244–263, 2013.

Rodríguez, M. A. and Bravo, L. Multi-Granular Schemas for Data Integration. In *Proceedings of the Alberto Mendelzon International Workshop on Foundations of Data Management*. Vol. 866. Ouro Preto, Brazil, pp. 142–153, 2012.

Rodríguez, M. A. and Egenhofer, M. J. Determining Semantic Similarity among Entity Classes from Different Ontologies. *IEEE Transactions on Knowledge and Data Engineering* 15 (2): 442–456, 2003.

Tim Berners-Lee, J. H. and Lassila, O. The Semantic Web. *Scientific American*, 2001.