On the Expressive Power of the Relational Algebra on Finite Sets of Relation Pairs

George H.L. Fletcher, Marc Gyssens, Jan Paredaens, and Dirk Van Gucht

***Abstract*—We give a language-independent characterization of the expressive power of the relational algebra on finite sets of source-target relation instance pairs. The associated decision problem is shown to be co-graph-isomorphism hard and in coNP. The main result is also applied in providing a new characterization of the generic relational queries.**

***Index Terms*—Query languages, relational algebra, data mapping, data integration, definability, expressibility, BP completeness, graph isomorphism, genericity, monotonicity.**

I. Introduction

We investigate a generalization of the classic result of Bancilhon and Paredaens on the expressive power of the relational algebra [1], [3], [10] concerning the following decision problem:

> BP-PAIR. Given a pair of relations (s, t), with s non-empty or t of positive arity, does there exist a relational algebra expression E such that $E(s) = t$?

Bancilhon and Paredaens established that BP-PAIR is equivalent to the problem of determining whether or not (1) every atom occurring in t also occurs in s, and (2) every automorphism of s is also an automorphism of t. To date, the complexity of BP-PAIR has not been established.

Example 1: Consider the following pairs of source/target instances.

s_1	
a	a
b	b

s_2	
b	b
c	c

s_3	
a	a
b	b
a	b

t_1	
a	a
b	b

t_2	
b	b
c	c

t_3

Clearly, each pair (s_i, t_i) satisfies BP-PAIR conditions (1) and (2), and hence, for each $i = 1, 2, 3$, there exists a relational algebra expression E_i such that $E_i(s_i) = t_i$.

It is also the case that there exists a single expression E such that $E(s_i) = t_i$, for each $i = 1, 2, 3$; for example, the expression $s - (s \times \pi_{\langle\rangle}(\sigma_{1\neq 2}(s)))$ behaves properly on each source instance. Suppose that t_2 also has tuple $\langle c, b\rangle$. In this case (s_2, t_2) violates condition (2), and hence there no longer exists an expression E_2 such that $E_2(s_2) = t_2$ (and consequently, there also no longer exists a single expression for mapping all pairs). What if we were to additionally add tuple $\langle b, c\rangle$ to t_2? In this case (s_2, t_2) again satisfies both (1) and (2), and hence there exists an expression E_2 such that $E_2(s_2) = t_2$. Unfortunately, in this case there still does not exist a general expression E which behaves properly on each (s_i, t_i). This does not follow, however, from either condition (1) or (2). What is it about this *set* of instances that makes it unmappable? Is it possible to characterize the sets that *are* mappable?

A. The Problem

Towards resolving such questions about the expressive power of the relational algebra on sets of source/target instance pairs, in this note we introduce and study the following generalized decision problem:

> BP-PAIRS. Given a set of pairs of relations $\{(s_1, t_1), \ldots, (s_k, t_k)\}$, $k \geqslant 1$, with

George Fletcher is with Washington State University, Vancouver. e-mail: fletcher@vancouver.wsu.edu

Marc Gyssens is with Hasselt University and the Transnational University of Limburg. e-mail: marc.gyssens@uhasselt.be

Jan Paredaens is with the University of Antwerp. e-mail: jan.paredaens@ua.ac.be

Dirk Van Gucht is with Indiana University, Bloomington. e-mail: vgucht@cs.indiana.edu

each s_i of arity $m \geqslant 0$ and each t_i of arity $n \geqslant 0$, does there exist a relational algebra expression E such that $E(s_i) = t_i$ for $i = 1, \ldots, k$?

Note that BP-PAIRS allows empty source relations. It is clear that the classic BP-PAIR problem reduces to a strict special case of the generalized BP-PAIRS problem (namely, where $k = 1$, the source relation is non-empty, and $n \geqslant 1$).

B. Practical Significance

The present investigation was motivated by practical query discovery problems arising in the context of recent research on data integration, extraction, and exchange. In each of these domains, a crucial problem is the instance-driven discovery of mapping queries between autonomous data sources. In the context of data integration, recent research has explored the use of corresponding example instances of source and target schemas in the derivation of appropriate source-to-target data mapping queries [2], [4]. In the context of data extraction, an extensive line of research has explored the use of example instances to derive "wrapper" queries for extraction of relevant information from data sources (e.g., [6]). In the context of data exchange, an important issue is the discovery of source-to-target dependencies for translation of instances of a source schema into appropriate instances of a target schema (cf. [9]). Important issues in each of these contexts are to characterize the goodness of sets of examples for query discovery and to understand the complexity of such derivations. The ubiquity of such instance-based reasoning in a variety of query discovery tasks led us to the present study of BP-PAIRS.

C. Summary of Results

In this note we first give an exact language-independent characterization of when a solution to a BP-PAIRS instance exists and show how to construct an appropriate mapping expression E when this is the case. Next, we establish that BP-PAIRS is co-graph-isomorphism-hard and in coNP. We then use these results to give a new characterization of the generic relational queries. We close by indicating topics for further investigation.

II. Preliminary Notions

In this section we give basic definitions and notation used in this note.

Definition 1: A *relation* r of *arity* $n \in \mathbb{N}$ is a finite subset of n Cartesian products of an infinitely enumerable domain $\mathbb{D}$ of uninterpreted atoms: $r \subset \mathbb{D}^n$. The *active domain* of r is the set of atoms occurring in r, denoted as $adom(r) = \bigcup_{i=1}^{n} \{a_i \mid \langle a_1, \ldots, a_i, \ldots, a_n \rangle \in r\}$.

Notice that there are only two 0-ary relations: the empty relation $\{\}$ and the relation with the empty tuple $\{\langle\rangle\}$. These are often used to encode **false** and **true**, respectively, as relations. In this way, boolean queries can be embedded in the relational model.

Definition 2: An *isomorphism* φ from a relation r of arity n to a relation s of arity n is a permutation of $\mathbb{D}$ such that, for all $a_1, \ldots, a_n \in \mathbb{D}$, it is the case that $\langle a_1, \ldots, a_n \rangle \in r$ if and only if $\langle \varphi(a_1), \ldots, \varphi(a_n) \rangle \in s$.

Definition 3: An *automorphism* φ of a relation r of arity n is an isomorphism from r to itself, i.e., for all $a_1, \ldots, a_n \in \mathbb{D}$, it is the case that $\langle a_1, \ldots, a_n \rangle \in r$ if and only if $\langle \varphi(a_1), \ldots, \varphi(a_n) \rangle \in r$. The set of automorphisms of r is denoted $Aut(r)$.

Notice that the restriction of an automorphism of r to $adom(r)$ is necessarily a permutation of $adom(r)$.

Definition 4: A *BP-set* is a finite set $\{(s_1, t_1), \ldots, (s_k, t_k)\}$ of $k \geqslant 1$ pairs of relations, such that, for $i = 1, \ldots, k$, s_i is of arity $m \geqslant 0$, and t_i is of arity $n \geqslant 0$.

We follow Paredaens' presentation of the *relational algebra* [10], extended with a constant operator **unit**. In what follows, for a tuple $t = \langle a_1, \ldots, a_n \rangle \in \mathbb{D}^n$ we denote by $t[i]$ the ith component of t, i.e., $t[i] = a_i$ for $1 \leqslant i \leqslant n$.

Definition 5: Let r and s be relations of arity m and n, respectively. The *relational algebra* is the set of well-formed expressions containing relation names and closed under the following seven operations on relations.

- The *product* of r and s is the relation $r \times s = \{\langle a_1, \ldots, a_m, b_1, \ldots, b_n \rangle \mid \langle a_1, \ldots, a_m \rangle \in r \text{ and } \langle b_1, \ldots, b_n \rangle \in s\}$.
- The *union* of r and s is the relation $r \cup s = \{t \mid t \in r \text{ or } t \in s\}$, which is only defined when $m = n$.
- The *difference* of r and s is the relation $r - s = \{t \mid t \in r \text{ and } t \notin s\}$, which is only defined when $m = n$.

- The *projection* of r on $\langle j_1, \ldots, j_\ell \rangle$ ($\ell \geqslant 0$ and, for $i = 1, \ldots, \ell$, $1 \leqslant j_i \leqslant m$}) is the relation $\pi_{\langle j_1,\ldots,j_\ell \rangle}(r) = \{\langle t[j_1], \ldots, t[j_\ell] \rangle \mid t \in r\}$.
- The *equality selection* of r on i and j (for $1 \leqslant i, j \leqslant m$) is the relation $\sigma_{i=j}(r) = \{t \mid t \in r \text{ and } t[i] = t[j]\}$.
- The *inequality selection* of r on i and j (for $1 \leqslant i, j \leqslant m$) is the relation $\sigma_{i \neq j}(r) = \{t \mid t \in r \text{ and } t[i] \neq t[j]\}$.
- The *unit* of r is the relation $\mathbf{unit}(r) = \{\langle\rangle\}$.

If E is an expression over relation names $R_1, \ldots, R_k$, then $E(r_1, \ldots, r_k)$ denotes the relation which results from the evaluation of E with each R_i bound to relation r_i, for $1 \leqslant i \leqslant k$.

Finally, we give a standard semantic notion for relational mappings.

Definition 6: A mapping Q from relations of some arity $m \geqslant 0$ to relations of some arity $n \geqslant 0$ is *generic* if, for each relation r of arity m and each permutation φ of $\mathbb{D}$, it is the case that $\varphi(Q(r)) = Q(\varphi(r))$.

III. Resolving the BP-PAIRS Problem

In Example 1, we claimed that the addition of tuples $\langle b, c \rangle$ and $\langle c, b \rangle$ to t_2 would make the BP-set $\{(s_1, t_1), (s_2, t_2), (s_3, t_3)\}$ an invalid instance of BP-PAIRS, despite the fact that each pair in the set is a valid instance of BP-PAIR. We now prove this. In particular, we now present the main result, a language-independent characterization of the expressive power of the relational algebra on BP-sets.

Theorem 1: Let $\{(s_1, t_1), \ldots, (s_k, t_k)\}$ be a BP-set. The following statements are equivalent:

1) There is a relational algebra expression E such that, for $i = 1, \ldots, k$, $E(s_i) = t_i$,
2) It holds that
   a) for $i = 1, \ldots, k$, $\mathit{adom}(t_i) \subseteq \mathit{adom}(s_i)$; and
   b) if φ is an isomorphism from s_i to s_j ($1 \leqslant i, j \leqslant k$), then it is also an isomorphism from t_i to t_j.

Proof: ($1 \Rightarrow 2$) This implication follows immediately, since relational algebra queries are both domain-preserving and generic [1], [3], [10].

($2 \Rightarrow 1$) Let m and n be the arity of each s_i and t_i, respectively, for $1 \leqslant i \leqslant k$.

First, we observe that , for each i, the pair (s_i, t_i) is an instance of the classic BP-PAIR problem. By putting $i = j$, condition (2b) implies that each automorphism of s_i is also an automorphism of t_i. Hence, there exists a relational algebra expression E_i taking one m-ary relation as argument and returning an n-ary relation as result such that $E_i(s_i) = t_i$. We must note here that the border cases $m = 0$ and/or $n = 0$ were not explicitly considered in the original proof of Paredaens [10]. However, the expression $E_i(r) = r - r$ will do in the case that $t_i = \{\}$, irrespective of m, and the expression $E_i(r) = \mathbf{unit}(r)$ will do in the case that $t_i = \{\langle\rangle\}$, irrespective of m.

Next, we note that for each s_i there is a relational algebra expression F_i such that a relation r is isomorphic with s_i if and only if $F_i(r) \neq \emptyset$. This fact was already shown by Bancilhon for the case when $s_i \neq \{\}$ [1]. In the case when $s_i = \{\}$, the boolean expression $F_i(r) = \mathbf{unit}(r) - \pi_{\langle\rangle}(r)$ identifies relations isomorphic with s_i.

Combining these results, the following relational algebra expression fulfills (1):

$$E(r) = \bigcup_{1 \leqslant i \leqslant k} \pi_{\langle 1,\ldots,n \rangle}(E_i(r) \times F_i(r)).$$

■

It is interesting to note that the proof of Theorem 1 provides an explicit PSPACE construction of an appropriate mapping expression, as is the case for the proof of Paredaens [10] of the BP-PAIR result.

At this point, however, we must emphasize that there is a fundamental difference between the classic BP-PAIR result and the BP-PAIRS result. The proof of Paredaens [10] of the BP-PAIR result reveals that the difference operator is not used in the construction of the required relational algebra expression in the case that $m, n \geqslant 1$ and both source and target are nonempty. The expressions constructed are thus *monotone* in the sense that $r_1 \subseteq r_2$ implies $E(r_1) \subseteq E(r_2)$.

In the expressions F_i ($1 \leqslant i \leqslant k$) in the proof of Theorem 1, the difference operator *is* used. This is not an incidental effect of the particular construction used, even in the case that $m, n \geqslant 1$ and both source and target are nonempty. Indeed, solutions to BP-PAIRS make essential use of the difference operator since BP-sets can capture nonmonotone query behavior (since k can be greater than 1), and the relational algebra expressions without difference are always monotone.[1]

Example 2: Consider the following pairs of source/target instances.

s_1	
a	a

s_2	
a	a
b	b

t_1	
a	a

t_2	
a	b
b	a

The BP-set $\{(s_1, t_1), (s_2, t_2)\}$ satisfies condition (2) of Theorem 1. Hence, there exists a relational algebra expression E such that $E(s_1) = t_1$ and $E(s_2) = t_2$. Obviously, E cannot be monotone, and therefore must contain the difference operator. This is also the case for the BP-set of Example 1.

IV. Complexity of the BP-PAIR and BP-PAIRS Problems

We next relate the complexity of BP-PAIR and BP-PAIRS to several well known graph decision problems. First, we present some terminology.

Definition 7: A *graph* G is a binary relation $\mathcal{E}$ over a finite domain $V \subset \mathbb{D}$. We write $G = (V, \mathcal{E})$, where V is called the set of *vertices* and $\mathcal{E} \subseteq V \times V$ is called the set of *edges*.

Definition 8: Two graph decision problems.

- Subgraph Isomorphism (SubGI): given two graphs G_1 and G_2, is G_1 isomorphic to a subgraph of G_2?
- Graph Isomorphism (GI): given two graphs G_1 and G_2, are they isomorphic?

SubGI is a typical NP-complete problem [8]. Clearly GI is also in NP; it is unknown, however, whether GI is in P, is NP-complete, or neither [8].

We immediately observe the following.

Lemma 1: BP-PAIRS is in coNP.

Proof: Recall that coNP is the class of problems which have polynomial time *disqualifications* (for example, see [8]). Given a BP-set $\{(s_1, t_1), \ldots, (s_k, t_k)\}$, then guess an i, a j, and an isomorphism φ from s_i to s_j, and check in polynomial-time whether or not φ is also an isomorphism from t_i to t_j. If not, then, using the characterization of BP-PAIRS (Theorem 1), reject $\{(s_1, t_1), \ldots, (s_k, t_k)\}$. ∎

Definition 9: Given a relation r and atom $v \in \mathbb{D}$, define $rv = r \times \{\langle v \rangle\}$.

We denote by $\overline{\mathsf{P}}$ the complement of decision problem P, and by $\mathsf{P} \leq_m^p \mathsf{P}'$ that P polynomial time many-one reduces to problem P′ [8].

We can now show the main result of this section.

Theorem 2:

$$\overline{\mathsf{GI}} \leq_m^p \mathsf{BP\text{-}PAIR} \leq_m^p \mathsf{BP\text{-}PAIRS} \leq_m^p \overline{\mathsf{SubGI}}.$$

Proof: We establish the first reduction by exhibiting a polynomial time many-one reduction f from GI to $\overline{\mathsf{BP\text{-}PAIR}}$. Let $G_1 = (V_1, \mathcal{E}_1)$ and $G_2 = (V_2, \mathcal{E}_2)$ be a pair of graphs, and assume, without loss of generality, that $V_1 \cap V_2 = \emptyset$. If $\mathcal{E}_1$ or $\mathcal{E}_2$ is non-empty, define $f(G_1, G_2) = (\mathcal{E}_1 v_1 \cup \mathcal{E}_2 v_2, \{\langle v_1 \rangle\})$, where v_1 and v_2 are two different elements of $\mathbb{D} - (V_1 \cup V_2)$. Otherwise, define $f(G_1, G_2) = (\{\langle u \rangle, \langle v \rangle\}, \{\langle u \rangle\})$ where u and v are two different elements of $\mathbb{D}$. Clearly f is polynomial time computable. If $(G_1, G_2) \in \mathsf{GI}$ and $\mathcal{E}_1$ or $\mathcal{E}_2$ is not empty, then there exists an isomorphism φ from G_1 to G_2. If we extend φ such that $\varphi(v_1) = v_2$, then φ is an automorphism of $\mathcal{E}_1 v_1 \cup \mathcal{E}_2 v_2$. But then, by Theorem 1, $f(G_1, G_2) \notin \mathsf{BP\text{-}PAIR}$, since φ is not an automorphism of $\{\langle v_1 \rangle\}$. Now, if $(G_1, G_2) \notin \mathsf{GI}$, then for each $\varphi \in Aut(\mathcal{E}_1 v_1 \cup \mathcal{E}_2 v_2)$ it clearly must be the case that $\varphi(v_1) = v_1$ and $\varphi(v_2) = v_2$. By Theorem 1, it follows immediately that $f(G_1, G_2) \in \mathsf{BP\text{-}PAIR}$. Finally, if $\mathcal{E}_1$ and $\mathcal{E}_2$ are both empty, then clearly $(G_1, G_2) \in \mathsf{GI}$ if and only if $f(G_1, G_2) \notin \mathsf{BP\text{-}PAIR}$.

The second reduction follows directly from the definition of BP-PAIRS. The third reduction follows from Lemma 1 since SubGI is NP-complete. ∎

V. An Observation on Generic Queries

As an application of Theorem 1, we have the following novel characterization of the generic relational queries.

Theorem 3: Let Q be a mapping from relations of arity $m \geqslant 0$ to relations of arity $n \geqslant 0$. Then the following statements are equivalent:

1) Q is generic.
2) For any finite set $\mathcal{R}$ of relations of arity m, there is a relational algebra expression $E_{\mathcal{R}}$ such that, for every $r \in \mathcal{R}$, $E_{\mathcal{R}}(r) = Q(r)$.

[1]For the same reason, one cannot simply reduce the BP-PAIRS problem for a BP-set $\{(s_1, t_1), \ldots, (s_2, t_2)\}$ to the BP-PAIR problem for the pair $(s_1 \times \cdots \times s_k, t_1 \times \cdots \times t_k)$, even in the case that $m, n \geqslant 1$ and all relations under consideration are nonempty.

3) For any pair $\mathcal{R} = \{r_1, r_2\}$ of relations of arity m, there is a relational algebra expression $E_\mathcal{R}$ such that, for $i = 1, 2$, $E_\mathcal{R}(r_i) = Q(r_i)$.

Proof: ($1 \Rightarrow 2$) Let $\mathcal{R} = \{r_1 \ldots, r_k\}$. Consider the pairs $(r_i, Q(r_i))$, $i = 1, \ldots, k$. Suppose that, for $i, j = 1, \ldots, k$, φ is an isomorphism from r_i to r_j. Extend φ in an arbitrary way to a permutation of $\mathbb{D}$. Since Q is generic, $\varphi(Q(r_i)) = Q(\varphi(r_i)) = Q(r_j)$. Hence, φ is an isomorphism from $Q(r_i)$ to $Q(r_j)$. Since it is also the case that $\mathit{adom}(Q(r_i)) \subseteq \mathit{adom}(r_i)$, we have from Theorem 1 that there exists a relational algebra expression $E_\mathcal{R}$ such that, for every $r \in \mathcal{R}$, $E_\mathcal{R}(r) = Q(r)$.

($2 \Rightarrow 3$) Obvious.

($3 \Rightarrow 1$) Let r be a relation of arity m and φ be a permutation on $\mathbb{D}$. Let $\mathcal{R} = \{r, \varphi(r)\}$. By assumption, there exists a relational algebra expression $E_\mathcal{R}$ such that $E_\mathcal{R}(r) = Q(r)$ and $E_\mathcal{R}(\varphi(r)) = Q(\varphi(r))$. Since the relational algebra is generic, we have that $\varphi(Q(r)) = \varphi(E_\mathcal{R}(r)) = E_\mathcal{R}(\varphi(r)) = Q(\varphi(r))$. ■

Theorem 3 highlights once more the fundamental difference between the classic BP-PAIR case and the BP-PAIRS case. Not only does the proof of Theorem 3 heavily rely on the the fact that $|\mathcal{R}| > 1$, but furthermore, without this condition the result simply does not hold. To see this, let $a \in \mathbb{D}$. Consider the mapping Q for which $Q(\{\langle a\rangle\}) = \{\langle a\rangle\}$ and $Q(r) = \emptyset$ for $r \neq \{\langle a\rangle\}$. Clearly, Q is computable, but not generic. To see this, choose $b \in \mathbb{D}$ such that $a \neq b$. Consider the permutation of $\mathbb{D}$ that swaps a and b and fixes all other elements of $\mathbb{D}$. While this permutation is an isomorphism from $\{\langle a\rangle\}$ to $\{\langle b\rangle\}$, it is not an isomorphism from $Q(\{\langle a\rangle\}) = \{\langle a\rangle\}$ to $Q(\{\langle b\rangle\}) = \emptyset$. Nevertheless, Q satisfies statement (2) of Theorem 3 for $|\mathcal{R}| = 1$.

VI. Final Remarks

All of the results established above also hold for the natural generalization of BP-PAIRS to the *nested* relational model, following Gyssens et al. [12]. It may also prove fruitful to investigate similar generalizations of instance-driven query discovery for graph [5] and XML [7] data. We close by noting several further open questions which naturally arise from the present investigation.

- Recently, results have been established on the complexity of repairing data mapping *expressions* for several logical languages [11]. In the context of reasoning about BP-sets, one can dually consider repairing *instances* for data mapping discovery.
  - Suppose a BP-set only satisfies condition (2a) of Theorem 1. What is the minimal number of tuple additions and/or deletions required to "repair" the set such that it also satisfies condition (2b)? For some $k \geqslant 0$, can the set be repaired with at most k such updates?
  - Suppose a BP-set $\{(s_1, t_1), \ldots, (s_k, t_k)\}$ fails to satisfy condition (2a) of Theorem 1. Can one find a renaming of the atoms in $\bigcup_{1 \leqslant i \leqslant k} \mathit{adom}(s_i)$ with atoms in $\bigcup_{1 \leqslant i \leqslant k} \mathit{adom}(t_i)$ such that the set satisfies both conditions (2a) and (2b)? In other words, for the given BP-set, does there exist a relational algebra expression E and binary relation $\tau \subseteq \bigcup_{1 \leqslant i \leqslant k} \mathit{adom}(s_i) \times \bigcup_{1 \leqslant i \leqslant k} \mathit{adom}(t_i)$ such that $E(s_i, \tau) = t_i$, for each $1 \leqslant i \leqslant k$? Note that BP-PAIRS is just the special case of this problem where τ is restricted to subsets of the identity relation on $\bigcup_{1 \leqslant i \leqslant k} \mathit{adom}(s_i)$.

  Are there natural characterizations and practical algorithmic solutions for such *instance* repair problems?
- A BP-set can be thought of as a "sample" or finite "trace" of an infinite query. Although Theorem 1 provides an explicit means to construct an appropriate mapping query when possible, there is no guarantee that in practice this query is a "desirable," "interesting," or the "best" mapping for a given context. For example, consider again the BP-set of Example 1. In this case, the mapping expression constructed using Theorem 1 consists of a union of expressions, each of which consists of crossproducts and unions of sizeable subexpressions [10]. In contrast, we saw in the Example that the succinct expression $s - (s \times \pi_{\langle\rangle}(\sigma_{1 \neq 2}(s)))$ is sufficient. Consequently, towards applications of Theorem 1 it is important to develop meaningful notions of query interestingness and goodness-of-fit.

References

[1] F. Bancilhon. On the Completeness of Query Languages for Relational Data Bases. *Proc. MFCS*, Springer LNCS 64, pp. 112–123, Zakopane, Poland, 1978.

[2] A. Bilke and F. Naumann. Schema Matching using Duplicates. *Proc. IEEE ICDE*, pp. 69–80, Tokyo, 2005.

[3] A.K. Chandra and D. Harel. Computable Queries for Relational Data Bases. *J. Comput. Syst. Sci.* 21(2):156–178, 1980.

[4] G.H.L. Fletcher and C.M. Wyss. Data Mapping as Search. *Proc. EDBT*, Springer LNCS 3896, pp. 95–111, Munich, 2006.

[5] M. Gemis, J. Paredaens, P. Peelman, and J. Van den Bussche. Expressiveness and Complexity of Generic Graph Machines. *Theory Comput. Syst.* 31(3):231–249, 1998.

[6] G. Gottlob, C. Koch, R. Baumgartner, M. Herzog, and S. Flesca. The Lixto Data Extraction Project—Back and Forth between Theory and Practice. *Proc. ACM PODS*, pp. 1–12, Paris, 2004.

[7] M. Gyssens, J. Paredaens, D. Van Gucht, and G.H.L. Fletcher. Structural Characterizations of the Semantics of XPath as Navigation Tool on a Document. *Proc. ACM PODS*, pp. 318–327, Chicago, 2006.

[8] J. Köbler, U. Schöning, and J. Torán. *The Graph Isomorphism Problem: Its Structural Complexity.* Birkhäuser, Boston, 1993.

[9] P.G. Kolaitis. Schema Mappings, Data Exchange, and Metadata Management. *Proc. ACM PODS*, pp. 61–75, Baltimore, 2005.

[10] J. Paredaens. On the Expressive Power of the Relational Algebra. *Information Processing Letters* 7(2):107–111, 1978.

[11] P. Senellart and G. Gottlob. On the Complexity of Deriving Schema Mappings from Database Instances. *Proc. ACM PODS*, pp. 23–32, Vancouver, Canada, 2008.

[12] M. Gyssens, J. Paredaens, and D. Van Gucht. A Uniform Approach Toward Handling Atomic and Structured Information in the Nested Relational Database Model. *Journal of the ACM* 36(4):790–825, 1989.