Finitely Representable Databases

(Extended Abstract)

Stéphane Grumbach*
I.N.R.I.A.
Rocquencourt BP 105
78153 Le Chesnay, France
stephane.grumbach@inria.fr

Jianwen Su**
Computer Science Department
University of California
Santa Barbara, California 93106, USA
su@cs.ucsb.edu

Abstract

We study classes of infinite but finitely representable databases based on constraints, motivated by new database applications such as geographical databases. The mathematical framework is based on classical decidable first-order theories. We investigate the theory of finitely representable models and prove that it differs strongly from both classical model theory and finite model theory. In particular, we show that most of the well known theorems of either one fail (compactness, completeness, locality, 0/1 laws, etc.). An immediate consequence is the lack of tools to consider the definability of queries in the relational calculus over finitely representable databases. We illustrate this very challenging problem through some classical examples.

1 Introduction

Until recently, databases were considered to be finite collections of data items. New applications such as those involving temporal and spatial data (e.g. geographical databases) lead very naturally to more general data models allowing infinite collections of items to be stored in the database. This is possible under the assumption that the databases admit a finite representation.

There have been very few theoretical results on infinite databases. General results on the completeness of query languages for infinite recursive databases were reported by Hirst and Harel in [HH93], where two classes of recursive databases are considered, and it is shown that: (i) quantifier free first-order logic is complete on the class of all recursive databases, and (ii) a version of Chandra and Harel's *QL* [CH80] is complete on highly symmetric databases. In this paper, we consider classes of databases[1] which lie between these two extreme examples.

In their seminal paper, Kanellakis, Kuper and Revesz [KKR90], introduced *constraint query languages*. The basic idea is to generalize the relations of the relational model [Cod70] by defining generalized tuples as conjunctions of constraints. For instance, $x^2 + y^2 = 1 \wedge x \leq 0$ defines a binary generalized tuple. A generalized, or finitely representable, relation is then a finite set of such tuples. In the real plane, it results in an infinite set of points, or tuples, over $\mathcal{R}^2$. The relational calculus over finitely representable relations constitutes a constraint query language which admits an efficient bottom-up and declarative semantics. An equivalent algebra was proposed by Kanellakis and Goldin [KG94]. It is well suited to express geometric queries. Practical issues, such as the indexing of such databases have been considered by Kanellakis, Ramaswamy, Vengroff, and Vitter [KRVV93].

In [KKR90], the data complexity of both the relational calculus and inflationary Datalog with negation over constraint databases is studied. Different types of constraints are considered, such

*Work supported in part by Esprit Project BRA AMUSING.

**Work supported in part by NSF grants IRI-9109520 and IRI-9117094. A part of work was done while visiting I.N.R.I.A.

[1] The finitely representable databases are not necessarily stricto sensu recursive. Their domain can indeed be the set of real numbers. Nevertheless, as soon as we restrict the domain to a countable set (such as integers or rationals), finitely representable databases are recursive.

as dense linear order inequalities, real polynomial inequalities, etc. These constraints are based on decidable theories which admit the elimination of quantifiers. It is proved that, although the decision problem of the underlying theories may be high, the data complexity of the query languages is tractable. This shows the practical interest of this approach, which has been pursued in [Rev90, Kup93, KG94].

The goal of this paper is to understand the underlying logic of constraint query languages. In other words, we study first-order logic when the semantics is restricted to *finitely representable models*. Our motivation is to investigate the expressive power of query languages over finitely representable databases.

We prove that the model theory of finitely representable structures, differs strongly from the classical model theory of all structures. In particular, we prove that, like for finite model theory (see the survey by Fagin [Fag93]), most of the classical theorems of logic, such as the compactness and the completeness theorems, fail for finitely representable structures.

The failure of most of the known theorems of classical model theory, shows the need to develop new tools meaningful for finitely representable structures. Powerful tools have been developed, such as the locality by Gaifman [Gai81], or the 0/1 laws by Fagin [Fag76] in the case of finite model theory. The model theory of finitely representable structures differs also from finite model theory. None of these tools is known to work in the presence of some regular mathematical structure such as an order relation for instance.

The problem of the definability of a query over finitely representable databases is therefore a very challenging issue. The Ehrenfeucht-Fraïssé games [Fra54, Ehr61] are among the few tools which carry over on finitely representable structures. Complexity theoretic arguments can be used also to prove non-definability. Indeed, some rather low upper-bounds (in terms of boolean circuits) are known for the relational calculus over finitely representable databases [KG94]. We also investigate other criteria such as the invariance of the properties, their closed form, etc.

We illustrate these techniques on queries from computational geometry [Yao90], graph theory, and geographical databases. We investigate their definability in first-order logic over finitely representable databases. We show in particular the non first-order definability of the connectivity of an area in the context of a dense order. The proof can be made either with an Ehrenfeucht-Fraïssé game argument, or by a complexity argument.

The paper is organized as follows. In Section 2, we define finitely representable databases and consider first-order logic as a query language for finitely representable databases. In Section 3, we present the differences between finitely representable model theory and both finite and classical model theory. Finally, in Section 4, we give some examples of finitely representable databases and study the definability of some natural queries.

2 Finitely Representable Databases

Throughout the whole paper, we consider only countable *first-order languages* with *equality*. In this section, we introduce the theory of *finitely representable models*. We first introduce the main definitions, and illustrate them with some examples of finitely representable databases. We then consider first-order queries.

2.1 Finite Representability

We fix the following. Let $\mathcal{L}$ and $\mathcal{L}'$ be first-order languages (with equality) such that $\mathcal{L}' \subseteq \mathcal{L}$, and let $\sigma = \{R_1, ..., R_n\}$ be a signature (or a database schema) such that $\mathcal{L} \cap \sigma = \emptyset$, where $R_1, ..., R_n$ are relation symbols. We distinguish between *logical predicates* (e.g., $=, \leq$) in $\mathcal{L}$ and *relations* in σ.

Consider, for instance, the language $\{\leq, +, \times, 0, 1\}$ of the real field $\boldsymbol{\mathcal{R}}$. Kanellakis, Kuper, and Revesz [KKR90] introduced the concept of a k-ary generalized tuple, which is a constraint expressed as a conjunction of atomic formulas over k-variables. For instance, $(x \times x + y \times y = 1 \wedge x \leq 0)$ is a binary generalized tuple representing a half circle. A k-ary finitely representable relation (or generalized relation in [KKR90]) is then a finite set of k-ary generalized tuples.

In this framework, a tuple (a, b) in the classical relational database [Cod70] is an abbreviation for the formula $(x = a \wedge y = b)$ represented using only the equality symbol "$=$" and constants. A triangle $(x \leq y \wedge x \geq 0 \wedge y \leq 10)$ is also a generalized tuple represented using only "$\leq$" and constants. The half circle mentioned above is represented using $\{=,\leq,+,\times,0,1\}$.

Let $\sigma = \{R\}$, where R is a binary relation symbol. A finitely representable relation can be seen as an *expansion* of $\mathcal{R}$ to σ, i.e., a structure over the vocabulary $\{\leq,+,\times,0,1,R\}$, which coincides with $\mathcal{R}$ on $\{\leq,+,\times,0,1\}$. The new relation R is the database. R may be infinite, but it has to be representable in a *finite* way with $\{\leq,+,\times,0,1\}$.

More formally, assume that $\mathcal{A}$ is an $\mathcal{L}$-structure. We are interested in expansions of $\mathcal{A}$ to σ. Intuitively, the relations in σ constitute a database *in the context* of $\mathcal{A}$. We next define the fundamental concept of finitely representable relations and structures.

Definition 2.1 Let $\mathcal{A}$ be an $\mathcal{L}$-structure with universe A. Let $R \subseteq A^n$ be some n-ary relation. The relation R is *finitely representable in* $\mathcal{L}'$ (or $\mathcal{L}'$*-representable* for short) *over* $\mathcal{A}$ ($\mathcal{A}$ is omitted when it is known from the context) if there exists a quantifier-free formula $\varphi(x_1,...,x_k)$ in $\mathcal{L}'$ with k distinct free variables $x_1,...,x_k$ such that:

$$\mathcal{A} \models \forall x_1 \cdots x_k (R(x_1,...,x_k) \leftrightarrow \varphi(x_1,...,x_k)).$$

Let $\mathcal{B}$ be an expansion of $\mathcal{A}$ to σ. The structure $\mathcal{B}$ is said to be $\mathcal{L}'$*-representable (over* $\mathcal{A}$*)* if for every relation symbol R in σ, $R^{\mathcal{B}}$ is $\mathcal{L}'$-representable (over $\mathcal{A}$).

If an expansion $\mathcal{B}$ of $\mathcal{A}$ to σ is $\mathcal{L}'$-representable over $\mathcal{A}$, then $\mathcal{B}$ is also called an $\mathcal{L}'$*-representable* $(\mathcal{A},\sigma)$*-expansion*, and the restriction of $\mathcal{B}$ to σ, $\mathcal{B}|_\sigma$ is called an $\mathcal{L}'$*-representable* $(\mathcal{A},\sigma)$*-(database) instance*. When everything is clear from the context, we will just say an $(\mathcal{A},\sigma)$-instance or simply an instance. In the following, we fix $\mathcal{A}$ and σ, and consider the class $K(\mathcal{A},\sigma,\mathcal{L}')$ of $\mathcal{L}'$-representable $(\mathcal{A},\sigma)$-instances. If the cardinality of the language $\mathcal{L}'$ is countable, then $K(\mathcal{A},\sigma,\mathcal{L}')$ is effectively enumerable. Moreover, if $\mathcal{A}$ has a countable universe such as the integers or the rationals, then every relation in each $(\mathcal{A},\sigma)$-instance in $K(\mathcal{A},\sigma,\mathcal{L}')$ is recursive.

Note that $K(\mathcal{A},\sigma,\mathcal{L}')$ has interesting closure properties. Indeed, it is closed under finite union and intersection and moreover under complementation. This differs from finite model theory (the complement of a finite model is not finite).

Finally, if φ is a sentence in $\mathcal{L} \cup \sigma$, we denote by $K_\varphi(\mathcal{A},\sigma,\mathcal{L}')$ the collection of $\mathcal{L}'$-representable $(\mathcal{A},\sigma)$-instances satisfying φ, that is, for each $\mathcal{L}'$-representable $(\mathcal{A},\sigma)$-expansion $\mathcal{B}$, if $\mathcal{B} \models \varphi$, then $\mathcal{B}|_\sigma \in K_\varphi(\mathcal{A},\sigma,\mathcal{L}')$. (We may denote $K_\varphi(\mathcal{A},\sigma,\mathcal{L}')$ simply by $K_\varphi^{\mathcal{A}}$ when σ is clear from the context and $\mathcal{L}' = \mathcal{L}$.)

In general, a finitely representable relation may be infinite (of arbitrary cardinality, that is it does not have to be enumerable), but it admits a finite representation. A finite relation is representable using only the equality predicate (and constants), and therefore is an $\{=\}$-representable relation. The converse doesn't hold. Nevertheless, the following can be easily verified.

Proposition 2.1 A monadic relation is $\{=\}$-representable iff it is finite or co-finite.

Remark: For practical purposes, one can assume that the database contains the quantifier free formulas defining the relations. The rest is built-in.

In the examples of the paper we use finite representability in the context of the following languages and structures:

1. $\mathcal{L}_\leq = \{\leq\} \cup \mathcal{Q}$ (the rational numbers[2]) and $\mathcal{A} = (\mathcal{R},\leq)$, i.e., $\mathcal{L}_\leq$ contains an order and a countable set of constants and $\mathcal{A}$ is the ordered set of real numbers (it satisfies the theory of dense order without endpoints); and
2. $\mathcal{L}_\times = \{\leq,+,\times,0,1\}$ and $\mathcal{A} = \mathcal{R}$, the real field (it satisfies the theory of real closed fields).

2.2 Queries

The notion of a *database query* was introduced by Chandra and Harel [CH80] as a mapping Q from structures over a given signature σ to relations of a fixed arity n, which is *partial recursive* and satisfies the following *consistency criterion*: If two structures over σ, $\mathcal{A}$ and $\mathcal{B}$ are isomorphic by an isomorphism μ, then $Q(\mathcal{A})$ and $Q(\mathcal{B})$ are isomorphic by the same isomorphism μ. This criterion was then called "genericity" in the database literature.

[2] Note that in the case of $\mathcal{L}_\leq$, we assume that the language contains a countable set of constants. Since rational numbers are definable in $\mathcal{L}_\times$, they are not necessary.

In the context of finitely representable databases, we have to ensure that the morphisms do not alter the finite representability of the databases. Let $\mathcal{L}' \subseteq \mathcal{L}$, and assume that $\mathcal{A}$ is an $\mathcal{L}$-structure with universe A. A mapping μ from A to A is said to be *$\mathcal{L}'$-preserving with respect to* an $\mathcal{L}'$-representable $(\mathcal{A},\sigma)$-instance I, if [3] $\mu(I)$ is also an $\mathcal{L}'$-representable $(\mathcal{A},\sigma)$-instance.

Note that when I is finite (so $\{=\}$-representable), each permutation of the universe is $\{=\}$-preserving with respect to I.

A class of $\mathcal{L}'$-representable $(\mathcal{A},\sigma)$-instances K is said to be *weakly closed under automorphisms* of $\mathcal{A}$, if for each $\mathcal{L}'$-representable $(\mathcal{A},\sigma)$-instances I in K, and for each automorphism μ of $\mathcal{A}$ which is $\mathcal{L}'$-preserving with respect to I, $\mu(I)$ is also in K. We use the following definition of queries which extends naturally the definition of [CH80] for finite databases.

Definition 2.2 Let $\mathcal{L}' \subseteq \mathcal{L}$. A *boolean $\mathcal{L}'$-query* K over $\mathcal{A}$ and σ is a partial recursive collection of $\mathcal{L}'$-representable $(\mathcal{A},\sigma)$-instances weakly closed under automorphisms of $\mathcal{A}$.

The weak closure under automorphisms may seem to be a complicated condition. The motivations are that (i) we cannot consider the full closure, since automorphisms of $\mathcal{A}$ may map finitely representable instances to non finitely representable instances, and (ii) we want to consider as good a closure as possible. This is needed to prove invariance properties (see below).

We next consider queries defined using first-order formulas. We first consider boolean queries. The next proposition, shows that under the restriction to finitely representable models, every first-order sentence defines a boolean query.

Proposition 2.2 Let $\mathcal{A}$ be an $\mathcal{L}$-structure. For each constant-free sentence φ in $\mathcal{L}\cup\sigma$, $K^{\mathcal{A}}_{\varphi}$ is weakly closed under automorphisms of $\mathcal{A}$.

Proof: The proof is accomplished by an induction on the structure of the formulas representing the $\mathcal{L}$-representable relations. Consider an $\mathcal{L}$-representable expansion $\mathcal{M}$ of $\mathcal{A}$ to σ such that $\mathcal{M}|_{\sigma} \in K^{\mathcal{A}}_{\varphi}$ (assuming for simplicity that σ contains only one relation symbol R). Then:

$$\mathcal{M} \models \forall \bar{x}\ (R(\bar{x}) \leftrightarrow \bigvee_i \bigwedge_j Atom_{i,j}(\bar{x})),$$

where the $Atom_{i,j}(\bar{x})$ are (negation of) atomic formulas in $\mathcal{L}$. Define $\phi = \varphi|^{R}_{\psi}$, where $\psi = \bigvee_i \bigwedge_j Atom_{i,j}(\bar{x})$ and $\varphi|^{R}_{\psi}$ is the formula obtained from φ by replacing each occurrence of R with the formula ψ. It follows that $\mathcal{M} \models \phi$. Now let μ be an $\mathcal{L}$-preserving automorphism of $\mathcal{A}$ with respect to $\mathcal{M}$ and let $\mathcal{N} = \mu(\mathcal{M})$. By definition, $\mathcal{N}$ is an $\mathcal{L}$-representable $(\mathcal{A},\sigma)$-expansion. Since μ is an automorphism of $\mathcal{A}$ and ϕ a sentence in $\mathcal{L}$, it is straightforward that $\mathcal{N} \models \mu(\phi)$, where $\mu(\phi)$ is obtained from ϕ by substituting each occurrence of each constant a with $\mu(a)$. Because φ is constant-free, $\mu(\phi) = \varphi|^{R}_{\mu(\psi)}$, which implies that $\mathcal{N}|_{\sigma} \in K^{\mathcal{A}}_{\varphi}$. $\Box$

The above proposition can be generalized to sentences having constants by considering only automorphisms that are identity on the constants in φ.

The semantics restricted to finitely representable models (in $\mathcal{L}$) is well defined in the sense that each sentence φ in $\mathcal{L}\cup\sigma$ defines a boolean query, since $K^{\mathcal{A}}_{\varphi}$ is weakly closed under automorphisms of $\mathcal{A}$.

Consider now non-boolean queries. In general, not every formula with free variables defines a query. Let σ,τ be two signatures. Intuitively, a "query" $\sigma \to \tau$ is a mapping from $\mathcal{L}$-representable $(\mathcal{A},\sigma)$-instances to $\mathcal{L}$-representable $(\mathcal{A},\tau)$-instances. Formally, it is defined as a collection of $\mathcal{L}$-representable $(\mathcal{A},\sigma\cup\tau)$-instances.

Let $\sigma = \{R_1, ..., R_n\}$ be a database schema and $\tau = \{R_o\}$ where R_o is a k-ary relation. A first-order formula $\varphi(x_1,...,x_k)$ in $\mathcal{L}\cup\sigma$, whose non-logical relation symbols are in σ, and with free variables $\{x_1,...,x_k\}$, defines a class of structures K_{φ} such that $\mathcal{B} \in K_{\varphi}$ if and only if (i) $\mathcal{B}$ is an expansion of $\mathcal{A}$ to $\sigma\cup\tau$, (ii) the restriction $\mathcal{B}|_{\mathcal{L}\cup\sigma}$ of $\mathcal{B}$ to $\mathcal{L}\cup\sigma$ is an $\mathcal{L}$-representable $(\mathcal{A},\sigma)$-instance, and (iii) $R^{\mathcal{B}}_{o} = \{(a_1,...,a_k) \mid \mathcal{B}|_{\mathcal{L}\cup\sigma} \models \varphi(a_1,...,a_k)\}$.

K_{φ} does not necessarily define a query. Indeed, $R^{\mathcal{B}}_{o}$ may well not be $\mathcal{L}$-representable. The next proposition exhibits a sufficient and necessary condition on the structure $\mathcal{A}$, such that every formula in $\mathcal{L}$ defines a *query*.

Proposition 2.3 Let $\mathcal{A}$ be some $\mathcal{L}$-structure, and

[3] where μ is extended to relations (subsets of A^k, for some arity k) in the natural way.

T be the complete theory such that $\mathcal{A} \models T$. Then, the following conditions are equivalent.

1. T admits a quantifier elimination procedure.
2. Every formula φ in $\mathcal{L} \cup \sigma$ with k free variables defines a query $\sigma \rightarrow \tau$ over $\mathcal{L}$-representable $(\mathcal{A}, \sigma)$-instances.

The direction $(1 \Rightarrow 2)$ was used in [KKR90] to see that the query languages were well-defined for finitely representable databases. The previous proposition shows that this is a necessary condition, and therefore finitely representable databases can only be defined over models of theories admitting a quantifier elimination procedure.

A theory admitting a quantifier elimination procedure, that is of great use in the context of finitely representable databases, is the theory of real closed fields over the language $\{=, \leq, +, \times, 0, 1\}$ [Tar51]. It should be noted that the functions allowed in the language play a fundamental role in the quantifier elimination procedure. In particular, van den Dries showed that the extension of the previous theory to a language with an exponential function, $\{=, \leq, +, \times, e^x, 0, 1\}$ doesn't admit a quantifier elimination procedure [vdD82]. The theory of dense order without endpoints is another example of a theory admitting a quantifier elimination procedure that is of great use here.

We next define the notion of *invariance* of queries. Let $\mathcal{L}' \subseteq \mathcal{L}$ and $\mathcal{A}$ be an $\mathcal{L}$-structure with universe A. A 1-1 mapping from A onto A is an $\mathcal{L}'$*-automorphism* of $\mathcal{A}$ if it is an automorphism of the substructure of $\mathcal{A}$ over the language $\mathcal{L}'$. Thus, an $\emptyset$-automorphism is a bijection over the universe.

Definition 2.3 Let $\mathcal{A}$ be an $\mathcal{L}$-structure. A query K over $\mathcal{A}$ and σ is $(\mathcal{L} - \mathcal{L}')$*-invariant* if K is weakly closed under $\mathcal{L}'$-automorphisms of $\mathcal{A}$.

If φ is some sentence in $\mathcal{L}$, $K_\varphi^{\mathcal{A}}$ is weakly closed under automorphisms of $\mathcal{A}$, and hence is $\emptyset$-invariant, as shown in Proposition 2.2. More generally, there are very natural examples of queries definable in some language $\mathcal{L}$, which are $\mathcal{L}'$-invariant for some strict subset $\mathcal{L}' \subset \mathcal{L}$.

Example 2.1 Consider the real field $\boldsymbol{\mathcal{R}}$, with a binary relation R. We view R as an (infinite) set of points in $\mathcal{R}^2$. R is *region connected* if two arbitrary points in R are connected by a line lying entirely in R. It can be easily verified that the class K, of $\{\leq, +, \times, 0, 1\}$-representable $(\mathcal{R}, \sigma)$-instances such that R is region connected, is a query which is weakly closed under $\{\leq\}$-automorphisms of $\mathcal{R}$, and therefore is $\{+, \times\}$-invariant. $\Box$

2.3 Finite Databases

We consider briefly the case of finite databases. They can be viewed as finitely representable databases over the integers $\mathcal{N}$ (with or without order). The representation is based on the equality predicate only, and no negation is allowed in tuples. The concepts defined above carry over, and corresponds to well known concepts of (finite) database theory.

First note that any automorphism on $\mathcal{N}$ is $\mathcal{L}$-preserving in the case of finite databases. The notion of $\mathcal{L}$-invariance corresponds to the usual notion of genericity [AU79, CH80], and if C is a set of constant symbols, $(\mathcal{L} - C)$-invariance coincides with the notion of C-genericity [Hul86].

Example 2.2 Consider databases with one finite relation R in the context of the integers with an order relation $(\mathcal{N}, \leq)$. The *parity query*, true if the parity of the cardinality of the finite relation R is even, commutes with $\emptyset$-automorphisms of $(\mathcal{N}, \leq)$ which are automorphisms of $\mathcal{N}$, and so is $\{\leq\}$-invariant. $\Box$

In [Kup90], finite databases over the real closed field were considered. It was claimed that if a query in $\mathcal{L} = \{\leq, +, \times, 0, 1\}$ is $\mathcal{L}$-invariant, then it is definable using only equality. Unfortunately, this result doesn't hold [Via93, Kup93a]. A simple counterexample can be found in [AHV92]. Therefore, the definability of the transitive closure of a graph, or of the parity of the cardinality of a relation in $\mathcal{L} = \{\leq, +, \times, 0, 1\}$, is still an open question unlike stated in [Kup90]. We give partial answers in Section 4.

3 Finite Models, Finitely Representable Models, and all Models

In this section, we analyze the differences between finite model theory, finitely representable model theory, and classical model theory. We shall see that the theory of finitely representable models differs from classical model theory, and prove that, as for finite model theory, the compactness theorem fails, and the validity over finitely representable models is in general co-r.e. and not r.e.

We next define the *satisfiability over finitely representable models.*

Definition 3.1 A sentence φ in $\mathcal{L} \cup \sigma$ is *satisfiable over finitely representable models with respect to* $\mathcal{A}$ *and* σ if there exists an $\mathcal{L}$-representable $(\mathcal{A}, \sigma)$-expansion satisfying φ.

The *validity over finitely representable models* is defined similarly. It is important to note that we always assume some fixed $\mathcal{A}$ and σ. The semantics is then restricted to $(\mathcal{A}, \sigma)$-expansions. We next prove the failure of the compactness theorem.

Proposition 3.1 The *Compactness Theorem* fails for finitely representable models.

Proof: To illustrate the failure of the compactness theorem, we give the proof for two different contexts.

(i) $\mathcal{L} = \{=\}$: Consider the structure $\boldsymbol{\mathcal{N}} = (\mathcal{N}, =)$, where N is the set of natural numbers. Assume a relational schema σ with one monadic relation R. For each integer k, consider the formula τ_k stating that there are at least k distinct elements in R and at least k distinct elements not in R. Let $\Sigma = \{\tau_k \mid k \geq 0\}$. Every finite subset of Σ is obviously satisfiable over finitely representable models with respect to $\mathcal{N}$ and σ, while Σ is not, since any $\{=\}$-representable monadic relation is finite or co-finite (Proposition 2.1).

(ii) $\mathcal{L} = \{=, \leq, +, \times, 0, 1\}$: Consider the real field $\boldsymbol{\mathcal{R}}$. Assume again a relational schema σ with one monadic relation R. Let τ_i be the sentence stating that if $x \in R$ and $0 \leq x \leq i$, then $x \in \{0, 1, ..., i\}$. Define $\Sigma = \{\tau_i \mid i \geq 0\}$. Obviously each finite subset of Σ has a model with respect to $\boldsymbol{\mathcal{R}}$ and σ which is $\mathcal{L}$-representable. On the other hand, Σ does not admit any $\mathcal{L}$-representable model. It follows indeed from Tarski's result [Tar51] that any monadic first-order formula over the reals define a finite set of segments of $\mathcal{R}$. $\Box$

In classical model theory, the completeness theorem implies that the set of valid first-order sentences is r.e. It follows from Church's Theorem that, as soon as the language contains some relation symbol that is not unary, the set of valid first-order sentences is not co-r.e. The contrary holds for finite model theory. Indeed, the set of first-order sentences valid over finite structures is co-r.e. and Trakhtenbrot proved [Tra50] that it is not r.e. A similar phenomenon holds for finitely representable models.

Proposition 3.2 Let $\mathcal{A}$ be an $\mathcal{L}$-structure such that the theory of $\mathcal{A}$ is decidable. Then the set of sentences in $\mathcal{L} \cup \sigma$ valid over $\mathcal{L}$-representable expansions with respect to $\mathcal{A}$ and σ is co-r.e.

Proof: We prove that the set of sentences in $\mathcal{L} \cup \sigma$ satisfiable over finitely representable models with respect to $\mathcal{A}$ and σ is r.e. Recall that $\mathcal{L}$ is assumed to be countable. The set of $\mathcal{L}$-representable $(\mathcal{A}, \sigma)$-instances can be effectively enumerated. It suffices to enumerate their representations, which are the formulas without quantifiers representing the relations in σ. Now, let $\mathcal{A}'$ be an expansion whose restriction to σ is as above, and assume that for each $i \in [1..n]$, $\psi_i(\bar{x}_i)$ is the finite representation of $R_i \in \sigma$. We have that

$$\mathcal{A}' \models \varphi \quad \text{iff} \quad \mathcal{A} \models \varphi|_{\psi_1}^{R_1} \cdots |_{\psi_n}^{R_n},$$

where $\phi|_{\psi}^{R}$ is the formula obtained from ϕ by replacing each occurrence of R by the formula ψ. Then the result follows from the decidability of the theory of $\mathcal{A}$. $\Box$

The theory of dense order without endpoints and the theory of real closed fields mentioned above are decidable [CK73]. Therefore, validity over finitely representable databases over $(\mathcal{Q}, \leq)$ or $(\mathcal{R}, \leq, +, \times, 0, 1)$ for instance, is co-r.e. We will see below that it is not r.e.

We next prove that in general the set of first-order sentences satisfiable over finitely representable models is not recursive. Like for the compactness, we state the result in two different contexts.

Theorem 3.3 Let $\mathcal{L}$ be a first-order language with $=$, and $\boldsymbol{\mathcal{N}}$ be the structure of the natural numbers with equality only. Then the set of sentences in $\mathcal{L} \cup \sigma$ valid over $\{=\}$-representable models with respect to $\boldsymbol{\mathcal{N}}$ and σ is not r.e.

Proof: The proof is based on Vaught's result [Vau60]. Let $V_{\text{r.e.}}$ (resp. V_{fin}) be the set of sentences true in all recursively enumerable (resp. finite) models. Vaught showed that for each set of sentences V, if $V_{\text{r.e.}} \subseteq V \subseteq V_{\text{fin}}$, then V is not r.e. Now, let $V^{\sigma}_{\mathcal{N}}$ be the set of sentences valid over finitely representable models with respect to $\mathcal{N}$ and σ. It is clear that $V_{\text{r.e.}} \subseteq V^{\sigma}_{\mathcal{N}} \subseteq V_{\text{fin}}$, and therefore $V^{\sigma}_{\mathcal{N}}$ is not r.e. $\Box$

Next, we consider languages with both $=$ and $\leq$, where $\leq$ satisfies the axioms of the theory of dense order without endpoints, and with possibly a countable set of constants and possibly $+$ and $\times$ with their usual interpretation (i.e., satisfying the axioms of the theory of real closed fields).

Theorem 3.4 Let $\mathcal{L}$ be a first-order language described as above. Assume that $\mathcal{A}$ is an $\mathcal{L}$-structure such that the theory of $\mathcal{A}$ admits a quantifier elimination procedure. Then the set of sentences in $\mathcal{L} \cup \sigma$ satisfiable over $\mathcal{L}$-representable models with respect to $\mathcal{A}$ and σ is not recursive.

Proof: The proof is done by a reduction from the satisfiability over finite structures. Let $\sigma = \{R\}$, where R is an n-ary relation symbol, $\mathcal{L}' = \{=\}$, and A be the domain of $\mathcal{A}$. Let ϕ be a sentence in $\mathcal{L}' \cup \sigma$. We construct a sentence ψ in $\mathcal{L} \cup \sigma$ such that ϕ has a finite model ($\mathcal{L}' \cup \sigma$-structure) iff ψ has an $\mathcal{L}$-representable model ($\mathcal{L} \cup \sigma$-structure). Let

$$\phi_i(x) = \exists x_1 \cdots x_{i-1} x_{i+1} \cdots x_n \; R(x_1, ..., x_{i-1}, x, x_{i+1}, ..., x_n)$$

be the projection of R on the i-th column. Then we define $\psi = \phi \wedge \bigwedge_{i=1}^{n} \alpha_i$, where ($x < y$ denotes $x \leq y \wedge x \neq y$)

$$\alpha_i = \forall xy \big((\phi_i(x) \wedge \phi_i(y) \wedge x < y) \rightarrow \exists z \, (x < z < y \wedge \neg \phi_i(z)) \big).$$

Suppose that ϕ admits a finite model $\mathcal{M} = (M, R^{\mathcal{M}})$ of cardinality ℓ. Let h be a 1-1 onto mapping from M to a finite subset of A. Let $\mathcal{A}'$ be an expansion of $\mathcal{A}$ to σ such that:

$$\mathcal{A}' \models \forall x_1 \cdots x_n \big(R(x_1, ..., x_n) \leftrightarrow \bigvee_{(a_1,...,a_n) \in R^{\mathcal{M}}} (x_1 = h(a_1) \wedge \cdots \wedge x_n = h(a_n)) \big).$$

Then, $\mathcal{A}'$ is an $\{=\}$-representable, hence $\mathcal{L}$-representable, model of ψ. Indeed, since $\leq^{\mathcal{A}}$ is dense, α_i holds in $\mathcal{A}'$.

On the other hand, suppose that ψ has an $\mathcal{L}$-representable model $\mathcal{A}'$, which is an expansion of $\mathcal{A}$ to σ. We first prove that $R^{\mathcal{A}'}$ is finite. Since $R^{\mathcal{A}'}$ is an $\mathcal{L}$-representable relation, let ψ_R be the quantifier-free formula which defines $R^{\mathcal{A}'}$.

By assumption, the theory of $\mathcal{A}$ admits a quantifier elimination procedure. Therefore, $R_i = \{x \mid \phi_i|^{R}_{\psi_R}(x)\}$ is also $\mathcal{L}$-representable. Let $\psi_{R_i}(x)$ be the quantifier-free monadic formula defining R_i. By [Tar51], $\psi_{R_i}(x)$ defines a finite set of segments over A. On the other hand, $\mathcal{A}'$ satisfies α_i, and so for every pair of distinct elements $a, b \in A$ such that $a, b \in R_i$ and $a < b$, there is some $c \in A$ such that $a < c < b$ and $c \notin R_i$. It follows that $\psi_{R_i}(x)$ defines a finite set of elements. Hence, $R_i^{\mathcal{A}'}$ is finite, and so is $R^{\mathcal{A}'}$. Consider the $(\{=\} \cup \sigma)$-structure $\mathcal{M} = (M, R^{\mathcal{M}})$, where $M = \cup_{i=1}^{n} R_i$ and $R^{\mathcal{M}} = R^{\mathcal{A}'}$. $\mathcal{M}$ is a finite model of ϕ. $\Box$

Note that finiteness can be expressed in the very specific case of finitely representable databases in the context of a dense order. This is a drastic difference with classical logic. The previous result admits the following corollaries.

Corollary 3.5 Under the same assumptions as in Theorem 3.4, the set of sentences in $\mathcal{L} \cup \sigma$ valid over $\mathcal{L}$-representable models with respect to $\mathcal{A}$ and σ is co-r.e. but not r.e.

We can therefore conclude with the following general result.

Corollary 3.6 The Completeness Theorem does not hold for finitely representable models.

The next theorem shows yet another similarity with finite model theory. The proof is easy and in the same spirit as the one for finite models.

Theorem 3.7 Consider an $\mathcal{L}$-representable $(\mathcal{A}, \sigma)$-expansion $\mathcal{B}$. There exists a sentence $\sigma_{\mathcal{B}}$ in $\mathcal{L} \cup \sigma$ such that for each $\mathcal{L}$-representable $(\mathcal{A}, \sigma)$-expansion $\mathcal{B}'$, $\mathcal{B}'$ is isomorphic to $\mathcal{B}$ iff $\mathcal{B}' \models \sigma_{\mathcal{B}}$.

On the other hand, the theory of finitely representable models differs from finite model theory. Gaifman [Gai81] proved that first-order logic was local in a topological sense. This fails in presence of an order relation. Indeed, every neighborhood based on Gaifman's distance contains the whole domain. Techniques based on asymptotic probabilities of the truth of sentences also fail in the presence of order or arithmetic. Some results were obtained for very specific languages (see [Com88]). They do not carry over in the general case.

Finally, we consider some consequences of the previous theoretical results on database queries. Let φ and ψ be sentences in $\mathcal{L} \cup \sigma$ and $\mathcal{A}$ be an $\mathcal{L}$-structure. The query (formula) ψ is said to be *contained in φ with respect to $\mathcal{L}$-representable $(\mathcal{A}, \sigma)$-instances*, denoted by $\psi \sqsubseteq_{\mathcal{A},\sigma} \varphi$, if $K_\psi^{\mathcal{A}} \subseteq K_\varphi^{\mathcal{A}}$. Queries φ and ψ are *equivalent with respect to $\mathcal{L}$-representable $(\mathcal{A}, \sigma)$-instances*, denoted $\varphi \equiv_{\mathcal{A},\sigma} \psi$, if $\psi \sqsubseteq_{\mathcal{A},\sigma} \varphi$ and $\varphi \sqsubseteq_{\mathcal{A},\sigma} \psi$.

Theorem 3.8 Under the same assumptions as in Theorem 3.4, the following problems are undecidable: the satisfiability, the equivalence, the containment, and the $\{\leq\}$-invariance of queries.

In particular, Theorem 3.8 indicates that the above problems are undecidable in the first-order constraint query language of [KKR90] over the real field.

4 Expressive power

In this section, we consider two kinds of database instances (i) over a dense order $\boldsymbol{\mathcal{R}}_{\leq} = (\mathcal{R}, \leq)$ and (ii) over the ordered real closed field $\boldsymbol{\mathcal{R}} = (\mathcal{R}, \leq, +, \times, 0, 1)$. We study the first-order definability of queries in the two languages corresponding to the above structures: $\mathcal{L}_{\leq} = \{\leq\} \cup \mathcal{Q}$ and $\mathcal{L}_{\times} = \{\leq, +, \times, 0, 1\}$.

We shall use the following proof techniques for finitely representable databases: Ehrenfeucht-Fraïssé games, data complexity arguments, invariance of queries, and not finitely representable output.

$\boldsymbol{\mathcal{R}}_{\leq}$ is a model of the theory of dense order without endpoints, and $\boldsymbol{\mathcal{R}}$ is a model of the theory of real closed fields. These theories are decidable and admit quantifier elimination procedures [CK73]. Therefore, every formula in $\mathcal{L}_{\leq} \cup \sigma$ (resp. $\mathcal{L}_{\times} \cup \sigma$) expresses a query over $\mathcal{L}_{\leq}$-representable $(\boldsymbol{\mathcal{R}}_{\leq}, \sigma)$-instances (resp. $\mathcal{L}_{\times}$-representable $(\boldsymbol{\mathcal{R}}, \sigma)$-instances).

The *data complexity* of queries over $(\boldsymbol{\mathcal{R}}_{\leq}, \sigma)$-instances and $(\boldsymbol{\mathcal{R}}, \sigma)$-instances has been studied in [KKR90, KG94]. Queries in $\mathcal{L}_{\leq} \cup \sigma$ over $\mathcal{L}_{\leq}$-representable $(\boldsymbol{\mathcal{R}}_{\leq}, \sigma)$-instances are computable in AC^0 data complexity, and queries in $\mathcal{L}_{\times} \cup \sigma$ over $\mathcal{L}_{\times}$-representable $(\boldsymbol{\mathcal{R}}, \sigma)$-instances are computable in NC data complexity.

On the other hand, the expressive power of these languages has not been seriously investigated. In [KKR90], an exact characterization of $PTIME$ was obtained with inflationary Datalog with negation over dense order. To the best of our knowledge there are no other exact characterization of complexity classes. Moreover, it is open if numerous well known queries are definable over finitely representable databases.

We first consider *k-dimensional region connectivity* (cf Example 2.1 where $k = 2$). Assume that $\sigma = \{R\}$, where R is a k-ary relation symbol. Intuitively, R defines some area (infinite set of points) in $\mathcal{R}^k$. Recall that R defines a connected region if every pair of points in R can be linked by a continuous line entirely in R.

Theorem 4.1 Let $k \geq 2$. The k-dimensional region connectivity query over $\mathcal{L}_{\leq}$-representable $(\boldsymbol{\mathcal{R}}_{\leq}, \sigma)$-instances is not definable in $\mathcal{L}_{\leq}$.

Theorem 4.1 can be proved using two different arguments, namely Ehrenfeucht-Fraïssé games, and a complexity theoretic argument. We illustrate both proof techniques below.

The first proof uses Ehrenfeucht-Fraïssé games [Fra54, Ehr61]. Let $\mathcal{A}$ and $\mathcal{B}$ be two $\mathcal{L}$-structures with universes A and B. Let r be a positive integer. The *game of length r associated with $\mathcal{A}$ and $\mathcal{B}$* is played by two players, I (the spoiler), and II (the duplicator), making r moves each. Player I starts by picking an element in A or B and player II picks an element in the opposite structure. This is repeated r times. At each move, player I has the choice of the structure, and player II must respond in the opposite structure. Let a_i (b_i) be the i^{th} element picked in A (B). Player II wins the round $\{(a_1, b_1), ..., (a_r, b_r)\}$ iff the mapping $a_i \rightarrow b_i$ is an isomorphism of the substructures of $\mathcal{A}$ and $\mathcal{B}$ generated by $\{a_1, ..., a_r\}$ and $\{b_1, ..., b_r\}$, respectively, $\mathcal{A}/\{a_1, ..., a_r\}$ and $\mathcal{B}/\{b_1, ..., b_r\}$.

Player II *wins the game of length* r associated with $\mathcal{A}$ and $\mathcal{B}$ if he has a winning strategy, i.e., player II can always win any game of length r on $\mathcal{A}$ and $\mathcal{B}$, no matter how player I plays. This is denoted by $\mathcal{A} \equiv_r \mathcal{B}$. Note that the relation $\equiv_r$ is an equivalence relation on structures.

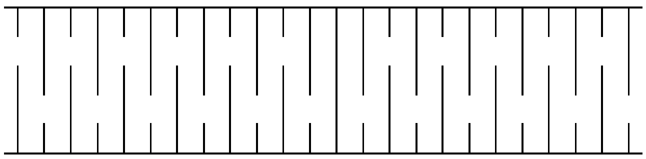
Intuitively, the equivalence $\mathcal{A} \equiv_r \mathcal{B}$ says that $\mathcal{A}$ and $\mathcal{B}$ cannot be distinguished by looking at just r constants at a time in the two structures. The main result concerning Ehrenfeucht-Fraïssé games states that the ability to distinguish among structures using games of length r is equivalent to the ability to distinguish among structures using some first-order sentence of quantifier depth r. Ehrenfeucht-Fraïssé's general result implies in particular the following proposition.

Proposition 4.2 Let ϕ be a sentence in $\mathcal{L} \cup \sigma$ with quantifier depth r, and let $\mathcal{A}'$ and $\mathcal{A}''$ be two $(\mathcal{L} \cup \sigma)$-structures. If $\mathcal{A}' \models \phi$ and $\mathcal{A}' \equiv_r A''$, then $\mathcal{A}'' \models \phi$.

Since $\mathcal{L}$-representable $(\mathcal{A}, \sigma)$-instances are $(\mathcal{L} \cup \sigma)$-structures, Proposition 4.2 can be applied in the context of finitely representable databases. This provides a technique for proving that a query K is not definable by any first-order sentence. Indeed, it is sufficient to exhibit, for each r, two $\mathcal{L}$-representable $(\mathcal{A}, \sigma)$-instances $\mathcal{A}_r$ and $\mathcal{B}_r$ such that $\mathcal{A}_r$ belongs to K, $\mathcal{B}_r$ does not, and $\mathcal{A}_r \equiv_r \mathcal{B}_r$.

We next prove Theorem 4.1 for a binary relation R. The players play on two $\mathcal{L}_{\leq}$-representable $(\mathcal{R}_{\leq}, \sigma)$-instances. They choose alternatively, real numbers. These numbers are the coordinates of points in the relation R. We assume that they have both r pebbles that they place on coordinates.

Proof of Theorem 4.1: (Sketch) It is sufficient to show the case when $k = 2$. We describe the main ideas of the two instances on which the game is played and the strategy of the duplicator player. The two instances, say $\mathcal{A}_r$ and $\mathcal{B}_r$, are built with segments. They look like graphs which differ only by one edge (i.e. segment). Each of them looks like two imbricated combs sharing one tooth in $\mathcal{A}_r$ and none in $\mathcal{B}_r$ (see the figure). The teeth are actually not straight but complex zigzag. The number of teeth and the number of angles in a tooth is exponential in the number of steps of the game. Moreover, the "distinguished points", such as the root of a teeth, or its extremity, have no coordinates in common. In particular, the root and the extremity of a tooth are not on a vertical line, and teeth all have different length. (We assume moreover that no rational coordinates can be defined explicitly. That corresponds to the case, where a sentence contains some rational constants. Otherwise, it suffices to define the structures away from these numbers.)



A skeleton of the instance $\mathcal{A}_r$

For simplicity reasons, we explain the strategy of the duplicator as if the two players were picking points in $\mathcal{R}^2$ instead of $\mathcal{R}$. In fact, they are choosing coordinates, and do not even necessarily alternate first and second coordinates. Nevertheless, this simplifies the description of the strategy, and the way this strategy can be implemented on the real game follows. The winning strategy of the duplicator goes as follows. Problems arise only when the spoiler plays on the tooth on $\mathcal{A}_r$ linking the two combs. If the spoiler plays on its two roots, the duplicator, plays on the roots of two opposite teeth in $\mathcal{B}_r$. Now, if the spoiler plays on their extremities, the duplicator plays on the extremities of other teeth in the neighborhood in $\mathcal{A}_r$. Since the number of teeth and the number of angles in a tooth are exponential in r, this can go on for r moves. The duplicator plays on points such that the coordinates of the marked points in the two instances satisfy identically $=$ and $\leq$. What makes this strategy possible is that the distances do not have to be respected. This is due to the fact that the language is restricted to the predicates $=$ and $\leq$. $\Box$

The technique of Ehrenfeucht-Fraïssé games, however, cannot easily be extended to $\mathcal{L}_{\times}$ because of the arithmetic operations. Indeed, any two instances such as $\mathcal{A}_r$ and $\mathcal{B}_r$ in the previous proof can be distinguished using a sentence in $\mathcal{L}_{\times}$. Any algebraic number can be defined in $\mathcal{L}_{\times}$. So, the following can be defined: there is a point (a, a') in $R^{\mathcal{A}_r}$ not in $R^{\mathcal{B}_r}$, where a and a' are algebraic.

We conjecture that k-dimensional region connectivity is not expressible in $\mathcal{L}_\times$ for $k \geq 2$. More generally, an interesting and important open problem is to know if the transitive closure of a finite graph is expressible in $\mathcal{L}_\times$.

Ehrenfeucht-Fraïssé games can be used to prove other results for $\{\leq, =\}$. In particular, it can be seen that the *parity* of the number (if finite) of points, segments, connected components, etc. cannot be expressed in $\{\leq, =\}$ nor can the transitive closure of finite graphs in this language (details in the full paper).

We now consider the data complexity approach. Non definability is proven by comparing the complexity upperbound of the language and the lowerbound of the problem being considered. The following lemma improves a result of [KKR90].

Lemma 4.3 [KG94] Each query in $\mathcal{L}_\leq$ over $\mathcal{L}_\leq$-representable $(\boldsymbol{\mathcal{R}}_\leq, \sigma)$-instances has data complexity AC^0, i.e., can be evaluated in AC^0 with respect to the size of the database.

We now present the complexity theoretic proof of the non definability of the region connectivity query in the case $k = 3$. Essentially, it is a reduction from the *parity* query that is known to be outside of AC^0 [FSS84, Ajt83]. The reduction is first-order [Imm87], and so in AC^0. Let $\{a_1, ..., a_n\}$ be the set of elements. For technical convenience and without loss of generality, we assume that for each $i \in [2..n]$, $a_{i-1} < a_i$. Now the relation $R(x, y, z)$ is obtained by adding the following elements. For each $i \in [1..n]$, we represent a_i by the point $(a_i, 0, 0)$ in R and for each $i \in [1..(n-2)]$, we construct (straight) line segments from $(a_i, 0, 0)$ via points $(a_i, 0, a_i)$, $(a_i, 1, a_i)$, $(a_{i+2}, 1, a_i)$, $(a_{i+2}, 0, a_i)$, and eventually to $(a_{i+2}, 0, 0)$. Finally, R also includes a set of lines starting from $(a_n, 0, 0)$ via points $(a_n, 1, 0)$, $(a_1, 1, 0)$, and then to $(a_1, 0, 0)$. Figure 1 shows some of the lines in R. It is easy to verify that R is region connected iff n is even. Hence, region connectivity in the 3-dimensional space is not definable. Note that this reduction does not naturally apply to the case when $k = 2$, since the lines linking odd nodes and even nodes would overlap. Furthermore, the reduction becomes non first order (nor in AC^0) if it lays the lines of even (or odd) nodes in the other side of the y axis.

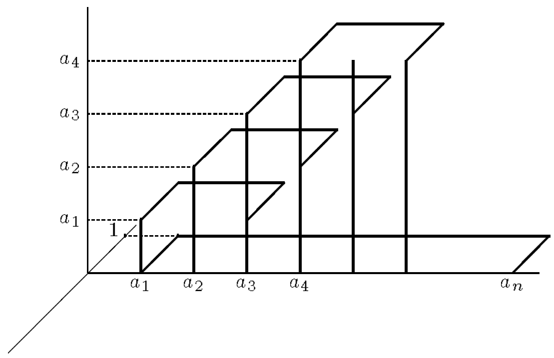


Figure 1: An Illustration of the Reduction

Lemma 4.3 also yields the following corollary by complexity arguments.

Corollary 4.4 The transitive closure query and the parity query over finite databases are not expressible in $\mathcal{L}_\leq$.

The following examples from geographical databases are proved using an invariance argument.

Example 4.1 Let $\sigma = \{R\}$ be a schema with a single binary relation. R defines a *grid* structure if it has a finite set of points and for each point (x, y) in R, there exist integers i_x, i_y such that $x = x_0 + i_x\Delta_x$ and $y = y_0 + i_y\Delta_y$ for some fixed $x_0, y_0, \Delta_x, \Delta_y$. It is easy to see that the grid query (checking if an input is a grid) is expressible in $\mathcal{L}_\times$. (The finiteness can be expressed by stating that each point is "isolated".) However, since the grid query is not $\{+, \times\}$-invariant, it is not expressible in $\{\leq, =\}$. $\Box$

Now, consider $\sigma = \{R\}$ as in the above example. The relation R defines a *convex* if for each pair of points in R, the straight line between them is entirely in R. Obviously, the *convexity* query is expressible in $\mathcal{L}_\times$. We now study if it is expressible in $\mathcal{L}_\leq$. We show that the query is not expressible in general over $\mathcal{L}_\times$-representable $(\boldsymbol{\mathcal{R}}, \sigma)$-instances, but it is expressible over $\mathcal{L}_\leq$-representable $(\boldsymbol{\mathcal{R}}_\leq, \sigma)$-instances. We first consider the later case.

Consider an $\mathcal{L}_\leq$-representable $(\boldsymbol{\mathcal{R}}_\leq, \sigma)$-instance $\mathcal{I}$. $R^{\mathcal{I}}$ is equivalent to a formula $\bigvee_i \bigwedge_j \psi_{i,j}$, where $\psi_{i,j}$ is either $s = t$, $s \leq t$, or their negations, and s, t are variables or constants in $\mathcal{Q}$. Hence each atomic formula represents either a line parallel to

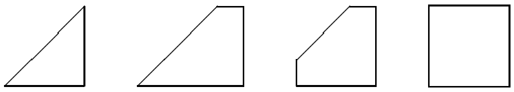


Figure 2: Some Possible Convex Regions

the horizontal or vertical axis or the line $x = y$, or the complement, or a half plane defined by a line of the above three types. It is observed that for $R^{\mathcal{I}}$ to be a convex, the region it defines must be of one of a fixed set of shapes. Several possible shapes are shown in Figure 2. Now the construction of the sentence in $\mathcal{L}_{\leq}$ expressing the convexity query is quite straightforward.

Note also that convexity of $R^{\mathcal{I}}$ is preserved by each $\{\leq\}$-preserving (with respect to $\mathcal{I}$) $\{\leq\}$-automorphism of $\mathcal{R}$. We next turn to $(\boldsymbol{\mathcal{R}}, \sigma)$-instances.

Proposition 4.5 The *convexity query* is not expressible in $\mathcal{L}_{\leq}$ for $(\boldsymbol{\mathcal{R}}, \sigma)$-instances.

Proof: Let $\sigma = \{R\}$ contain a binary relation symbol R. Suppose that there is a sentence φ in $\mathcal{L}_{\leq} \cup \sigma$ that expresses the convexity query for $(\boldsymbol{\mathcal{R}}, \sigma)$-instances. Let $a = \max\{c \mid c$ is a constant appearing in $\varphi\}$. Without loss of generality, we assume that $a \geq 1$. We construct an instance $\mathcal{I}$ such that $R^{\mathcal{I}}(x, y) \equiv ((y + a = x) \wedge (a \leq x \leq a + 2))$. Obviously, $R^{\mathcal{I}}$ consists of a single line segment which by definition is a convex. Consider the function f defined as:

$$f(z) = \begin{cases} z & \text{if } z \leq a+1 \\ \frac{z+1-a}{2} + a & \text{if } z \geq a+1 \end{cases}$$

Note that $f(R^{\mathcal{I}})$ contains two connected line segments and is not a convex. On the other hand, f is an $\{\leq\}$-automorphism, i.e., it preserves $\leq$. Using an inductive argument similar to that in Proposition 2.2, one can verify that for the expansions $\mathcal{B}, \mathcal{B}_f$ of $\boldsymbol{\mathcal{R}}$ with $R^{\mathcal{I}}$ and with $f(R^{\mathcal{I}})$ (respectively), either $\mathcal{B} \models \varphi$ and $\mathcal{B}_f \models \varphi$, or $\mathcal{B} \not\models \varphi$ and $\mathcal{B}_f \not\models \varphi$. This contradicts the assumption that φ expresses convexity. $\Box$

Next we consider several non-boolean queries over finite graphs. It is interesting, to investigate the expressive power of the first-order calculus, in the presence of arithmetic.

Let $\sigma = \{S\}$, where S is a binary relation (defining a finite set of points in $\mathcal{R}^2$) and $\tau = \{T\}$ where T is of arity 4. The *Euclidian spanning tree* query returns the relation T which defines the Euclidian spanning tree of the points in S. Since sentences in $\mathcal{L}_{\leq}$ cannot "check" distances, it can be shown that this query is not expressible in $\mathcal{L}_{\leq}$ over $(\boldsymbol{\mathcal{R}}, \sigma)$-instances, nor over $(\mathcal{R}, \sigma)$-instances, by an argument similar to Proposition 4.5.

The *Voronoi diagram* query returns the relation V of arity 4, which associates to each point in S, its Voronoi region. Obviously, the last query may generate an $(\boldsymbol{\mathcal{R}}, \tau)$-instance, when the input is an $(\boldsymbol{\mathcal{R}}_{\leq}, \sigma)$-instance. The query is not closed under $\boldsymbol{\mathcal{R}}_{\leq}$-instances. However, formulas in $\mathcal{L}_{\leq}$ only output $(\boldsymbol{\mathcal{R}}_{\leq}, \tau)$-instances on input $(\boldsymbol{\mathcal{R}}_{\leq}, \sigma)$-instances. In computational geometry, there are many similar queries where the output will in general be an $\boldsymbol{\mathcal{R}}$-instance, such as the *Convex hull*, and the *Delaunay triangle*, etc. None of these queries are expressible in $\mathcal{L}_{\leq}$ for $(\boldsymbol{\mathcal{R}}_{\leq}, \sigma)$-instances.

We now consider the language $\mathcal{L}_{\times}$. It is easy to see that the convex hull, Voronoi diagram, and Delaunay triangle queries are expressible in $\mathcal{L}_{\times}$. For example, to express the Voronoi diagram query, suppose S holds a finite set of points. The following formula constructs a 4-ary relation $V(x, y, u, v)$ such that the point (x, y) is in S and is the closest point in S to a point (u, v).

$$\begin{aligned} V(x, y, u, v) \leftrightarrow (S(x, y) \wedge \forall x' \forall y' S(x', y') \rightarrow \\ \mathit{dist}(x, y, u, v) \leq \mathit{dist}(x', y', u, v)), \end{aligned}$$

where *dist* is a formula which computes the distance between two points.

Now consider the Euclidian travel salesman query. It was shown to be *NP*-complete [Yao90]. Since each query in $\mathcal{L}_{\times}$ can be evaluated in *NC*, it follows that it is not expressible in $\mathcal{L}_{\times}$ unless *NC*=*NP*.

Acknowledgments

The authors wish to thanks Richard Lassaigne, Victor Vianu and the referees for useful comments.

References

[Ajt83] M. Ajtai. Σ_1^1 formulae on finite structures. *Ann. of Pure and Applied Logic*, 24:1–48, 1983.

[AHV92] S. Abiteboul, R. Hull, and V. Vianu. *Database Theory.* Draft, 1992.

[AU79] A. V. Aho and J. D. Ullman. Universality of data retrieval languages. In *Proc. Symp. on Princ. of Programming Languages*, 1979.

[CH80] A. Chandra and D. Harel. Computable Queries for Relational Data Bases. *Journal of Computer and System Sciences*, 21(2):156–178, Oct. 1980.

[CK73] C.C. Chang and H.J. Keisler. *Model Theory*, volume 73 of *Studies in Logic.* North-Holland, 1973.

[Cod70] E.F. Codd. A relational model of data for large shared data banks. *Communications of ACM*, 13:6:377–387, 1970.

[Com88] K. Compton. 0-1 laws in logic and combinatorics. In I. Rival, editor, *Proc. NATO Advanced Study Institute on Algorithms and Order*, pages 353–383. Reidel, Dordrecht, 1988.

[Ehr61] A. Ehrenfeucht. An application of games to the completeness problem for formalized theories. *Fund. Math*, 49, 1961.

[Fag76] R. Fagin. Probabilities on finite models. *Journal of Symbolic Logic*, 41(1):50–58, 1976.

[Fag93] R. Fagin. Finite model theory - a personal perspective. *Theoretical Computer Science*, 116:3–31, 1993.

[FSS84] M. Furst, J. B. Saxe, and M. Sipser. Parity, circuits, and the polynomial-time hierarchy. *Math. Syst. Theory*, 17:13–27, 1984.

[Fra54] R. Fraissé. Sur les classifications des systèmes de relations. *Publ. Sci. Univ Alger*, I:1, 1954.

[Gai81] H. Gaifman. On local and non local properties. In J. Stern, editor, *Proc. Herbrand Symposium Logic Colloquium*, pages 105–135. North Holland, 1981.

[HH93] T. Hirst and D. Harel. Completeness results of recursive data bases. In *Proc. 12th ACM Symp. on Principles of Database Systems*, pages 244–252, 1993.

[Hul86] R. Hull. Relative information capacity of simple relational schemata. *SIAM J. of Computing*, 15:3, 1986.

[Imm87] N. Immerman. Languages that capture complexity classes. *SIAM Journal on Computing*, 16(4):760–778, 1987.

[KG94] P. C. Kanellakis and D. Q. Goldin. Constraint programming and database query languages. In *Proc. 2nd Conference on Theoretical Aspects of Computer Software (TACS)*, April 1994. (To appear in LNCS Spring-Verlag volume).

[KKR90] P. Kanellakis, G Kuper, and P. Revesz. Constraint query languages. In *Proc. 9th ACM Symp. on Principles of Database Systems*, pages 299–313, Nashville, 1990.

[KRVV93] P. Kanellakis, S. Ramaswamy, D. Vengroff, and J. Vitter. Indexing for data models with constraints and classes. In *Proc. 12th ACM Symp. on Principles of Database Systems*, pages 233–243, 1993.

[Kup90] G.M. Kuper. On the expressive power of the relational calculus with arithmetic constraints. In *Proc. Int. Conf. on Database Theory*, pages 202–211, 1990.

[Kup93] G.M. Kuper. Aggregation in constraint databases. In *Proc. First Workshop on Principles and Practice of Constraint Programming*, 1993.

[Kup93a] G.M. Kuper. Personal communications, 1993.

[Rev90] P. Revesz. A closed form for datalog queries with integer order. In *Proc. Int. Conf. on Database Theory*, pages 187–201, 1990.

[Tar51] A. Tarski. *A Decision method for elementary algebra and geometry.* Univ. of California Press, Berkeley, California, 1951.

[Tra50] B. A. Trakhtenbrot. Impossibility of an algorithm for the decision problem in finite classes. *Doklady Akademii Nauk SSSR*, 70:569–572, 1950.

[Vau60] R. L. Vaught. Sentences true in all constructive models. *Journal of Symbolic Logic*, 25(1):39–53, March 1960.

[vdD82] L. van den Dries. Remarks on Tarski's problem concerning $(R, +, \times, exp)$. In *Logic Colloquium.* Elsevier, North-Holland, 1982.

[Via93] V. Vianu. Personal communications, 1993.

[Yao90] F.F. Yao. *Handbook of Theorical Computer Science*, volume A, chapter 7 Computational Geometry, pages 343–389. J. van Leeuwen, North Holland, 1990.