Metafinite Model Theory*

Erich Grädel† Yuri Gurevich‡

February 17, 1997

Abstract

Motivated by computer science challenges, we suggest to extend the approach and methods of finite model theory beyond finite structures. We study definability issues and their relation to complexity on *metafinite structures* which typically consist of *(i)* a primary part, which is a finite structure, *(ii)* a secondary part, which is a (usually infinite) structure that can be viewed as a structured domain of numerical objects, and *(iii)* a set of "weight" functions from the first part into the second.

We discuss model-theoretic properties of metafinite structures, present results on descriptive complexity, and sketch some potential applications.

Contents

*A previous version of this paper has been published in: D. Leivant (Ed.), *Logic and Computational Complexity*, Springer (1995), 313–366.

†Lehrgebiet Mathematische Grundlagen der Informatik, RWTH Aachen, D-52056 Aachen, Germany, graedel@informatik.rwth-aachen.de

‡EECS Department, University of Michigan, Ann Arbor, MI 48109-2122, USA, gurevich@umich.edu.
Partially supported by NSF grant CCR 92-04742 and ONR grant N00014-94-1-1182.

1 Finite models and beyond

Although questions involving finite structures have always been of interest to logicians, finite model theory has emerged as a separate research area only in the 1970's and early 1980's. Part of the motivation came from applications in computer science, in particular from databases and complexity theory. As was pointed out in [27], finite structures pose a nontrivial challenge for mathematical logic, in particular for model theory. Being closely related to the foundations of mathematics, classical logic is preoccupied with infinity. In fact most important classical results and techniques of mathematical logic (such as compactness, completeness, the usual preservation theorems) fail when only finite structures are considered. It was suggested in [27] that logicians should systematically develop a model theory of finite structures that is able to cope with the challenges from computer science.

1.1 Motivation

Many of the finite objects appearing in computer science refer at least implicitly to infinite structures. In particular, this is the case with *objects that consist of both structures and numbers*, like e.g. graphs with weights on the edges. Such objects arise in many areas of mathematics and computer science, e.g. in optimization theory, databases, complexity theory and combinatorics. Although a single such object may be representable by a finite structure, it is not always desirable to do so. The numbers appearing in it live in an infinite structured domain, e.g. the field of reals or the arithmetic of natural numbers, and the arithmetical operations that we want to perform on these numbers may take us out of any *a priori* fixed finite subdomain. Thus it is desirable to work directly on the infinite structure, but to adjust the logical languages in an appropriate way so that certain complications coming from the infinity of the structure are avoided.

Databases. To explain the challenge of going beyond finite models and integrating structures and numbers, we first look at database theory, a particularly important area for such an approach. We refer to the books [1, 58] and the survey article [41] for background on database theory.

The common practice of viewing (a state of) a relational database as a finite structure is not always adequate; we are not the first to say that (see Sect. 1.4 in this connection). Let us look a little closer at the relationship between databases and finite model theory. In fact, database theory doesn't start with identifying relational databases and finite relational structures. Informally, a relational database is a finite collection of relations, each of which is a finite subset $R \subseteq D_1 \times \cdots \times D_m$ of tuples in a cartesian product of domains D_i. The domains need not be finite, in fact it is often assumed that all domains are *countably infinite*. The *active domain* of the database is the set of those domain elements that appear in some relation. Since the relations are finite, so is the active domain. So actually, a database is a countably infinite structure all whose relations are finite. By considering the substructure induced by the active domain, a finite structure is obtained carrying all the relevant information. For many theoretical considerations one can forget at this point where the domain elements came from, and work with the finite structure instead.

However, in real databases some of the domains are not just plain sets, but themselves are (infinite) mathematical structures, e.g., the natural numbers with arithmetic. Traditionally the relations and functions structuring these domains are not considered as parts of the database; supposedly they are imposed "from outside". But of course, this additional structure of the domains is used in database applications. Commercial query languages like SQL have arithmetical operations and comparisons, as well as so-called *aggregate functions* like mean, sum, max, min that are applicable to the appropriate domains. In this case the restriction to the active domain is no longer convincing, since arithmetical operations may produce new numbers that were not previously stored in the database.

We thus believe that a more realistic logical approach to databases should be *systematically* developed, that does not adhere to the strict finiteness condition, but but nevertheless retains the essential achievements of finite model theory.

Discrete dynamic systems. Databases evolve in time and can be viewed as special discrete dynamic systems. Additional examples are ubiquitous in computer science: microprocessors, operating systems, compilers, programming languages, communication protocols. Discrete dynamic systems play an enormous rôle in computer science and engineering. The problem of formal specification of discrete dynamic system is very important and attracts much attention. In practice, the most popular approaches to the specification problem are operational approaches which formalize states of discrete dynamic systems in one form or another. For a logician, it is natural to formalize states as structures of first-order logic. This avenue has been pursued in the evolving algebra approach; it is quite practical and fruitful [29].

Since states are finite they can be formalized as finite structures. However, it turns out that often it is more convenient and practical to incorporate various background structures into states and deal with infinite states. This is a rule, rather than an exception, in the evolving algebra literature (see [11]). Here we restrict ourselves to one simple example.

Imagine that a state of interest includes a stack of some objects which may be popped or pushed during the transition to the next state. There are many ways to implement a stack. Respectively, there are many ways to represent a stack in a finite structure. But you may want to avoid excessive detailization, for example to make your verification proof simpler and cleaner. One solution is to have an auxiliary infinite universe of stacks with built-in pop and push operations and a nullary function that gives the stack of interest to us. The details of this simple example are explained in the EA Tutorial mentioned in [11]. More involved variations of the example appear in many places, in particular in Jim Huggins' correctness proof of the Kermit communication protocol (also referred to in [29]) where stacks are replaced by queues.

1.2 Metafinite structures

Logics with counting. There are logics, studied in the framework of finite model theory, that go some way towards integrating logic and arithmetic. These are the *logics with counting*, augmenting familiar logics like first-order logic or fixed-point logic with the ability to count the number of tuples in any definable relation. Syntactically this can be done by either *counting terms* or *counting quantifiers*.

The motivation for considering these logics comes from the observation that from the point of view of expressiveness, first-order logic (FO for brevity) has two main deficiencies: It has *no mechanism for recursion or unbounded iteration*, and it *cannot count*. There are several well-studied logics and database query languages that add recursion in one way or another to FO (or part of it), notably the various forms of fixed point logic, the query language Datalog and its extensions.

On ordered finite structures, some of these languages express precisely the queries that are computable in PTIME or other complexity classes. However, this is not the case for classes of arbitrary (not necessarily ordered) structures, and most of the known counterexamples involve counting. Thus, Immerman [36] proposed to add counting quantifiers to fixed point logic and asked whether this would suffice to capture PTIME. Although Cai, Fürer and Immerman [12] eventually answered this question negatively, fixed point logic with counting turned out to be an important logic, defining a natural level of expressiveness below PTIME, with a number of equivalent characterizations [24].

Logics with counting are two-sorted. With a one-sorted finite structure $\mathfrak{A}$ with universe A, one associates the two-sorted structure $\mathfrak{A}^* := (\mathfrak{A}, \mathfrak{N})$ where $\mathfrak{N} = (\{0, \dots, n\}, <))$ where $n = |A|$ and $<$ ist the usual ordering on $\{0, \dots, n\}$.

The two sorts are related by *counting terms* of the form $\#_x[\varphi]$ taking values in the second, numerical sort. The interpretation of $\#_x[\varphi]$ is the number of first-sort elements a that satisfy $\varphi(a)$. *(Inflationary) fixed point logic with counting* (FP + C) and *partial fixed point logic with counting* (PFP + C) are defined by closing first-order logic under counting terms and the usual FP + C (respectively PFP + C) rules for constructing formulas.

The predicates defined by fixed point operators may be mixed, i.e. range over both sorts. We refer to [24, 38, 48, 49] and to Sect. 4.3 and 5.3 below for more background and results on fixed point logics with counting.

It should be noted, that although the second, numerical sort is of rather restricted form — just a linear ordering — this suffices to define any polynomial-time computable numerical function in fixed point logic. Thus it makes no difference if the numerical sort has additional relations and functions, e.g. modular addition and multiplication, as long as these are polynomial-time computable.

Here we will consider similar two-sorted structures with the following essential differences:

- The numerical sort need not be finite.

- The structures may contain functions from the first to the second sort.
- We consider more general operations than counting.

Metafinite structures. Meeting the challenge to extend the approach and methods of finite model theory beyond finite models and integrating structures and numbers, we propose here a more general class of structures, which we call *metafinite structures*, and a number of logics to reason about them. Typical metafinite structures consist of *(i)* a primary part, which is a finite structure, *(ii)* a secondary part, which may be finite or infinite, and *(iii)* a set of "weight" functions from the first part into the second. Here is an example: a graph, the set of natural numbers with the usual arithmetical operations, and a weight function from the vertices (or the edges) of the graph to the natural numbers.

By itself, the notion of metafinite structures may seem to be an old hat. Indeed, they are just a special kind of two-sorted structures. The novelty of our approach is not so much in the structures themselves but rather in the logics for such structures, which access the primary and the secondary part in different ways.

The term "metafinite structure" is loose; in most cases, in this paper, the secondary part will be an infinite numerical domain, so the structures will be in fact perfectly infinite. The term "metafinite" reflects our intention to apply the approach and methods of finite model theory to these structures. In fact the infinity that we seek is very modest. It should not manifest itself too obtrusively, deviating our attention to phenomena that are pertinent to infinite structures only. Therefore our logics of metafinite structures — appropriate modifications of the usual logics of interest in finite model theory, such as first-order logic, fixed point logics or $L^{\omega}_{\infty\omega}$ — access the infinite part only in a limited way, for instance without variables (and therefore without quantifiers) over the secondary part. An important feature of these logics is that they contain, besides formulae and terms in the usual sense, a calculus of functions from the primary to the secondary part, which we call *weights*.

Encoding problems. One may object that a weighted structure, which consists of a finite structure and a collection of numbers, can be represented as a pure finite structure or a binary string. This is true, but not always satisfactory.

To encode a graph with weights on edges by a unweighted graph one can for instance, replace every edge (u, v) of weight m by m distinct nodes, each of them connected to u and v but to no other nodes. While the graph obtained in this way contains all information about the original weighted graph, it is very inconvenient to perform arithmetical computations on the encoded weights.

On the other side, encoding a structure (with or without weights) as a binary string requires that we order the structure and thus forces us to deal with presentations of structures rather than the structures themselves, which contradicts the spirit of the relational database approach as well as the spirit of finite model theory.

1.3 Potential applications

We have mentioned databases and discrete dynamic systems as motivations for metafinite model theory. There are numerous other areas where this approach may be useful. We intended also to write a section on applications of metafinite model theory but this has to be deferred to a later paper. Instead we mention a few potential application areas here.

Optimization theory. Many important optimization problems are NP-hard and thus cannot be efficiently solved, unless P = NP. One way to cope with such problems is to design *approximation algorithms* which do not necessarily find optimal solutions, but at least approximate ones. A typical requirement is that the cost of the approximate solution is within a constant factor of the optimal one (see e.g. [50] and the references given there). In fact many optimization problems admit efficient approximation algorithms, whereas for others it has been shown that finding approximate solutions is NP-hard as well.

Papadimitriou and Yannakakis [51] set forth a new, logical approach for studying the approximation properties of optimization problems. Exploiting Fagin's logical characterization of NP by means of existential second-order logic, they introduced two syntactically defined classes of maximization problems, MAX SNP and MAX NP, and proved that all problems in these classes admit efficient approximation algorithms. The work of Papadimitriou and Yannakakis, together with other developments in complexity theory, led to spectacular non-approximability results. In particular, the characterization of NP in terms of *probabilistically checkable proofs*, obtained by Arora et. al. [5], implies that no MAX SNP-hard problem can have a polynomial-time approximation scheme, unless P = NP.

Many practical optimization problems take inputs which are structures with weights, e.g. graphs with one or more weight functions assigning numbers to vertices or edges. Important examples are the TRAVELLING SALESMAN PROBLEM, MAX FLOW/MIN CUT, most scheduling problems, and so on (see [16] for additional examples).

As mentioned already in [51], the result of Papadimitriou and Yannakakis can be extended to problems with weights. However, the weighted versions of MAX SNP and MAX NP, as defined in [51], use the weights only in a rather limited way. We claim that metafinite structures provide the right framework to extend this approach to a more general definability theory of optimization problems with weights. This claim has been substantiated by recent work of A. Malmström [45] who used the approach of metafinite model theory to establish connections between the logical presentation and the approximation properties of optimization problems with weights. In particular, Malmström exhibits a syntactically defined class of optimization problems (with weights in $\mathbb{N}$) that admit fully polynomial-time approximation schemes.

Numerical invariants of structures. In many branches of mathematics, functions that assign numerical parameters to mathematical structures play an important rôle. For instance, a large part of graph theory is devoted to the study of numerical invariants of graphs, such as genus, chromatic number, clique number, diameter, girth, etc. Metafinite model theory provides a framework for studying definability issues of numerical invariants and relating them, for instance, to computational complexity.

Fault-tolerance of queries. Suppose we have a relational database where every entry has some probability of being incorrect. What is the probability that the result of a given query is correct? What is the expected difference between the results on the observed and "actual" databases. Again, such questions involve finite structures together with numbers.

An *unreliable database* can be defined as a pair $(\mathfrak{A}, \mu)$ consisting of a finite structure $\mathfrak{A}$ and a probability function μ that assigns to each atomic or negated atomic fact a probability of being wrong. With $(\mathfrak{A}, \mu)$ we can associate a probability space of databases $\mathfrak{B}$ with probabilities $\nu(\mathfrak{B})$ to be understood as the probability that the 'actual' database is $\mathfrak{B}$.

Given a query Q for an unreliable database $(\mathfrak{A}, \mu)$, it is interesting to determine its *fault-*

tolerance. For a Boolean query, the fault-tolerance is just the probability that the evaluation against the observed database $\mathfrak{A}$ gives the correct answer for the actual database $\mathfrak{B}$. For queries of positive arity, the fault-tolerance is defined in terms of the expected Hamming distance of $Q(\mathfrak{A})$ and $Q(\mathfrak{B})$, i.e. the expected number of tuples that distinguish between $Q(\mathfrak{A})$ and $Q(\mathfrak{B})$. In Sect. 3.3 we will show how to address these questions in the framework of metafinite model theory. Note that we can also consider unreliable metafinite databases. This gives examples where the secondary part has itself several sorts, namely one or more sorts for the numbers appearing in the database, and one sort over the real interval $[0, 1]$ for the error probabilities.

Computations over the real numbers. Blum, Shub, and Smale [10] introduced a model for computations over the real numbers (and other mathematical structures as well) which is now usually called a BSS machine. It is essentially a random access machine, with the important difference that real numbers are treated as basic entities and that arithmetic operations on the reals are performed in a single step, independently of the magnitude or complexity of the numbers involved. Many basic concepts and fundamental results of computability and complexity theory reappear in the BSS model: the existence of universal machines, the classes $\mathrm{P}_{\mathbb{R}}$ and $\mathrm{NP}_{\mathbb{R}}$ (real analogues of P and NP) and the existence of $\mathrm{NP}_{\mathbb{R}}$-complete problems. An example of an $\mathrm{NP}_{\mathbb{R}}$-complete problem is the question whether a given multivariate polynomial of degree four has a real root.

In finite model theory there exist numerous results relating computational complexity and logical definability on finite structures. The subarea investigating such questions is sometimes called *descriptive complexity theory*. The question arises whether similar results can be obtained for complexity over the reals. The main problem for characterising complexity over $\mathbb{R}$ in a model-theoretic setting is to define the right class of structures that permit a clear separation between the finite, discrete aspects of the problems and computations (like indices of tuples, time, indices of registers, the finite control of the machines) on one side and the arithmetic of real numbers on the other side.

It has been shown by Grädel and Meer [23] that this can be achieved by $\mathbb{R}$-structures, a special case of metafinite models, with the ordered field of reals as secondary part. $\mathbb{R}$-structures admit a number of results relating expressibility and complexity that parallel those of descriptive complexity theory in the classical case. In particular, Grädel and Meer established analogues to Fagin's logical characterization of NP in terms generalized spectra [20], and to the Immerman-Vardi Theorem, that fixed point logic captures polynomial time on ordered structures [35, 60]. We will explain some of these results in Sect. 4.

1.4 Related approaches

In database theory there have been a number of proposals for going beyond the strict finiteness condition and taking care of infinite data. In part this was motivated by new areas of application, such as geographical databases, that involve spatial data. We mention a few (by no means all) of the relevant papers.

The study of infinite *recursive structures* has a long tradition in mathematical logic, by the work of Malcev, Nerode, Rabin, Vaught and their scientific descendents. Recently there have been some papers on recursive structures that study questions related to finite model theory. Hirst and Harel [31] investigated *recursive databases*, given by a finite set of recursive relations over the natural numbers. They studied the notion of a computable query in this context and exhibited complete languages for two specific classes of recursive databases. On the class of all recursive databases, quantifier-free first-order logic suffices to define all computable queries, whereas a variant of QL – the complete language from [13] for the classical relational model — is complete on highly symmetric recursive databases. In another paper Hirst and Harel studied finite model theory issues, such as 0-1 laws and descriptive complexity, in the context of recursive structures [32]. Grädel and Malmström [22] discuss resource bounded measures on recursive structure and prove 0-1 laws. Stolboushkin [55] shows that important properties of first-order logic (compactness, completeness, preservation theorems) fail on the class of recursive structures. This work is related to ours by the motivation to extend the questions and methods of finite model theory to classes of infinite structures. However, metafinite model theory is radically different from recursive model theory.

Kanellakis, Kuper and Revesz [42] considered databases that are given by semi-algebraic constraints over the real (or rational) numbers. This model can handle spatial data and geometric queries in a very nice and convincing way. Classical relational query languages can be extended with mathematical theories that admit quantifier elimination, such as the theory of real closed fields, to provide a generalized notion of query language, called constraint query languages. Complexity issues for such query languages are addressed in [42], and it has been shown that although the decision problem of the underlying mathematical theory may have exponential complexity, the resulting constraint query languages admit efficient evaluation algorithms. In this context we refer to [8, 9, 26, 56] for further model theoretic results on finitely representable databases.

Kabanza, Stevenne and Wolper [40] present an extension of the relational database model for reasoning abount infinite temporal data. In this model, time is represented by a second sort over the integers and generalized relations are defined by linear constraints, i.e. in Presburger arithmetic. It is proved that first-order queries over such databases can be evaluated in polynomial time.

A proposal that is by far the closest to our approach appears in the penultimate section of the seminal paper of Chandra and Harel [13], the same paper that also laid much of the foundation for the theory of computable queries in the classical, relational model. In that section, Chandra and Harel define the notion of an *extended database*. For a finite domain D and a countable infinite domain F, an extended database is a finite collection of finite mixed relations of the form $R \subseteq D^k \times F^\ell$ and functions of the form $w : D^k \to F$. Moreover F is "intended to include interpreted features such as numbers, strings (if needed), etc.". In our terminology, an extended database is a metafinite structure with mixed relations. Chandra and Harel define the notion of an extended database query and show that their language QL can be generalized to a complete query language EQL that expresses precisely the extended computable queries. The internal structure of the secondary part F is not really used, except for the assumption that F is effectively enumerable.

As far as we know, this proposal of Chandra and Harel has not been further pursued in database theory, in sharp contrast to the ideas developed in the rest of their paper.

However, it should be noted that practical query languages, like SQL, have operations for computing the maximum, the average, etc. for a given set of numbers and thus they deal, in fact, with metafinite structures.

2 Metafinite structures

2.1 Basic definitions

In the following, German letters $\mathfrak{A}, \mathfrak{B}, \ldots, \mathfrak{R}, \ldots$, stand for finite or infinite structures; their universes are denoted by corresponding Latin letter $A, B, \ldots, R, \ldots$.

There are many variations of metafinite structures. We define here three basic notions:

- Simple metafinite structures.
- Metafinite structures with multiset operations.
- Metafinite algebras.

Metafinite structures with multiset operations are the most general of these notions, and we will refer to them just as metafinite structures. However, to simplify the exposition we start with the simple variant.

Definition 2.1. A *simple metafinite structure* is a triple $\mathfrak{D} = (\mathfrak{A}, \mathfrak{R}, W)$ consisting of

(i) a finite structure $\mathfrak{A}$, called the primary part of $\mathfrak{D}$;

(ii) a finite or infinite structure $\mathfrak{R}$, called the secondary (or numerical) part of $\mathfrak{D}$.[1] We always assume that $\mathfrak{R}$ contains two distinguished elements 0 and 1 (or TRUE and FALSE);

(iii) a finite set W of functions $w : A^k \to R$;

The *vocabulary* of $\mathfrak{D}$ is the triple $\Upsilon(\mathfrak{D}) = (\Upsilon_a, \Upsilon_r, \Upsilon_w)$ where each component of $\Upsilon(\mathfrak{D})$ is the set of relation or function symbols in the corresponding component of $\mathfrak{D}$. (We always consider constants as functions of arity 0.) The two distinguished elements 0,1 of $\mathfrak{R}$ are named by constants of Υ_r.

In finite model theory, we are mostly interested in definability questions concerning *classes* of finite structures. Contrary to classical model theory, a single finite structure often is of lesser interest; for instance, it can be characterized up to isomorphism in first-order logic. Here our main interest are definability questions concerning *classes of metafinite structures with fixed secondary part.* We write $M_\Upsilon[\mathfrak{R}]$ for the class of metafinite structures of vocabulary Υ with secondary part $\mathfrak{R}$ and $\mathrm{Fin}(\Upsilon_a)$ for the class of finite structures with vocabulary Υ_a.

Metafinite structures with multiset operations. Multisets generalize sets in the sense that they allow multiple occurrences of elements. For instance, a function $f : A \to R$, defines a multiset $\mathrm{mult}(f) = \{\!\{f(a) : a \in A\}\!\}$ over R (the notation $\{\!\{\ldots\}\!\}$ indicates that we allow multiple occurrences of elements).

A multiset M over R can also be described by a function $m : R \to \mathbb{N}$ where $m(r)$ is the multiplicity of r in M. For any set R, let $\mathrm{fm}(R)$ denote the class of all finite multisets over R.

In some of the metafinite structures that we will consider, the secondary part $\mathfrak{R}$ is not just a (first-order) structure in the usual sense; it comes with a collection of *multiset*

[1] We denote the numerical part by $\mathfrak{R}$ for "ARithmetic".

operations, i.e. operations $\Gamma : \mathrm{fm}(R) \to R$, mapping finite multisets over R to elements of R. Natural examples on, say, the real numbers are addition, multiplication, counting, mean, maximum, minimum.

Definition 2.2. A *structure with multiset operations* is a pair $\mathfrak{R} = (\mathfrak{R}_0, Op)$ where $\mathfrak{R}_0$ is a first-order structure and Op is a set of operations $\Gamma : \mathrm{fm}(R) \to R$ (where R is the universe of $\mathfrak{R}_0$). The vocabulary Υ_r of $\mathfrak{R}$ consists of the vocabulary of $\mathfrak{R}_0$ together with the names of the operations in Op. A *metafinite structure with multiset operations* is a triple $\mathfrak{D} = (\mathfrak{A}, \mathfrak{R}, W)$ as in Definition 2.1 except that $\mathfrak{R}$ is a structure with multiset operations.

Let us give some motivation for this definition. The logics that we will consider contain formulae and terms. Terms may take values in both parts of a metafinite structures. While the rôle of terms over the primary part is rather limited, the terms taking values in the secondary part are called *weight terms* and are of crucial importance here.

A weight term $F(x_1, \ldots, x_k)$ defines, on a metafinite structure $\mathfrak{D} = (\mathfrak{A}, \mathfrak{R}, W)$, a function $F^{\mathfrak{D}} : A^k \to R$. The collection of values assumed by $F^{\mathfrak{D}}$ forms a *finite multiset*

$$\mathrm{mult}(F^{\mathfrak{D}}) = \{\!\{ F^{\mathfrak{D}}(\bar{a}) : \bar{a} \in A^k \}\!\}.$$

We want to have in our languages the expressive means to apply to weight terms natural operations like, say, summation to build the new weight $\sum_{\bar{a}} F^{\mathfrak{D}}(\bar{a})$. Algebraically, this means that we want to have operations mapping finite multisets over R to elements of R. These multiset operations will allow us to build new weight terms.

Remark. We consider metafinite structures with multiset operations as the default, and will usually refer to them just as metafinite structures.

Metafinite algebras. In principle we can always reduce the primary part of a metafinite structure $\mathfrak{D}$ to a naked set A by pushing all the data into the functions in W. Indeed, we can first replace every function $f : A^k \to A$ by a $(k+1)$-ary relation, and then encode every predicate $Q \subseteq A^k$ by its characteristic function $\chi_Q : A^k \to \{0,1\} \subseteq R$.

Definition 2.3. A *metafinite algebra* is a metafinite structure (with or without multiset operations) whose primary part is a plain set, i.e. $\Upsilon_a = \varnothing$. The elimination of Υ_a-symbols as just described, associates with every metafinite structure $\mathfrak{D}$ a metafinite algebra $\mathfrak{D}^a$, called the *algebraic form* of $\mathfrak{D}$.

As we will explain later, the passage to metafinite algebras permits a lean presentation of a logic as a pure calculus of terms. In many cases, this is convenient, in others it is not.

Other variations. There exist several other conceivable variations of metafinite structures that are worth exploring. For instance, instead of allowing only functions from the primary to the secondary part, we may admit *mixed relations* $P \subseteq A^k \times R^m$ or *mixed functions* $f : A^k \times R^m \to R$. Mixed relations may be particularly interesting for database applications; however, to allow for finite presentations of the databases some restrictions on the admissible relations have to be imposed. A natural restriction is that mixed relations be finite and that mixed functions map all but finitely many elements to 0. But there are other possibilities of finite presentations, e.g., that the relations are recursive [31] or given by semi-algebraic constraints [42, 26].

We won't consider metafinite structures with mixed relations in this paper. However, the design and investigation of query languages for metafinite databases of this kind is one of the promising directions for future research.

Another variation, important in particular for databases, is when the secondary part has *several infinite sorts*, e.g. one for the natural numbers, one for strings, one for real numbers, and so on. While this extension poses no principal difficulty, it often requires heavier notation and we won't consider such structures in this paper.

2.2 Arithmetical structures and $\mathbb{R}$-structures

Of particular interest to us are metafinite structures, whose secondary part is a structure $\mathfrak{N}$ over the natural numbers such that the following hold:

- As a minimum, $\mathfrak{N}$ has the constants 0,1, the functions $+, \cdot$, the ordering relation $<$ and the multiset operations $\max, \min, \sum, \prod$.
- All functions, relations and multiset operations of $\mathfrak{N}$ can be evaluated in polynomial time.

Let us make the second point more precise:

Definition 2.4. Let $\mathfrak{N}_p$ be the structure with the universe $\mathbb{N}$, with all polynomial-time computable functions $f : \mathbb{N}^k \rightarrow \mathbb{N}$ (for all finite arities k) and with all relations $R \subseteq \mathbb{N}^k$ (of arbitrary finite arity k) whose characteristic functions are polynomial-time computable. To define the class of PTIME computable operations on $\mathrm{fm}(\mathbb{N})$, we have to be a little more careful: we assume that multisets $M \in \mathrm{fm}(\mathbb{N})$ are represented by listing all elements, repeatedly if they occur more than once. Thus, if $\mathrm{mult}_M(n)$ is the multiplicity of n in M, the *cost* of M is $\|M\| := \sum_n \mathrm{mult}_M(n) \log n$. Now, $\mathrm{OpP}(\mathbb{N})$ denotes the set of all operations $\Gamma : \mathrm{fm}(\mathbb{N}) \rightarrow \mathbb{N}$ that are computable in polynomial time (with respect to this representation). *Polynomial-time arithmetic*, denoted PTA, is the pair $(\mathfrak{N}_p, \mathrm{OpP})$. A *PTA-structure* is a metafinite structure whose secondary part is PTA.

On the other hand, as the minimal variant for the secondary part the structure, we have $\mathfrak{N}_0 = (\mathbb{N}, 0, 1, +, \cdot, <, \max, \min, \sum, \prod)$.

Definition 2.5. An *arithmetical structure* is a metafinite structure with secondary part $\mathfrak{N}$ such that $\mathfrak{N}$ is an expansion of $\mathfrak{N}_0$ and a reduct of PTA. A *simple arithmetical structure* is obtained from an arithmetical structure by omitting the multiset operations.

Another interesting class are $\mathbb{R}$-structures, used by Grädel and Meer [23] for developing a descriptive complexity theory over the real numbers.

Definition 2.6. An $\mathbb{R}$-structure is a simple metafinite structure with secondary part

$$\mathfrak{R} = (\mathbb{R}, +, -, \cdot, /, \leq, (c_r)_{r \in \mathbb{R}}).$$

It is convenient to include subtraction and division as primitive operations and assume that every element $r \in \mathbb{R}$ is named by a constant c_r so that every rational function $g : \mathbb{R}^k \rightarrow \mathbb{R}$ can be written as a term (without quantifiers)[2].

[2] A rational function is a quotient of two polynomials.

In [23] a slightly different presentation has been used including also the sign function

$$\operatorname{sgn}(x) := \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$

as a basic function. Clearly, this function is efficiently computable, but is not a rational function. We don't need this function here, because we have chosen to include in our logics a characteristic function rule (see Definition 3.1) from which the sign function is easily definable.

2.3 Global functions, numerical invariants and their complexity

Let $\mathcal{K}$ be a class of metafinite structures with secondary part $\mathfrak{R}$.

Definition 2.7. A *global (weight) function* on $\mathcal{K}$ of arity k is a function F that assigns to every structure $\mathfrak{D} = (\mathfrak{A}, \mathfrak{R}, W) \in \mathcal{K}$ a (local) function $F^{\mathfrak{D}} : A^k \to R$ in such a way that isomorphisms between structures are preserved: for every isomorphism $h : \mathfrak{D} \to \mathfrak{D}'$ we have that for all $a_1, \ldots, a_k \in A$

$$hF^{\mathfrak{D}}(a_1, \ldots, a_k) = F^{\mathfrak{D}'}(ha_1, \ldots, ha_k).$$

In most cases, $\mathfrak{R}$ will be a "numerical" structure (e.g. the natural numbers with arithmetical operations, or the field of rational, real or complex numbers) which is rigid, thus the restriction to the secondary part of any isomorphism between structures of $\mathcal{K}$ is the identity on $\mathfrak{R}$. In such cases, we call a nullary global function — assigning to each isomorphism class of structures a numerical value — a *numerical invariant*.

There are many interesting examples of numerical invariants both in the case of structures without weights and in the case of structures with weights: the order of the automorphism group of the structure; in graph theory, the usual graph parameters like the chromatic number, clique number or genus; and in optimization theory, the cost of an optimal solution, e.g. the length of a shortest TSP tour.

Examples of global numerical functions of positive arity are the distance between vertices x, y of a given graph, the order of an element x of a given group (i.e. the cardinality of the cyclic subgroup generated by x), etc.

Our notion of global functions generalizes the notions of global functions and global relations in finite model theory and the notion of relational queries in database theory.

Thus questions concerning computability, complexity and expressibility of relational queries on finite structures can be viewed as special cases of the corresponding questions for global functions on classes of metafinite structures.

Complexity of global functions. The notion of complexity for global functions depends on a given computational model and the *cost* (or *size*) associated with the elements of the secondary part. For instance, if the secondary part consists of natural numbers or binary strings, then we have a natural notion of cost given by the number of bits. On the other side, if we study complexity over real numbers with respect to the Blum-Shub-Smale model, then we treat every element of $\mathbb{R}$ as a basic entity of cost one.

To obtain a flexible and general notion of the complexity of global functions, we associate with the secondary part $\mathfrak{R}$ a *cost function*

$$\|\ \| : R \to \mathbb{N}.$$

The cost of a weight function $w : A^k \to R$ is then defined as $\|w\| := \sum_{\bar{a} \in A^k} \|w(\bar{a})\|$. The cost of a metafinite algebra $\mathfrak{D} = (A, \mathfrak{R}, W)$ is $\|\mathfrak{D}\| = \sum_{w \in W} \|w\|$ and the cost of a metafinite structure can be defined as the cost of the associated metafinite algebra. Note that this cost is always finite, and that the secondary part — which is assumed to be fixed — is given for free.

Proviso. For arithmetical structures, we let $\|n\| = 1 + \lfloor \log n \rfloor$, i.e. the length of the binary representation of n (with the convention that $\log 0 = 0$). For $\mathbb{R}$-structures, our default is that $\|r\| = 1$ for all $r \in \mathbb{R}$, which reflects the use of $\mathbb{R}$-structures for capturing complexity classes with respect to the Blum-Shub-Smale model.

For a metafinite structure $\mathfrak{D} = (\mathfrak{A}, \mathfrak{R}, W)$ we write $|\mathfrak{D}|$ for the cardinality of the primary part $\mathfrak{A}$ and let

$$\max \mathfrak{D} := \max_{w \in W} \max_{\bar{a}} \|w(\bar{a})\|$$

be the cost of the maximal weight. Then $\|\mathfrak{D}\| \leq p(|\mathfrak{D}|, \max \mathfrak{D})$ for some polynomial $p(n, m)$ that depends only on the vocabulary of $\mathfrak{D}$. Since most of the popular complexity classes are invariant under polynomial increase of the relevant input parameters, it therefore makes sense to measure the complexity of a computation on a structure $\mathfrak{D}$ in terms of $|\mathfrak{D}|$ and $\max \mathfrak{D}$.

For instance, an algorithm M on a class $\mathcal{C}$ of metafinite structures runs in polynomial-time (respectively, logarithmic space) if, on every input $\mathfrak{D} \in \mathcal{K}$, the computation of M terminates in at most $q(|\mathfrak{D}|, \max \mathfrak{D})$ steps, for some polynomial q, (respectively, uses at most $O(\log |\mathfrak{D}| + \log \max \mathfrak{D})$ of work space).

More generally, we can define the following notion of complexity

Definition 2.8. Let $\mathcal{K}$ be a class of metafinite structures with secondary part $\mathfrak{R}$, and $\|\ \| : R \to \mathbb{N}$ a cost function. Let $\mathcal{M}$ be a computation model suitable for evaluating global functions on $\mathcal{K}$. A *resource measure* for $\mathcal{M}$ is a function T associating with every $\mathcal{M}$-algorithm M and every input x a number $T_M(x) \in \mathbb{N} \cup \{\infty\}$. We say that M evaluates the global F on $\mathcal{K}$ with resource bound $t(n, m)$ if, given any structure $\mathfrak{D} \in \mathcal{K}$, and any tuple $\bar{a}$ of length appropriate for F, M computes $F^{\mathfrak{D}}(\bar{a})$ in such a way that $T_M(\mathfrak{D}, \bar{a}) \leq t(|\mathfrak{D}|, \max \mathfrak{D})$.

3 Logics of metafinite structures.

Fix any logic L suitable for finite structures, e.g. first-order logic, fixed point logic or the infinitary logic $L^{\omega}_{\infty\omega}$. There are several ways to extend L to a logic of metafinite structures.

3.1 Simple languages.

The first such extension, let us call it L^* for the time being, is suitable for reasoning about simple metafinite structures. It is given by the following definition.

Definition 3.1. Let $\Upsilon = (\Upsilon_a, \Upsilon_r, \Upsilon_w)$ be a vocabulary of simple metafinite structures (so that Υ_r does not contain multiset operations). Fix a countable set $V = \{x_0, x_1, \dots, \}$ of variables. *These variables range over the primary part only; we don't use variables taking values in the secondary part.*

The language of $L^*(\Upsilon)$ contains the following expressions:

- Terms over the primary part, denoted by $t_1, t_2, \dots$, which are called *point terms*. On a metafinite structure $\mathfrak{D} = (\mathfrak{A}, \mathfrak{R}, W)$, a point term $t(x_1, \dots, x_k)$ defines a function $t^{\mathfrak{D}} : A^k \to A$.
- Terms over the secondary part, which are called *weight terms* and are denoted by $F, G, H, \dots$. On $\mathfrak{D}$, a weight term $F(x_1, \dots, x_k)$ defines a weight function $F^{\mathfrak{D}} : A^k \to R$.
- *Formulae.* On $\mathfrak{D}$, a formula $\varphi(x_1, \dots, x_k)$ defines a predicate over the primary part of $\mathfrak{D}$, namely $\varphi^{\mathfrak{D}} = \{\bar{a} : \mathfrak{D} \models \varphi(\bar{a})\} \subseteq A^k$.

The terms, weights and formulae of $L^*(\Upsilon)$ are defined inductively by the following rules:

(i) the set of point terms is the closure of the set of V of variables under applications of function symbols in Υ_a.

(ii) If $t_1, \dots, t_k$ are point terms and w is a k-ary function symbol of Υ_w, then the expression $w(t_1, \dots, t_k)$ is a weight term.

(iii) If $F_1, \dots, F_k$ are weight terms and g is a k-ary function symbol of Υ_r, then the expression $g(F_1, \dots, F_k)$ is a weight term. In particular, all *closed* terms (in the usual sense) over Υ_r are weight terms of $L^*(\Upsilon)$.

(iv) Atomic formulae are either equalities of point terms, or equalities of weight terms, or expressions $P(t_1, \dots, t_k)$ or $Q(F_1, \dots, F_k)$ where P and Q are k-ary predicate symbols in Υ_a and Υ_r, respectively.

(v) The set of formulae of L^* is closed under all rules of L for building formulae. However, note that all variables appearing in these formulae range over the primary part only.

(vi) *The characteristic function rule:* If φ is a formula of L^*, then $\chi[\varphi]$ is a weight term of L^*, with the same free variables as φ and the following semantics.

$$\chi[\varphi]^{\mathfrak{D}}(\bar{a}) := \begin{cases} 1 & \text{if } \mathfrak{D} \models \varphi(\bar{a}) \\ 0 & \text{otherwise.} \end{cases}$$

The *basic* terms are the point terms and the weight terms that can be built using only the rules *(i)* – *(iv)* and *(vi)*. Note that the set of basic terms depends only on Υ, not on L.

Remark. The characteristic function rule has been included for reasons of convenience, to make the set of weight terms more expressive. It is conceivable that in certain contexts, logics without this rule may be more natural.

3.2 Logics with multiset operations

We now turn to logics that make use of multiset operations. As described in Sect. 2.1, multiset operations can be used to define new weight terms. Their role is similar to that of quantifiers; in fact quantifiers can be viewed as special multiset operations.

In the case where Υ_w contains multiset operations, we add to the inductive definition of $L^*(\Upsilon)$ the following **multiset operation rule**:

Syntax: Let $\bar{x}$ and $\bar{y}$ be tuples of variables, $F(\bar{x}, \bar{y})$ be a weight term of vocabulary Υ and φ be a formula of vocabulary Υ. Then, for every multiset operation Γ of Υ_r, the expression

$$\Gamma_{\bar{x}}(F(\bar{x}, \bar{y}) : \varphi)$$

is a weight term of vocabulary Υ, with free variables $\bar{y}$.

Semantics: Let $G(\bar{y})$ be the weight term $\Gamma_{\bar{x}}(F(\bar{x}, \bar{y}) : \varphi)$. The interpretation of $G(\bar{y})$ on an Υ-structure $\mathfrak{D}$ with valuation $\bar{b}$ for $\bar{y}$ is

$$G^{\mathfrak{D}}(\bar{b}) := \Gamma(\{\!\{F^{\mathfrak{D}}(\bar{a}, \bar{b}) : \text{for all } \bar{a} \text{ such that } \mathfrak{D} \models \varphi(\bar{a}, \bar{b})\}\!\}).$$

To enhance readability, we will sometimes omit the free variables and use the abbreviated notation $\Gamma_{\bar{x}}(F : \varphi)$. Furthermore, we may omit true φ and write simply $\Gamma_{\bar{x}} F(\bar{x}, \bar{y})$.

There are some important multiset operations that are invariant under adding arbitrary occurrences of 0 to the multiset: $\Gamma(S) = \Gamma(S \cup \{\!\{0, 0, \ldots, 0\}\!\})$ for all $S \in \mathrm{fm}(R)$. On $\mathbb{N}$, for instance, this is the case for $\sum$ and max. In such case, we may use $\Gamma_{\bar{x}}(F \cdot \chi[\varphi])$ rather than $\Gamma_{\bar{x}}(F : \varphi)$.

Example 3.2. (Binary representations) Consider arithmetical structures with primary part of the form $\mathfrak{A} = (\{0, \ldots, n-1\}, <, P)$ where P is a unary relation. P is interpreted as a bit sequence $u_0 \cdots u_{n-1}$ representing the natural number $\sum_{i=0}^{n-1} u_i 2^i$ (where $u_i = 1$ iff $\mathfrak{A} \models P(i)$). The number represented by P is definable by the term

$$\sum_x (\chi[Px] \prod_y (2 : y < x)).$$

Example 3.3. (Counting elements) On arithmetical structures, we can count in FO*. For any formula $\varphi(\bar{x})$ there is a weight term $\#_{\bar{x}}[\varphi(\bar{x})]$ counting the number of tuples $\bar{a}$ such that $\varphi(\bar{a})$ is true. Indeed, let

$$\#_{\bar{x}}[\varphi(\bar{x})] := \sum_{\bar{x}} \chi[\varphi].$$

Example 3.4. (Counting equivalence classes) Let $\mathfrak{D}$ be an arithmetical structure and $\varphi(x, y)$ be a binary formula, defining an equivalence relation $\sim_\varphi$ on A. If we have division as a basic function in $\mathfrak{N}$, then the index of $\sim_\varphi$, denoted $\#[A/\varphi]$, is definable in FO* in the following way.

By the previous example, $F(x) = \#_y[\varphi(x, y)]$ is a weight term of FO*. The index of $\sim_\varphi$ can be written as a sum of rational numbers: $\#[A/\varphi] = \sum_x (F(x))^{-1}$. To do everything over $\mathbb{N}$, let $G = \prod_x F(x)$; thus the weight $G/F(x)$ is also FO*-definable and we get

$$\#[A/\varphi] = \Big(\sum_x G/F(x)\Big)/G.$$

Multiset operations play an important rôle in metafinite model theory. They partially compensate for the limited access to the secondary part and greatly enhance the expressive power of the logics that we consider. We also believe that they provide the right logical formalism for the *aggregate operators* used in real-life database query languages (see e.g. [1, Chapter 7.3: "Confronting the Real World"]).

3.3 An excursion: reliability of database queries

We present here a more elaborate example for the use of multiset operations that addresses the issue of fault tolerance of relational queries mentioned in Sect. 1.3.

Definition 3.5. An *unreliable database* is a pair $(\mathfrak{A}, \mu)$ where $\mathfrak{A}$ is a finite structure and μ a probability function on the set of atomic statements $R\bar{a}$ about $\mathfrak{A}$.

Think about $\mathfrak{A}$ as the observed database. For every first-order statement $\varphi(\bar{a})$ about $\mathfrak{A}$, let $\mathrm{Wrong}(\varphi(\bar{a}))$ be the event that the truth-value of $\varphi(\bar{a})$ in $\mathfrak{A}$ differs from the truth-value of $\varphi(\bar{a})$ in the actual database. $\mu(R\bar{a})$ is the probability of the event $\mathrm{Wrong}(R\bar{a})$. It is supposed that the events $\mathrm{Wrong}(R\bar{a})$ are independent.

Let $\mathfrak{B}$ be a database of the same vocabulary as $\mathfrak{A}$ and with the same universe as $\mathfrak{A}$. Let $D(\mathfrak{B})$ be the collection of atomic statements $R\bar{a}$ that are true in $\mathfrak{B}$. The probability that $\mathfrak{B}$ is the actual database is

$$\nu(\mathfrak{B}) := \prod_{\varphi \in |D(\mathfrak{A}) - D(\mathfrak{B})|} \mu(\varphi) \prod_{\varphi \in D(\mathfrak{A}) \Leftrightarrow \varphi \in D(\mathfrak{B})} (1 - \mu(\varphi))$$

where $|D(\mathfrak{A}) - D(\mathfrak{B})|$ is the symmetric difference of $D(\mathfrak{A})$ and $D(\mathfrak{B})$.

Given a relational query $\psi(\bar{x})$ of arity k, let $\psi^{\mathfrak{A}} = \{\bar{a} \in A^k : \mathfrak{A} \models \psi(\bar{a})\}$. The *Hamming distance* between $\psi^{\mathfrak{A}}$ and $\psi^{\mathfrak{B}}$ is the cardinality of the symmetric difference $|\psi^{\mathfrak{A}} - \psi^{\mathfrak{B}}|$.

Definition 3.6. Fix an unreliable database $(\mathfrak{A}, \mu)$ and a k-ary query ψ. For every tuple $\bar{a} \in A^k$, let $P_\psi(\bar{a})$ be the probability of the event $\mathrm{Wrong}(\psi(\bar{a}))$. Summing up $P_\psi(\bar{a})$ over all tuples $\bar{a} \in A^k$ gives the expectation $E(H_\psi)$ of the Hamming distance between $\psi^{\mathfrak{A}}$ and $\psi^{\mathfrak{B}}$ where $\mathfrak{B}$ is the actual database. The number $F_\psi := 1 - [E(H_\psi)/n^k]$ is the *fault-tolerance* of ψ.

Note that the expected Hamming distance $E(H_\psi)$ is a numerical invariant on unreliable databases. It assigns to any given $(\mathfrak{A}, \mu)$ the expectation of $|\psi^{\mathfrak{A}} - \psi^{\mathfrak{B}}|$. Also the fault tolerance F_ψ is a numerical invariant. If ψ has free variables, then $P_\psi(\bar{x})$ is a global function of positive arity; if ψ is a sentence then P_ψ and $E(H_\psi)$ coincide and $F_\psi = 1 - P_\psi$.

We are interested in definability and complexity questions for these invariants. Similar notions for studying query reliability appear in [18].

Unreliable databases can be modeled by metafinite structures where the secondary part $\mathfrak{R}$ is the field of reals with the multiset operations $\sum, \prod$. Let $(\mathfrak{A}, \mu)$ be an unreliable database of relational vocabulary Υ_a. View μ as a tuple of probability functions μ_R where R is a proper predicate (not the equality sign) in Υ_a and μ_R is the restriction of μ to atoms of the form $R\bar{a}$. With an unreliable database $(\mathfrak{A}, \mu)$, we associate the metafinite structure $(\mathfrak{A}, \mathfrak{R}, \{\mu_R : R \in \Upsilon_a\})$. We investigate the following questions:

Definability: Is it true that the expected Hamming distance (between the results of ψ on the observed database and actual database) and the fault-tolerance F_ψ of every first-order query are *first-order definable* numerical invariants?

If this should indeed be the case and if, moreover, a first-order definition of the fault-tolerance of ψ can be uniformly and efficiently generated from ψ, then one can automatically modify any first-order query so that the evaluation gives not only the result of the query on the observed database, but also its reliability. This brings us to the second question.

Complexity: What is the computational complexity (with respect to the size of the unreliable database) of calculating the expected Hamming distance and the fault-tolerance of first-order queries. Here we can either assume, that the given probabilities are rational, or we take a real-number model of computation.

Of course the same questions can be asked for other query languages and for more complicated (and more realistic) models for unreliable databases. But this will be done elsewhere.

We first consider the *quantifier-free queries.*

Proposition 3.7. *Let $\psi(\bar{x})$ be a quantifier-free. Then*

(i) $P_\psi(\bar{x})$ is a first-order definable global weight function;

(ii) The expected Hamming distance and the fault-tolerance of ψ are first-order definable numerical invariants.

(iii) The expected Hamming distance and the fault-tolerance of ψ on a given unreliable database $(\mathfrak{A}, \mu)$ are computable in polynomial time.

Proof. Note that *(iii)* is an immediate consequence of *(ii)*. Since $F_\psi = 1 - E(H_\psi)/n^k$ and $E(H_\psi) = \sum_{\bar{x}} P_\psi(\bar{x}))$, it suffices to prove *(i)*.

Let $N(x_1, \ldots, x_k)$ be the assertion that $x_i \neq x_j$ if $i < j$, and let $N(\psi(\bar{x})) = N(\bar{x})$. It suffices to find a weight term that expresses P_ψ only in the case when $N(\psi)$ holds. Indeed, let $\alpha_1, \ldots, \alpha_m$ be all different complete and consistent assertions about the equality relation on the components of $\bar{x}$. P_ψ is the sum of probabilities $P_{\alpha_i \wedge \psi}$, and each $\alpha_i \wedge \psi$ is equivalent to a formula of the form $N(\bar{y}) \wedge \varphi(\bar{y})$.

If β is an atom $R\bar{x}$ then, by definition, $P_\beta(\bar{x}) = \mu_R(\bar{x})$. Further $P_\neg\beta(\bar{x}) = P_\beta(x)$. Let now $\psi(\bar{x}) = \beta_1 \wedge \cdots \wedge \beta_m$ be a conjunction of m literals (i.e. atoms and negated atoms) which are distinct and none is the negation of another one.

Let T be the set of the 2^m formulae $\varphi = \beta_1' \wedge \cdots \wedge \beta_m'$ where the literals β_i' and β_i either coincide or one is the negation of the other. We identify each $\varphi \in T$ with the set of its subformulae, i.e we sometimes write $\beta \in \varphi$ to express that β is one of the subformulae of φ.

Suppose that for the observed database $\mathfrak{A}$, we have $\mathfrak{A} \models \varphi(\bar{a})$ (and hence $\mathfrak{A} \models \neg\varphi'$ for all other $\varphi' \in T$). Since the terms $P_\beta(\bar{x})$ describe probabilities of independent events, the term

$$G_{\varphi,\psi}(\bar{x}) := \prod_{\beta \in \psi \cap \varphi} (1 - P_\beta(\bar{x})) \prod_{\beta \in \psi - \varphi} P_\beta(\bar{x})$$

describes the probability that $\psi(\bar{a})$ holds in the actual database. The event Wrong$(\psi(\bar{a}))$ is the disjoint union of

(a) the event that $\mathfrak{A} \models \psi(\bar{a})$, but $\neg\psi(\bar{a})$ holds in the actual database,

(b) the events that $\mathfrak{A} \models \varphi(\bar{a})$ for some $\varphi \neq \psi$ from T, but $\psi(\bar{a})$ is true in the actual database.

Thus, the probability of the event Wrong($\psi(\bar{a})$) is described by

$$P_\psi(\bar{x}) := \chi[\psi(\bar{x})](1 - G_{\psi,\psi}(\bar{x})) + \sum_{\varphi \in T - \{\psi\}} \chi[\varphi(\bar{x})] G_{\varphi,\psi}(\bar{x})$$

which is a first-order term.

This generalizes to quantifier-free queries as follows. Let $\psi(\bar{x})$ be a quantifier-free formula satifying $N(\psi)$, and let $\beta_1, \ldots, \beta_m$ be the atoms occuring in ψ. Again we form the set T of all conjunctions $\beta_1' \wedge \cdots \wedge \beta_m'$ where $\beta_i' = \beta_i$ or $\beta_i' = \neg\beta_i$. Further, let $T^+(\psi) = \{\varphi \in T : \varphi \models \psi\}$ and $T^-(\psi) = \{\varphi \in T : \varphi \models \neg\psi\} = T - T^+(\psi)$. Clearly ψ is logically equivalent to $\bigvee_{\varphi \in T^+(\psi)} \varphi$. It thus follows that the probabilty of the event Wrong($\psi(\bar{x})$) is descibed by the term

$$P_\psi(\bar{x}) := \sum_{\varphi \in T^+(\psi)} \Big(\chi[\varphi(\bar{x})] \sum_{\vartheta \in T^-(\psi)} G_{\varphi,\vartheta}(\bar{x})\Big) + \sum_{\varphi \in T^-(\psi)} \Big(\chi[\varphi(\bar{x})] \sum_{\vartheta \in T^+(\psi)} G_{\varphi,\vartheta}(\bar{x})\Big).$$

This proves the proposition. □

Remark. Note that the term describing P_ψ may have exponential length with respect to ψ. But this is no major problem because we are mainly interested in data complexity: the query is fixed and the complexity is measured in terms of the size of the database. This is the usual and reasonable practice in database theory because the length of the query is usually much smaller than the size of the database.

As we prove next, it is unlikely that Proposition 3.7 can be generalized to all first-order queries. In fact, unless P = #P, even the error probabilities of *conjunctive queries* cannot be computed in polynomial time. (Thus a claim made in [18], to the effect that the reliability of any first-order query is polynomial-time computable, appears to be incorrect.)

Recall that conjunctive queries are queries of the form $\exists x_1 \cdots \exists x_k(\varphi_1 \wedge \cdots \wedge \varphi_\ell)$ where each φ_i is atomic. The class #P consists of all functions f into $\mathbb{N}$ for which there exists a nondeterministic polynomial-time Turing machine such that the number of accepting computations on any input for M coincides with the value of f on that input. For many NP-complete decision problems and also for some problems in P, the related problem of counting the number of witnesses (rather than determining whether there exists at least one) is #P-complete. For background on #P we refer to [50, 59].

Proposition 3.8. *There exist conjunctive queries ψ such that calculating the value of P_ψ is* #P*-hard.*

Proof. We will reduce the problem #Monotone 2-Sat to the problem of computing the value of P_ψ for a conjunctive Boolean query ψ.

The problem #Monotone 2-Sat, proved to be #P-complete by Valiant [59], takes as input instances propositional formula in 2-CNF without negations, i.e. formulae of the form

$\bigwedge_{i=1}^{n} Y_i \vee Z_i$ where Y_i and Z_i are propositional variables. The desired answer is the number of satisfying assignments.

A propositional formula of this form can be modeled by structure (A, L, R) where the universe A is the (disjoint) union of the set of clauses and the set of propositional variables of the formula and the atomic statements Luv (resp. Ruv) express that the left (resp. right) variable in clause u is v. Further, we model an assignment of truth values to the propositional variables by the set S of variables that are set to *false* under this assignment.

Given a positive 2-CNF formula $\bigwedge_{i=1}^{n} Y_i \vee Z_i$ one can construct in polynomial time the unreliable database $(\mathfrak{A}, \mu)$ where $\mathfrak{A} = (A, L, R, S)$ models the given formula together with the assignment that sets all variables to *false* (thus S is the set of all variables in the formula). The error probability are defined as follows: All atomic statements Luv, Ruv have error probability 0, and

$$\mu(Sv) = \begin{cases} 1/2 & \text{if } v \text{ is a variable} \\ 0 & \text{otherwise.} \end{cases}$$

Thus the probability space associated with $(\mathfrak{A}, \mu)$ is essentially the uniform distribution over all assignments of truth values to the variables in the given 2-CNF formula.

Now, consider the conjunctive query

$$\psi := \exists x \exists y \exists z (Lxy \wedge Rxz \wedge Sy \wedge Sz)$$

which expresses, on $\mathfrak{A} = (A, L, R, S)$, that the assignment defined by S does *not* satisfy the formula modeled by (A, L, R). Clearly $\mathfrak{A} \models \psi$ and the error probabilty P_ψ is just the number of assignments that satisfy the given formula, divided by the total number of assignments. Thus, if we could calculate the error probability of ψ in polynomial time, we could solve #MONOTONE 2-SAT (and thus any problem in #P) in polynomial-time. $\square$

Remark. It is easy to see that computing P_ψ is in #P for all first-order ψ.

3.4 Pure term calculi

We now explain how, for metafinite algebras, logics can be presented as pure calculi of weight terms. We first assume, for simplicity, that the secondary part $\mathfrak{R}$ is an algebra, i.e. the vocabulary Υ_r contains no relation symbols. Thus we deal with vocabularies $\Upsilon = (\Upsilon_r, \Upsilon_w)$ where Υ_w is a set of function symbols and Υ_r a set of function and multiset operation symbols.

Definition 3.9. FOT(Υ) is the calculus of *first-order terms* of vocabulary Υ. The set of terms and the notion of the *rank* for the terms (that will be exploited later) are defined inductively as follows:

(i) If $x_1, \ldots, x_k$ are variables, and w is a k-ary weight function in Υ_w, then $w(x_1, \ldots, x_m)$ is a term of rank 0 in FOT(Υ).

(ii) If $F_1, \ldots, F_m \in$ FOT(Υ) and $g \in \Upsilon_r$ is a m-ary function symbol, then $g(F_1, \ldots, F_m)$ is a term of FOT(Υ), whose rank is the maximum of the ranks of $F_1, \ldots, F_m$.

(iii) If F and G belong to $\mathrm{FOT}(\Upsilon)$, then so does $\chi[F = G]$. The rank of $\chi[F = G]$ is the maximum of the ranks of F and G.

(iv) If F and G belong to $\mathrm{FOT}(\Upsilon)$, $\bar{y}$ is an ℓ-tuple of variables and Γ a multiset operation from Υ_r, then $\Gamma_{\bar{y}}(F(\bar{x},\bar{y}) : G = 1)$ is a term of $\mathrm{FOT}(\Upsilon)$, of rank $\ell + \max\{\mathrm{rk}(F), \mathrm{rk}(G)\}$.

We use also a simplified form $\Gamma_{\bar{y}}\ F(\bar{x},\bar{y})$ as an abbreviation for $(\Gamma_{\bar{y}}\ (F(\bar{x},\bar{y}) : 1 = 1)$.

In the cases e.g. of arithmetical or $\mathbb{R}$-structures with maximization as a multiset operation, first-order logic can be simulated by FOT, in the sense that the characteristic function of every first-order formula is equivalent to a term in FOT. Indeed, this follows by a straightforward induction using the following equalities:

$$\begin{aligned} \chi[\psi \wedge \varphi] &= \chi[\psi]\chi[\varphi] \\ \chi[\neg\psi] &= 1 - \chi[\psi] \\ \chi[\exists x \psi] &= \max_x(\chi[\psi]). \end{aligned}$$

In fact, this holds for all secondary parts as long as we have two definable functions $\wedge$ and $\neg$, interpreted on $\{0,1\} \subseteq R$ in the usual way, and any multiset operation that distinguishes, say, multisets with occurrences of 1 from those without (or the empty multiset from the nonempty ones).

Remark. The restriction to algebraic structures is not necessary. When we deal with an arbitrary vocabulary $\Upsilon = (\Upsilon_a, \Upsilon_r, \Upsilon_w)$ for metafinite structures, we can still present first-order logic as a pure calculus of weight terms. We just have to replace in clause *(i)* of Definition 3.9 the variables x_i by arbitrary point terms over Υ_a (as in clause *(ii)* of Definition 3.1), and add the rules defining for every predicate Q and already defined terms $F_1, \ldots, F_m$ also the term $\chi[Q(F_1, \ldots, F_m)]$.

3.5 Second-order multiset operations

In several contexts, for instance for dealing with NP-optimization problems or with counting problems in the class #P, it is convenient to have logics with second-order constructs.

Multiset operations can be viewed as a generalization of quantifiers. Therefore, natural variants of second order logics can be defined by applying multiset operations to predicate variables.

Definition 3.10. Suppose we have a logic L in the usual sense (say, second-order logic or its existential fragment Σ^1_1), then L^{**} is the smallest logic closed under the rules of L^* together with the following rule.

Multiset operation rule (second order):

Syntax: Let $\Upsilon = (\Upsilon_a, \Upsilon_r, \Upsilon_w)$ be a vocabulary and $\Upsilon' = (\Upsilon_a \cup \{\bar{X}\}, \Upsilon_r, \Upsilon_w)$ where $\bar{X}$ is a tuple of relation variables. If F is a weight term and φ a formula of vocabulary Υ' with free variables among $\bar{x}, \bar{y}$, then, for every multiset operation Γ of Υ_r, the expression

$$\Gamma_{\bar{X},\bar{x}}(F : \varphi)$$

is a weight term of vocabulary Υ, with free variables $\bar{y}$.

Semantics: The interpretation of this expression on an Υ-structure $\mathfrak{D}$ with valuation $\bar{b}$ for $\bar{y}$ is

$$\Gamma\{\!\{F^{(\mathfrak{D},\bar{X})}(\bar{a},\bar{b}) : (\mathfrak{D},\bar{X}) \models \varphi(\bar{a},\bar{b})\}\!\}.$$

Example 3.11. (The Travelling Salesman Problem) NP-optimization problems like the TSP can be expressed in a very direct way in this framework, since the arithmetic that is necessary to determine the length of a tour and to minimize is separated from the graph.

Let *order*$(<)$ express that $<$ is a linear ordering, and let *succ*$(<, x, y)$ be a formula which, for any given linear ordering $<$, says that either y is the successor of x, or x is the maximal and y the minimal element of the ordering. Then the length of the shortest tour of any instance (V, w) of the TSP, where $w : V \times V \to \mathbb{N}$ is the weight function giving the distances, is defined by the weight

$$opt_{\mathrm{TSP}}(V, w) = \min_{<}\Big(\sum_{x,y}(w(x, y) : succ) : order\Big).$$

A more challenging example: the genus of a graph. The *genus* $\gamma(G)$ of an undirected graph G is the smallest $g \in \mathbb{N}$ such that G can be embedded into the sphere with g handles.

The genus is one of the most important graph parameters. It is hard to compute; the corresponding decision problem — given a graph G and a number k, decide whether $\gamma(G) \leq k$ — is NP-complete.

It is more convenient for us to work with a different, purely combinatorial characterization of the genus.

Definition 3.12. A *rotation system* on a undirected graph $G = (V, E)$ is a ternary predicate $P \in V^3$ which defines for every node a cycle on the edges incident to it. More precisely: if $(x, y, z) \in P$, then $(x, y) \in E$ and $(y, z) \in E$, and for all $y \in V$, the directed graph $H_y = (S_y, C_y)$ with

$$\begin{aligned} S_y &:= \{x : (x, y) \in E\} \\ C_y &:= \{(x, z) : (x, y, z) \in P\} \end{aligned}$$

is a cycle. A *P-face* is defined by a cycle $x_0, \ldots, x_{r-1}$ in G such that, for all $i < r$, $(x_{i-1}, x_i, x_{i+1}) \in P$ (here, indices are expressed modulo r). The *P-genus* of G, denoted $\gamma(P)$ is defined by Euler's formula

$$n - e + f(P) = 2 - 2\gamma(P)$$

where n is the number of vertices, e the number of edges and $f(P)$ the number of P-faces.

The following result is well-known in graph theory (see e.g. [25])

Proposition 3.13. *The genus of G is the minimal P-genus of G.*

For convenience our logical definition of the genus is based on transitive closure logic. This is a familiar logic in finite model theory which augments first-order logic by the ability to define transitive closures. It admits, for every formula $\varphi(\bar{x}, \bar{y})$ with k-tuples $\bar{x}, \bar{y}$ of free variables, also the formula $[\mathrm{TC}_{\bar{x},\bar{y}}\ \varphi](\bar{a}, \bar{b})$ expressing that $(\bar{a}, \bar{b})$ is contained in the reflexive and transitive closure of the binary relation that φ defines on k-tuples.

It is easy to see that there exists a formula ψ of vocabulary $\{E, P\}$ in transitive closure logic such that for every graph G and every ternary predicate P on G

$$(G, P) \models \psi \text{ if and only if } P \text{ is a rotation system on } G.$$

The number of P-faces is the number of equivalence classes of directed edges with respect to the reachability relation defined by P. It is not difficult to construct a formula $\alpha(P, Q)$ in transitive closure logic saying that Q is a binary relation containing at most one directed edge on each P-face:

$$\begin{aligned}\alpha(P, Q) \quad = \quad & \forall x \forall y \forall u \forall v ((Qxy \wedge Quv \wedge \\ & [\mathrm{TC}_{xy,uv}\, y = u \wedge P(x, y, v)](xy, uv)) \rightarrow (x, y) = (u, v)).\end{aligned}$$

Given that $\psi(P)$ expresses that P is a rotation system, that the weight $\#Q$ is definable in $FO^*(\{Q\})$ and that n and e are obviously definable, we can define the genus of an undirected graph by

$$\gamma = 1 + \tfrac{1}{2}(e - n - \max_{P,Q}(\#Q : \psi \wedge \alpha)).$$

4 Descriptive complexity

One of the goals of metafinite model theory is the *descriptive complexity theory of problems with weights*. For finite models, the results of Fagin, Immerman, Vardi and others provide logical characterizations of NP, P and also for most of the other important complexity classes, at least on ordered structures. We refer to the survey articles [21, 28, 36, 37], to Chapter 6 in [19] and Chapter 2.3 in [7] for background on descriptive complexity.

Here we investigate generalizations of these results in the realm of metafinite structures. For simplicity, we focus on arithmetical structures; we also mention $\mathbb{R}$-structures but refer to [23] for proofs. However, the approach can be extended to problems on metafinite structures with arbitrary secondary part. This requires the definition of a suitable computation model and a suitable notion of complexity. We will defer the detailed development to a subsequent paper.

We start with the observation that first-order formulae can be evaluated in polynomial-time.

Proposition 4.1. *If the basic functions, relations and multiset operations of $\mathfrak{R}$ can be evaluated in polynomial time (with respect to the given cost function), then the same is true for every first-order definable global function on $M_\Upsilon[\mathfrak{R}]$.*

The proof is a straightforward induction.

4.1 Metafinite spectra

We first consider Fagin's characterization of NP by existential second-order logic [20].

Definition 4.2. A class $\mathcal{K}$ of finite Υ_a-structures is a *generalized spectrum* if there exists a first-order sentence ψ of a vocabulary $\Upsilon_a \cup \{R_1, \ldots, R_m\}$ such that $\mathfrak{A} \in \mathcal{K}$ if and only if there exists an expansion $\mathfrak{B}$ of $\mathfrak{A}$ with $\mathfrak{B} \models \psi$.

Remark. An equivalent definition is that a generalized spectrum is the class of finite models of an existential second-order sentence $\exists R_1 \cdots \exists R_m \psi$. However, as discussed below, there are several possibilities of generalizing second-order logic to metafinite structures, and we don't want to commit ourselves to one particular variant. We will therefore mostly work with (generalizations of) the definition given above.

Informally, Fagin's Theorem states that the generalized spectra are precisely the model classes recognizable in nondeterministic polynomial time. For a precise statement of this result, we have to keep in mind that to serve as an input for a classical computational device like a Turing machine, a finite structure needs to be encoded by a string. At least implicitly, such an encoding requires that an ordered representation of the structure is chosen. The precise form of the encoding is not important, as long as it satisfies some reasonable simple properties. So when we say that a class of structures is in NP we actually mean that the set of encodings of structures in that class is in NP.

Theorem 4.3 (Fagin). *Let $\mathcal{K}$ be a class of finite structures of a fixed finite vocabulary which is closed under isomorphisms. Then $\mathcal{K}$ is in NP if and only if it is a generalized spectrum.*

Does Fagin's Theorem generalize to metafinite structures? To address this problem, we need to make precise two notions:

- The notion of a *metafinite spectrum*, i.e. a generalized spectrum of metafinite structures.
- The notion of nondeterministic polynomial time complexity in the context of metafinite structures.

We start with two notions of metafinite spectra. Recall the $M_\Upsilon[\mathfrak{R}]$ is the class of metafinite structures with secondary part $\mathfrak{R}$ and vocabulary $\Upsilon = (\Upsilon_a, \Upsilon_r, \Upsilon_w)$ (where of course Υ_r is the vocabulary of $\mathfrak{R}$).

Definition 4.4. A class $\mathcal{K} \subseteq M_\Upsilon[\mathfrak{R}]$ is a *metafinite spectrum* if there exists a first-order sentence ψ of a vocabulary $\Upsilon' \supseteq \Upsilon$ such that $\mathfrak{D} \in \mathcal{K}$ if and only if there exists an expansion $\mathfrak{D}' \in M_{\Upsilon'}[\mathfrak{R}]$ of $\mathfrak{D}$ with $\mathfrak{D}' \models \psi$. (Note that the secondary part is not expanded.) A *primary metafinite spectrum* is defined in a similar way, except that only the primary part of the structures is expanded, but not the set of weight functions. This means that the expanded structures $\mathfrak{D}'$ have the same set of weight functions as $\mathfrak{D}$.

Remark. These two notions of metafinite spectra correspond to two variants of (existential) second-order logic. The more restrictive one allows second-order quantifiers only over primary relations, whereas the general one allows quantification over weight functions as well. Thus, a primary metafinite spectrum is the class of models $\mathfrak{D} \in M_\Upsilon[\mathfrak{R}]$ which are models of an existential second-order sentence of the form $\exists R_1 \cdots \exists R_m \psi$ where $R_1, \ldots, R_m$ are relation variables over the primary part and ψ is first-order (in the sense of Definition 3.1). Since relations over the primary part can be replaced by their characteristic functions, a metafinite spectrum in the more general sense is the class of models of a sentence $\exists F_1 \cdots \exists F_m \psi$ where F_i are function symbols ranging over weight functions.

4.2 Generalizations of Fagin's Theorem

We show that both notions of metafinite spectra capture (suitable variants of) nondeterministic polynomial-time in certain contexts, but fail to do so in others.

First we consider arithmetical structures where the secondary part is $\mathfrak{N}$, as given by Definition 2.5. We assume that the cost of natural numbers is given by the length of their binary representations. As described in Sect. 2.3, this gives a natural notion of the complexity of global functions, and in particular of an NP-class of arithmetical structures. So the question is whether, or under what circumstances, NP is captured by the class of metafinite spectra or primary metafinite spectra.

The original proof of Fagin's Theorem generalizes to the case of arithmetical structures with not too large weights.

Definition 4.5. A class $\mathcal{K}$ of metafinite structures has *small weights* if there exists a $k \in \mathbb{N}$ such that $\max \mathfrak{D} \leq |\mathfrak{D}|^k$ for all $\mathfrak{D} \in \mathcal{K}$.

Recall that $\max \mathfrak{D}$ stands for the *cost* of the largest weight. Thus, a class of arithmetical structures has small weights if the values of the weights are bounded by a function $2^{p(|\mathfrak{D}|)}$ for some polynomial p. We obtain the following first generalization of Fagin's result.

Theorem 4.6. *Let $\mathcal{K} \subseteq M_\Upsilon[\mathfrak{N}]$ be a class of arithmetical structures with small weights, which is closed under isomorphisms. The following are equivalent:*

(i) $\mathcal{K}$ is in NP.

(ii) $\mathcal{K}$ is a primary generalized spectrum.

Proof. It is obvious that *(ii)* implies *(i)*. The converse can be reduced to Fagin's Theorem as follows. We assume that for every structure $\mathfrak{D} = (\mathfrak{A}, \mathfrak{N}, W)$ in $\mathcal{K}$, we have that $\max \mathfrak{D} \leq n^k$ where $n = |\mathfrak{D}| = |\mathfrak{A}|$; further we suppose without loss of generality that an ordering $<$ on A is available (otherwise we expand the vocabulary with a binary relation $<$ and add a conjunct $\beta(<)$ asserting that $<$ is a linear order). We can then identify A^k with the initial subset $\{0, \ldots, n^k - 1\}$ of $\mathbb{N}$, viewed as bit positions of the binary representations of the weights of $\mathfrak{D}$. With every $\mathfrak{D} \in \mathcal{K}$ we associate a finite structure $\mathfrak{D}_f$ by expanding the primary part $\mathfrak{A}$ as follows: For every weight function $w \in W$ of arity j we add a new relation P_w of arity $j + k$ with

$$P_w := \{(\bar{a}, \bar{t}) : \text{ the } \bar{t}\text{-th bit of } w(\bar{a}) \text{ is } 1\}.$$

Then $\mathcal{K}$ is in NP if and only if $\mathcal{K}_f = \{\mathfrak{D}_f : \mathfrak{D} \in \mathcal{K}\}$ is an NP-set of finite structures, and in fact, we can choose the encodings in such a way that $\mathfrak{D}$ and $\mathfrak{D}_f$ are represented by the same binary string. Thus, if $\mathcal{K}$ is in NP, then by Fagin's Theorem $\mathcal{K}_f$ is a generalized spectrum, defined by a first-order sentence ψ.

As in Example 3.2, one can construct a first-order sentence α (whose vocabulary consists of the weight functions $w \in \Upsilon_w$ and the corresponding primary relations P_w), which expresses that the P_w encode the weight functions w in the sense defined above. Then $\psi \wedge \alpha$ is a first-order sentence witnessing that $\mathcal{K}$ is a primary metafinite spectrum. ☐

Remark. The same result holds for simple arithmetical structures.

However, without the restriction that the weights be small, it is no longer true that every NP-set is a primary metafinite spectrum. If we have inputs with huge weights compared to the primary part, then relations over the primary part cannot encode enough information to describe computations that are bounded by a polynomial in the length of the weights.

It is tempting to use unrestricted metafinite spectra instead. However, metafinite spectra in the general sense capture a much larger class than NP, namely the class of all recursively enumerable sets!

We first note that any tuple $\bar{a} \in \mathbb{N}^k$ can be viewed as an arithmetical structure with the empty primary vocabulary and k nullary weight functions $a_1, \dots, a_k$. Thus an arithmetical relation $S \subseteq \mathbb{N}^k$ can be viewed as a special class of arithmetical structures.

Theorem 4.7. *Every recursively enumerable set $S \subseteq \mathbb{N}^k$ is a metafinite spectrum. In particular, there exist undecidable metafinite spectra.*

Proof. By Matijasevich's Theorem (see [46]) every recursively enumerable set $S \subseteq \mathbb{N}^k$ is Diophantine, i.e. can be represented as

$$S = \{\bar{a} \in \mathbb{N}^k : \text{ there exists } b_1, \dots, b_m \in \mathbb{N} \text{ such that } Q(\bar{a}, \bar{b}) = 0\}$$

for some polynomial $Q \in \mathbb{Z}[x_1, \dots, x_k, y_1, \dots, y_m]$. Let $P, P' \in \mathbb{N}[\bar{x}, \bar{y}]$ such that $Q(\bar{x}, \bar{y}) = P(\bar{x}, \bar{y}) - P'(\bar{x}, \bar{y})$. Thus S is a metafinite spectrum; the desired first-order sentence uses additional weight functions $b_1, \dots, b_m$ and asserts that $P(\bar{a}, \bar{b}) = P'(\bar{a}, \bar{b})$. □

This can be extended to any r.e. class of arithmetical structures, with arbitrary vocabulary. To prove this, we describe how to encode structures $\mathfrak{D} \subseteq M_\Upsilon[\mathfrak{N}]$ by tuples $c(\mathfrak{D}) \in \mathbb{N}^k$ where k depends only on Υ. (In fact, it is no problem to reduce k to 1.) For future use of such encodings we will be more restrictive than necessary for this result.

Similar to the case of finite structures, an encoding involves the selection of a linear order on the primary part. In fact we find it more convenient to have a *ranking* of the primary part rather than just a linear ordering.

Definition 4.8. Suppose that $\mathfrak{R}$ contains a copy of $(\mathbb{N}, <)$. A *ranking* of a metafinite structure $\mathfrak{D} = (\mathfrak{A}, \mathfrak{R}, W)$ is a bijection $r : A \to \{0, \dots, n-1\} \subseteq R$. A class $\mathcal{K} \subseteq M_\Upsilon[\mathfrak{R}]$ is *ranked* if Υ contains a weight function r whose interpretation on every $\mathfrak{D} \in \mathcal{K}$ is a ranking.

From a ranking one can trivially define a linear order of the primary part. Also a ranking r can be extended to a ranking $r_m : A^m \to \{0, \dots, n^m - 1\}$ of m-tuples. On the other hand, a ranking need *not* be first-order definable from a linear order; take e.g. $\mathfrak{R} = (\mathbb{N}, <)$. However, on arithmetical structures, $\sum$ is available and thus a ranking is definable from a linear order by $r(x) = \sum_y \chi[y < x]$.

We write $\mathcal{R}_\Upsilon$ for the class of *ranked arithmetical structures* of vocabulary Υ.

Lemma 4.9 (Coding Lemma.). *For every vocabulary Υ of ranked arithmetical structures there exists an encoding function*

$$\begin{aligned} c :\ & \mathcal{R}_\Upsilon \longrightarrow \mathbb{N}^k \\ & \mathfrak{D} \longmapsto c(\mathfrak{D}) = c_1(\mathfrak{D}), \dots, c_k(\mathfrak{D}) \end{aligned}$$

with the following properties:

(i) c is definable by first-order terms;

(ii) The primary part and the weight functions of $\mathfrak{D}$ can be reconstructured from $c(\mathfrak{D})$ in polynomial time;

(iii) there exists a polynomial $p(n,m)$ such that $c_i(\mathfrak{D}) \le 2^{p(|\mathfrak{D}|, \max \mathfrak{D})}$ for every $i \le k$.

Proof. Encode every weight function $w : A^m \to \mathbb{N}$ by a pair (q,s) of natural numbers, where

$$\begin{aligned} q &= \max_{\bar{x}} w(\bar{x}) + 1 \\ s &= \sum_{\bar{x}} w(\bar{x}) q^{r_m(\bar{x})}. \end{aligned}$$

This encoding is first-order definable: for q this is obvious, and

$$s = \sum_{\bar{x}} \Big(w(\bar{x}) \prod_{\bar{y}} (q : r_m(\bar{y}) < r_m(\bar{x})) \Big).$$

To encode $\mathfrak{D}$ we pass to the associated algebra $\mathfrak{D}^a$ and represent it by the sequence of pairs (q,s) that encode the weight functions of $\mathfrak{D}^a$. Obviously, properties *(i), (ii), (iii)* are satisfied. □

Theorem 4.10. *Every recursively enumerable class of arithmetical structures is a metafinite spectrum.*

Proof. Let $\mathcal{K} \subseteq M_\Upsilon[\mathfrak{N}]$ be recursively enumerable. Then the set

$$c(\mathcal{K}) := \{c(\mathfrak{D}, r) : \mathfrak{D} \in \mathcal{K},\ r \text{ is a ranking of } \mathfrak{D}\} \subseteq \mathbb{N}^k$$

is also recursively enumerable and therefore Diophantine. The desired first-order sentence ψ uses, besides the symbols of Υ, a unary weight function r and nullary weight functions $b_1, \ldots, b_m$ and expresses *(i)* that r is a ranking and *(ii)* that $Q(c(\mathfrak{D}, r), \bar{b})) = 0$ for a suitable polynomial $Q \in \mathbb{Z}[x_1, \ldots, x_k, y_1, \ldots, y_m]$ defining $c(\mathcal{K})$. □

Conversely, it is easy to see that every metafinite spectrum of arithmetical structures is recursively enumerable, so we obtain:

Corollary 4.11. *On arithmetical structures, metafinite spectra capture the r.e. sets.*

But there are other contexts where metafinite spectra do indeed capture (a suitable notion of) nondeterministic polynomial-time. An important example are computations over the real numbers with the model of Blum-Shub-Smale.

Theorem 4.12 (Grädel, Meer). $\mathrm{NP}_\mathbb{R}$ *coincides with the class of metafinite spectra of $\mathbb{R}$-structures.*

The proof is given in [23].

Definition 4.13. Let $\mathcal{K} \subseteq M_\Upsilon[\mathfrak{R}]$, and suppose that we have fixed a cost function on R. We say that $\mathcal{K}$ is a *polynomially bounded metafinite spectrum* if there exists a first-order sentence ψ of vocabulary $\Upsilon' \supseteq \Upsilon$ and a polynomial $p(n,m)$ such that $\mathcal{K}$ is the class of all $\mathfrak{D} \in M_\Upsilon[\mathfrak{R}]$ for which there exists an expansion $\mathfrak{D}'$ with

- $\mathfrak{D}' \models \psi$
- $\max \mathfrak{D}' \leq p(|\mathfrak{D}|, \max \mathfrak{D})$

Remark. If the cost function is universally bounded by a constant (as in the case of $\mathbb{R}$-structures), then trivially every metafinite spectrum is polynomially bounded.

Conjecture 4.14. *Let $\mathcal{K} \subseteq M_\Upsilon[\mathfrak{N}]$ be a class of arithmetical structures, which is closed under isomorphism. Then the following are equivalent:*

(i) $\mathcal{K} \in \mathrm{NP}$.

(ii) $\mathcal{K}$ is a polynomially bounded metafinite spectrum.

It is not difficult to prove that every polynomially bounded metafinite spectrum is in NP, i.e. that *(ii)* implies *(i)*. The other direction is related to a conjecture of Adleman and Manders concerning the notion of *Diophantine complexity* (see [3, 4, 33, 39, 43, 46]).

Adleman and Manders introduced the class D of all relations $S \subseteq \mathbb{N}^k$ that can be represented in the form

$$\bar{a} \in S \Longleftrightarrow \exists y_1 \cdots \exists y_m \Big(\bigwedge_{i=1}^{m} y_i \leq 2^{\max_i \|a_i\|^\ell} \wedge Q(\bar{a}, \bar{y}) = 0 \Big)$$

for some $\ell \in \mathbb{N}$ and some polynomial Q with integer coefficients. They conjectured that every arithmetical relation in NP can be given such a Diophantine representation, i.e. that $D = \mathrm{NP}$. This conjecture implies (and in fact is equivalent to) Conjecture 4.14.

It is obvious that the analogue of Conjecture 4.14 for PTA-structures is true, since there we have all polynomial-time computable functions available. But in fact, much weaker expansions of $\mathfrak{N}_0$ will do as well. Let $\tilde{\mathfrak{N}}$ be obtained from $\mathfrak{N}_0$ by adding at least one of the following functions or relations:

- the so-called *logical and* function, mapping numbers a, b with binary expansions $a = \sum_{i=0}^{m} a_i 2^i$ and $b = \sum_{i=0}^{\ell} b_i 2^i$ to
$$a \& b := \sum_{i=0}^{\min(\ell,m)} \min(a_i, b_i) 2^i.$$
- the partial order $\preceq$ with $a \preceq b$ iff $a\&b = a$, (i.e. every bit of a is less than or equal to the corresponding bit of b);
- the function $(a, b, c) \mapsto \binom{a}{b} \pmod{c}$;
- the modular factorial function $(a, b) \mapsto a! \pmod{b}$.

Then, results of Jones and Matijasevich [39] imply

Theorem 4.15. *Every class in NP of arithmetical structures with secondary part $\tilde{\mathfrak{N}}$ is a polynomially bounded metafinite spectrum.*

The ordering $\preceq$ or the logical and can be directly used to describe computations. Binomial coefficients, and therefore factorials, suffice to define $\preceq$ since $a \preceq b$ if and only if $\binom{a}{b}$ is odd. This follows from Lucas' theorem that, for every prime p, given p-ary representations $a = \sum_i a_i p^i$ and $b = \sum_i b_i p^i$ we have that $\binom{a}{b} = \prod_i \binom{a_i}{b_i} \pmod p$.

We can reformulate Theorem 4.15 as follows. If $\mathcal{K}$ is an isomorphism-closed class of arithmetical structures with secondary part $\tilde{\mathfrak{N}}$ (or PTA), then $\mathcal{K}$ is in NP if and only if it can be characterized as the model class of a second-order sentence with bounded quantifiers in the following way:

$$\mathfrak{D} \in \mathcal{K} \text{ iff } \mathfrak{D} \models (\exists F_1 \leq 2^{p(n,m)}) \cdots (\exists F_k \leq 2^{p(n,m)})\psi$$

where ψ is first-order and p is a polynomial. Here $(\exists F_i \leq 2^{p(n,m)}) \dots$ is to be understood as an abbreviation for $\exists F_i[\forall \bar{x}(F_i(\bar{x}) \leq 2^{p(|\mathfrak{D}|, \max \mathfrak{D})}) \wedge \dots]$.

From results of Hodgson and Kent [33, 43], we obtain a more involved characterization that works also for the secondary part $\mathfrak{N}$, and in fact also for simple arithmetical structures. Here, the second-order prefix has besides the exponentially bounded existential quantifiers $(\exists F_i \leq 2^{p(n,m)})$, also polynomially bounded universal quantifiers of the form $(\forall G_i \leq p(n,m))$. Hodgson and Kent proved that if one generalizes the class D of Adleman and Manders by allowing also polynomially bounded universal quantifiers in the prefix, then one obtains a precise arithmetical characterization of NP. In fact one can even do away with all but one of these universal quantifiers and obtain a normal form which is the analogue to the so-called Davis normal form for r.e sets. The Davis normal form theorem says that every recursively enumerable set $S \subseteq \mathbb{N}^k$ can be represented as

$$S = \{\bar{a} \in \mathbb{N}^k : \exists y_1 (\forall z \leq y_1) \exists y_2 \cdots \exists y_m Q(\bar{a}, \bar{y}, z) = 0\}$$

(where $Q \in \mathbb{Z}[x_1, \dots, x_k, y_1, \dots, y_m, z]$); it was an important step towards the eventual solution of Hilbert's 10th problem by Matijasevich. For NP-classes of arithmetical structures this gives the following logical characterization.

Theorem 4.16. *An isomorphism-closed class $\mathcal{K} \subseteq M_\Upsilon[\mathfrak{N}]$ is in NP if and only if there exists a first-order formula ψ and a polynomial $p(n,m)$ such that $\mathcal{K}$ is the class of all $\mathfrak{D} \in M_\Upsilon[\mathfrak{N}]$ with*

$$\mathfrak{D} \models (\exists F_1 \leq 2^{p(n,m)})(\forall G \leq p(n,m))(\exists F_2 \leq 2^{p(n,m)}) \cdots (\exists F_k \leq 2^{p((n,m)})\psi.$$

4.3 Fixed point logics and polynomial-time

Fixed point logics on finite structures. In finite model theory, fixed point logics play a central rôle. They provide a general and flexible method of inductively defining new predicates and thus remedy one of the main deficiencies (with respect to expressiveness) of first order logic: the lack of a mechanism for unbounded recursion or iteration.

We recall the definition of (inflationary) fixed point logic [27]. Let Υ_a be a vocabulary, $R \notin \Upsilon_a$ an r-ary predicate and $\psi(\bar{x})$ a formula of vocabulary $\Upsilon_a \cup \{R\}$ with free variables $\bar{x} = x_1, \dots, x_r$. Then ψ defines, for every finite Υ_a-structure $\mathfrak{A}$, an operator $F_\psi^{\mathfrak{A}} : \mathcal{P}(A^r) \to \mathcal{P}(A^r)$ on the class of r-ary relations over A by

$$F_\psi^{\mathfrak{A}} : R \longmapsto R \cup \{\bar{a} : (\mathfrak{A}, R) \models \psi(\bar{a})\}.$$

By definition, this operator is *inflationary*, i.e $R \subseteq F_\psi^{\mathfrak{A}}(R)$ for all $R \subseteq A^r$. Therefore the inductive sequence $R^0, R^1, \ldots$ defined by $R^0 := \varnothing$ and $R^{j+1} := F_\psi^{\mathfrak{A}}(R^j)$ is increasing, i.e. $R^j \subseteq R^{j+1}$ and therefore reaches a fixed point $R^j = R^{j+1}$ for some $j \leq |A|^r$. It is called the *inflationary fixed point* of ψ on $\mathfrak{A}$, and denoted by R^∞.

Definition 4.17. The (inflationary) fixed point logic FP is defined by adding to the syntax of first order logic the *fixed point formation rule:* if $\psi(\bar{x})$ is a formula of vocabulary $\sigma \cup \{R\}$ as above and $\bar{u}$ is an r-tuple of terms, then

$$[\mathrm{FP}_{R,\bar{x}}\ \psi](\bar{u})$$

is a formula of vocabulary Υ_a, whose semantics is that $\bar{u} \in R^\infty$.

Example 4.18. Here is a fixed point formula that defines the reflexive and transitive closure of the binary predicate E:

$$\mathrm{TC}(u,v) \equiv [\mathrm{FP}_{T,x,y}\ (x = y) \vee (\exists z)(Exz \wedge Tzy)](u,v).$$

Many other variants of fixed point logics have been studied, most notably the *least fixed point logic*, denoted LFP, and the *partial fixed point logic*, denoted PFP. It was proved independently by Immerman [35] and Vardi [60] that, on ordered finite structures, LFP characterizes precisely the queries that are computable in polynomial time. Gurevich and Shelah [30] proved that FP and LFP have the same expressive power on finite structure, so in particular, FP also characterizes PTIME in the presence of a linear ordering. On the class of arbitrary (not necessarily ordered) finite structures, FP and LFP are strictly weaker than PTIME-computability. In fact, on very simple classes of structures, such as structures with the empty vocabulary (i.e. pure sets), FP collapses to first-order logic. Also, the 0-1 law holds for FP, which shows that, on arbitrary finite structures, FP cannot express non-trivial statements about cardinalities.

The fixed point logic FP^***.** Definition 3.1 gives a general way of extending a logic L for finite structures to a logic L^* for metafinite structures. Applying this definition to FP, we get the logic FP^*, the extension of first-order logic FO^* by the rule for building fixed point formulae $[\mathrm{FP}_{R,\bar{x}}\ \psi](\bar{u})$ of vocabulary $(\Upsilon_a, \Upsilon_r, \Upsilon_w)$ from already given formulae ψ of vocabulary $(\Upsilon_a \cup \{R\}, \Upsilon_r, \Upsilon_w)$. It is important to emphasize that the inductively defined predicate R is a *predicate over the primary part* and that $\bar{u}$ is a tuple of point terms. We first observe that the fixed point construction preserves PTIME-computability.

Proposition 4.19. *If the basic functions, relations and multiset operations of* $\mathfrak{R}$ *can be evaluated in polynomial time (with respect to the given cost function), then the same is true for all* FP^**-definable global functions on* $M_\Upsilon[\mathfrak{R}]$.

As in the case of Fagin's Theorem we can also transfer Immerman's and Vardi's logical characterization of PTIME to the case of arithmetical structures with small weights.

Theorem 4.20. *Let* $\mathcal{K} \subseteq \mathcal{R}_\Upsilon$ *be a class of ranked arithmetical structures with small weights. For every global function* G *on* $\mathcal{K}$ *the following are equivalent*

(i) G *is computable in polynomial time.*

(ii) G is FP**-definable.*

We omit the proof, which follows by straightforward application of the same arguments as in the proof of Theorem 4.6.

Again, as in the case of metafinite spectra, the restriction to small weights is necessary. For an extreme example, consider polynomial-time predicates $S \subseteq \mathbb{N}$. Each such S gives rise to a decision problem where an instance is an arithmetical structure $\mathfrak{D}$, with a single nullary weight a, and the question is whether $a \in S$. Of course this problem is completely independent of the primary part of the structure, which in particular can be trivial. Fixed point constructions are of absolutely no help here and neither are quantifiers or multiset operations. Thus FP* can decide S if and only if the characteristic function $\chi_S(a)$ is available as a basic term. Obviously there exist polynomial-time predicates S for which this is not the case.

Thus, FP* cannot fully capture PTIME on arithmetical structures, even in the presence of a ranking.

But this is not the only weakness of FP*. Another important limitation is the *absence of any recursion mechanism over numbers and weight functions.* We will exhibit certain interesting consequences of this, by comparing the power of FP* with the fixed point logic with counting (FP + C) on unordered structures. This logic does not include large numbers in the secondary sort, but it has recursion over relations that range over both parts.

Fixed point logic with counting. As we mentioned already in the introduction, among the logics studied in finite model theory, (FP + C) is the closest to our approach. It was first proposed by Immerman, who started from the observation that counting is probably the most basic class of low-complexity queries not expressible in fixed point logic. The original hope was that the addition of counting to FP in a reasonable way should give a logic that could express all of PTIME. It should be pointed out, that there are different ways of adding counting mechanisms to a logic, which are not necessarily equivalent. The most straightforward possibility is the addition of quantifiers of the form $\exists^{\geq 2}$, $\exists^{\geq 3}$, etc., with the obvious meaning. While this is perfectly reasonable for the infinitary logics $L^k_{\infty\omega}$, it is not general enough for fixed point logic, because it does not allow recursion over the counting parameters i in quantifiers $\exists^{\geq i}x$. In fact if the counting parameters are fixed numbers, then adjoining the quantifiers $\exists^{\geq i}x$ does not give additional power to logics whose formulae may have an arbitrary number of variables (as FO or FP). These counting parameters should therefore be considered as variables that range over the natural numbers. To define in a precise way a logic with counting and recursion which is applicable also to counting the numbers, one extends the original objects of study, namely finite (one-sorted) structures $\mathfrak{A}$ to two-sorted auxiliary structures $\mathfrak{A}^*$ with a second numerical (but also finite) sort.

We are now ready to formally introduce (FP + C). With any one-sorted finite structure $\mathfrak{A}$, one associates the two-sorted structure $\mathfrak{A}^* := (\mathfrak{A}, (\mathbf{n}, <))$ with a copy of $\mathfrak{A}$ on the first sort elements and the linear order $(\mathbf{n}, <)$ on the second sort elements, where $n = |A| + 1$ and $<$ is the standard order on $\mathbf{n} = \{0, \dots, n-1\}$.

We take $n = |A| + 1$ rather than $n = |A|$ to be able to represent the cardinalities of all subsets of $|A|$ within n.

We start with first-order logic and two-sorted vocabularies $(\Upsilon_a, \{<\})$, with the usual semantics over structures $\mathfrak{A}^*$. Latin letters $x, y, z, \dots$ are used as variables over the first sort, and Greek letters $\lambda, \mu, \nu, \dots$ as variables over the second sort. Note that, contrary to logics of metafinite structures, we have here no restriction on the access of the logic to second sort elements. For instance, we can quantify over number variables to build formulae of the form $\exists\mu\varphi$.

The two sorts are related by *counting terms*, defined by the following rule: Let $\varphi(x)$ be a formula with a free variable x of sort one, then $\#_x[\varphi]$ is a second-sort term, with the set of free variables $\mathrm{free}(\#_x[\varphi]) = \mathrm{free}(\varphi) - \{x\}$. The interpretation of $\#_x[\varphi]$ is the number of first-sort elements a that satisfy $\varphi(a)$. First-order logic with counting, denoted (FO + C), is the closure of two-sorted first-order logic under counting terms.

Example 4.21. To illustrates the use of counting terms we present a formula $\psi(E_1, E_2) \in$ (FO + C) expressing that two equivalence relations E_1 and E_2 over the first sort are isomorphic.

$$\psi(E_1, E_2) := (\forall\mu)(\#_x[\#_y[E_1xy] = \mu] = \#_x[\#_y[E_2xy] = \mu]).$$

The *(inflationary) fixed point logic with counting* (FP + C) is obtained by adding to (FO + C) the mechanism for building fixed point predicates that may range over both sorts.

Definition 4.22. The logic (FP + C) is the closure of two-sorted first-order logic under

(i) the rule for building counting terms;

(ii) the usual rules of first-order logic for building terms and formulae;

(iii) the fixpoint formation rule: Suppose that $\psi(\bar{x}, \bar{\mu})$ is a formula of vocabulary $\Upsilon \cup \{R\}$ where $\bar{x} = x_1, \ldots, x_k$, $\bar{\mu} = \mu_1, \ldots, \mu_\ell$, and R has mixed arity (k, ℓ), and that $(\bar{u}, \bar{\nu})$ is a $k + \ell$-tuple of first- and second-sort terms, respectively. Then

$$[\mathrm{FP}_{R,\bar{x},\bar{\mu}}\ \psi](\bar{u}, \bar{\nu})$$

is a formula of vocabulary Υ.

The semantics of $[\mathrm{FP}_{R,\bar{x},\bar{\mu}}\ \psi]$ on $\mathfrak{A}^*$ is defined in the same way as for the logic FP, namely as the inflationary fixed point R^∞ of the operator

$$F_\psi^{\mathfrak{A}^*} : R \longmapsto R \cup \{(\bar{u}, \bar{\nu}) \mid (\mathfrak{A}^*, R) \models \psi(\bar{u}, \bar{\nu})\}.$$

(FP + C) was first introduced by Immerman, in a different but equivalent form, with counting quantifiers rather than counting terms. The present version appeared first in [24].

Example 4.23. An interesting example for an (FP + C)-computable global function is the *stable colouring* of a graph. Given a graph G with a colouring $f : V \to 0, \ldots, r$ of its vertices, we define a refinement f' of f, where vertex x has the new colour $f'x = (fx, n_1, \ldots, n_r)$ where $n_i = \#y[Exy \wedge (fy = i)]$. The new colours can be sorted lexicographically so that they form again an initial subset of $\mathbb{N}$. Then the process can be iterated until a fixed point, the stable colouring of G is reached. It is known that almost all graphs have the property that no two vertices have the same stable colour. Thus stable colourings provide a polynomial-time graph-canonization algorithm for a dense class of graphs. It should be clear that the stable colouring of a graph is definable in (FP + C) (see [38] for more details).

Over arithmetical structures, we can define counting in FO^* and hence FP^*, as shown in Example 3.3. One might therefore feel that FP^*, having both a fixed point constructor and the ability to count, is at least as powerful as (FP + C).

To make this a precise question, we have to consider a setting where the two logics can be compared. We compare their expressive powers on classes $\mathcal{K} \subseteq \mathrm{Fin}(\Upsilon_a)$ of finite, one-sorted structures.

Definition 4.24. With every finite structure $\mathfrak{A}$ and every secondary part $\mathfrak{R}$ we associate the metafinite structure $\mathfrak{A}_{\mathfrak{R}} := (\mathfrak{A}, \mathfrak{R}, \varnothing)$, with primary part $\mathfrak{A}$, secondary part $\mathfrak{R}$ and the empty set of weight functions. We say, that a model class $\mathcal{K} \subseteq \mathrm{Fin}(\Upsilon_a)$ of finite structures is FP^*-definable over $\mathfrak{R}$, if there exists a sentence $\psi \in \mathrm{FP}^*$ such that

$$\mathcal{K} = \{\mathfrak{A} \in \mathrm{Fin}(\Upsilon_a) : \mathfrak{A}_{\mathfrak{R}} \models \psi\}.$$

As usual we say that $\mathcal{K}$ is (FP + C)-definable if there exists a sentence $\theta \in (\mathrm{FP} + \mathrm{C})$ such that

$$\mathcal{K} = \{\mathfrak{A} \in \mathrm{Fin}(\Upsilon_a) : \mathfrak{A}^* \models \theta\}.$$

Proposition 4.25. *Let* $\mathfrak{N}$ *be any reduct of* PTA. *Then every model class* $\mathcal{K} \subseteq \mathrm{Fin}(\Upsilon_a)$ *which is* FP^**-definable over* $\mathfrak{N}$*, is also* (FP + C)*-definable.*

Proof. This follows by straightforward induction over terms and formulae of FP^*, using the facts that *(i)* every FP^*-definable global function can be evaluated in polynomial time and that *(ii)* every polynomial-time computable function or relation appearing in the secondary part can be expressed by an (FP + C)-definition over the numerical sort (since the numerical sort is ordered). □

The converse is not always true. Indeed, let $\mathfrak{N} = \mathfrak{N}_0$. If we consider the case that $\Upsilon_a = \varnothing$, then, by taking cardinalities, a class $\mathcal{K} \subseteq \mathrm{Fin}(\varnothing)$ can be viewed as a set of natural numbers. On $\mathrm{Fin}(\varnothing)$, (FP + C) captures polynomial-time with respect to the cardinality of the structures, i.e. $\mathcal{K} \subseteq \mathrm{Fin}(\varnothing)$ is (FP + C)-definable if and only if $\{1^n : n \in \mathcal{K}\}$ is decidable in polynomial time. On the other hand, FP^* on structures $(A, \mathfrak{N}_0, \varnothing)$ is equivalent to FO^* whose power can be precisely described as follows: Every sentence φ can be written as a Boolean combination of inequalities $f(n) \leq g(n)$ where $f, g \in T$ are terms in one variable n that represents the cardinality of A. Since all elements of A are indistinguishable, the terms $\sum_x F$ or $\prod_x F$ produced by means of the multiset operations can simply be rewritten as $n \cdot F$ and F^n respectively. (Applications of max and min have no effect at all.) Thus the set T of terms can be defined by closing the constants and n under addition, multiplication and under raising to nth power (i.e. given $t(n)$, one can form $t(n)^n$). A simple diagonalization arguments proves that there exist predicates $S \subseteq \mathbb{N}$ which cannot be defined in this way, but nevertheless $\{1^n : n \in S\}$ is decidable in polynomial time.

Indeed, let φ be a Boolean combination of inequalities $f \leq g$ with $f, g \in T$. Syntactically, φ is a string in a finite alphabet whose symbols are $0, 1, n, +, \cdot$, etc. We can order this alphabet and assign numbers to strings in the usual way. Let $n(\varphi)$ be the number associated with φ and S be the set of those numbers $n(\varphi)$ such that φ is false at $n(\varphi)$. Clearly, S is not defined by any φ. Moreover, since φ is equivalent to $(\varphi \wedge 0 < 1)$, $(\varphi \wedge 0 < 1 \wedge 0 < 1)$ etc., $\{n : \varphi \text{ holds at } n\}$ differs from S on infinitely many numbers.

It thus suffices to prove that there exists a polynomial-time algorithm that, given $1^{n(\varphi)}$, computes the truth value of φ at $n(\varphi)$ and inverts the result. This is obvious, once we have checked, by an easy induction on the formation rules of T, that for every term $f \in T$, the logarithm of the value $f(n)$ is bounded by a polynomial in n.

We thus have proved the following result.

Proposition 4.26. *There exist model classes $\mathcal{K}$ of finite structures which are* (FP + C)*-definable, but not* FP**-definable over $\mathfrak{N}_0$.*

The fact that $\mathfrak{N}_0$ forms a counterexample to the converse of Proposition 4.25 survives various enrichments of $\mathfrak{N}_0$. In fact, the same proof works if $\mathfrak{N}_0$ is extended by any finite collection of polynomial-time computable functions and any finite collection of multiset operations Γ such that the value of Γ at multisets $\{\!\{t, t, \dots, t\}\!\}$, consisting of n occurrences of t, can be computed in polynomial-time with respect to n and $\log t$. However, there is a limit to such generalizations. We will prove in Sect. 5 that the converse of Proposition 4.25 does hold in the case that $\mathfrak{N} = \mathrm{PTA}$.

Remark. Note that the problem of capturing polynomial-time on *ranked* PTA-structures is trivial and does not require a fixed-point construction. As pointed out above, if a ranking is available, then the primary part can be encoded by a tuple of natural numbers and this encoding is definable by first-order terms. Any polynomial-time property is thus reducible to a PTIME property of numbers which is a basic relation of PTA. Thus a global function on ranked PTA-structures is PTIME-computable if and only if it is first-order definable. Furthermore FO* and FP* coincide on ranked PTA-structures.

4.4 A functional fixed point logic

One possibility to overcome the limitations of languages of type L^* is to apply recursion in one way or another to weight functions.

We discuss here, as one particular example, a *fixed-point calculus* for *partially defined weight functions*. It is convenient to deal with partial functions by extending the secondary part $\mathfrak{R}$ with a new element to a structure $\mathfrak{R}^*$ with universe $R \cup \{\mathrm{undef}\}$ in the following way:

The relations of $\mathfrak{R}^*$ coincide with their restrictions to $\mathfrak{R}$, and the functions and multiset operations of $\mathfrak{R}$ are extended to $\mathfrak{R}^*$ in some arbitrary way. For many functions, the natural choice will be to set $f^{\mathfrak{R}^*}(\bar{a}) = \mathrm{undef}$ whenever the argument $\bar{a}$ contains undef. However, for some functions there are other reasonable possibilities: For multiplication, it actually makes more sense to set

$$a \cdot \mathrm{undef} = \mathrm{undef} \cdot a = \begin{cases} 0 & \text{if } a = 0 \\ \mathrm{undef} & \text{if } a \neq 0. \end{cases}$$

Fix a signature Υ and a function symbol Z not contained in Υ. Let $G(Z, \bar{x})$ be a weight term of signature $(\Upsilon_a, \Upsilon_r, \Upsilon_w \cup \{Z\})$ and free variables $\bar{x} = x_1, \dots, x_r$ where r is the arity of Z. We write $G^{\mathfrak{D},Z}(\bar{x})$ for the value of $G(Z, \bar{x})$ for a given interpretation $(\mathfrak{D}, Z)$.

For every structure $\mathfrak{D} \in M_\Upsilon[\mathfrak{R}^*]$, the term $G(Z, \bar{x})$ gives rise to an operator $F_G^{\mathfrak{D}}$ which updates *partially defined* functions Z as follows:

$$F_G^{\mathfrak{D}}(Z)(\bar{a}) := \begin{cases} G^{\mathfrak{D},Z}(\bar{a}) & \text{if } Z(\bar{a}) = \text{undef} \\ Z(\bar{a}) & \text{otherwise.} \end{cases}$$

This gives an inductive definition of a sequence of partial weight functions $Z^j : A^r \to \mathbb{R}$.

Z^0 is undefined everywhere (i.e. $Z^0(\bar{a}) = \text{undef}$ for all $\bar{a}$)
$Z^{j+1} = F_G^{\mathfrak{D}}(Z^j)$.

The operator $F_G^{\mathfrak{D}}$ updates Z only at points where Z is undefined, so this process reaches a fixed point after a polynomial number of iterations: $Z^j = Z^{j+1}$ for some $j \leq |A|^r$. We denote this fixed point by Z^∞ and call it the *fixed point of* $G(Z, \bar{x})$ *on* $\mathfrak{D}$.

Definition 4.27. *Functional fixed point logic*, denoted FFP, is obtained by augmenting the first-order logic FO* with the following rule for building terms:

If $G(Z, \bar{x})$ is a weight term of signature $(\Upsilon_a, \Upsilon_r, \Upsilon_w \cup \{Z\})$, if $\bar{x} = x_1, \dots, x_r$ is a tuple of variables (where r is the arity of Z), and if $\bar{u} = u_1, \dots, u_r$ is a tuple of point terms, then

$$\mathbf{fp}[Z(\bar{x}) \leftarrow G(Z, \bar{x})](\bar{u})$$

is a weight term of signature $(\Upsilon_a, \Upsilon_r, \Upsilon_w)$. Its value on a given structure $\mathfrak{D}$, is $Z^\infty(\bar{u})$.

Note that, on arithmetical structures, FFP can define weights of double exponential magnitude. Indeed suppose we have an arithmetical structure with a ranking r and let us adopt the conventions that max and + produce undef whenever any of the arguments is undefined, and that $0 \cdot \text{undef} = 0$. Set

$$G(Z, x) := 2\chi[r(x) = 0] + \max_y\Big(\chi[r(x) = r(y) + 1] \prod_z Z(y)\Big).$$

Then, for every structure $\mathfrak{D}$ with $|\mathfrak{D}| = n$ we have that

$$\mathbf{fp}[Z(x) \leftarrow G(Z, x)](y) = 2^{n^{r(y)}}.$$

This even works for simple arithmetical structures, because the term $\prod_z Z(y)$ — which evaluates to $Z(y)^n$ — can be simulated by a fixed point construction.

However, in the context of computations over $\mathbb{R}$ with the Blum-Shub-Smale model, the magnitude of the numbers is no serious problem, since one assumes unit cost for each $r \in \mathbb{R}$. In fact it has been shown in [23] that functional fixed-point logic is the right logic for describing polynomial-time computability in that model, in the sense that it gives rise to the following analogue of the Immerman-Vardi Theorem.

Theorem 4.28 (Grädel, Meer). *On ranked* $\mathbb{R}$*-structures,* FFP *captures* $\mathrm{P}_{\mathbb{R}}$.

Remark. For some applications the update operator $F_G^{\mathfrak{D}}$, used for FFP, may not be adequate, since the values different from undef are never updated. Instead we may consider a different update operator $\tilde{F}_G^{\mathfrak{D}}$ with

$$\tilde{F}_G^{\mathfrak{D}}(Z)(\bar{a}) := G^{\mathfrak{D},Z}(\bar{a}).$$

Of course, the inductive process defined by such an operator need not reach a fixed point. But — as in the case of the partial fixed point logic PFP considered in finite model theory — we can define Z^∞ to be the fixed point of the sequence $Z^0, Z^1, \ldots$, defined by $\tilde{F}_G^{\mathfrak{D}}$, if the fixed point exists, and some default value, e.g. the everywhere undefined function, otherwise.

We don't further investigate this approach here. The study of this, and related variants of functional fixed point logics, as well as other means of inductive definability of weight functions, is one of the promising directions for future research.

5 Back and forth from finite to metafinite structures

As explained in the introduction, our goal is to extend the approach and methods of finite model theory to the more general class of metafinite structures. We show in this section that an important methodology of finite model theory, namely the Ehrenfeucht-Fraïssé games and their various generalizations, is indeed applicable in our more general context.

The aspect that we consider here is the indistinguishability of two metafinite structures by (infinitary) logics with a bounded number of variables, but with arbitrary multiset operations. We show that this reduces to the indistinguishability of two associated *finite* structures by *first-order* formulae *with counting.*

Throughout this section, we consider structures with a fixed secondary part $\mathfrak{R}$ and assume that the primary part is always relational.

5.1 Indistinguishability by logics with k variables

Definition 5.1. Let $\mathfrak{D} = (\mathfrak{A}, \mathfrak{R}, W)$ and $\mathfrak{D}' = (\mathfrak{B}, \mathfrak{R}, W')$ be structures in $M_\Upsilon(\mathfrak{R})$, let $\bar{a}$ and $\bar{b}$ be ℓ-tuples of elements of $\mathfrak{A}$ and $\mathfrak{B}$ respectively, and let L be a logic of metafinite structures. We say that $(\mathfrak{D}, \bar{a})$ and $(\mathfrak{D}', \bar{b})$ are L-equivalent — in symbols: $(\mathfrak{D}, \bar{a}) \equiv_L (\mathfrak{D}', \bar{b})$ — if for every weight term $F(x_1, \ldots, x_\ell)$ of L,

$$F^{\mathfrak{D}}(\bar{a}) = F^{\mathfrak{D}'}(\bar{b}).$$

Since in our logics we have for every formula its characteristic function available as a weight term, the L-equivalence of $(\mathfrak{D}, \bar{a})$ and $(\mathfrak{D}', \bar{b})$ implies in particular that for every formula $\varphi(\bar{x})$ of L

$$\mathfrak{D} \models \varphi(\bar{a}) \text{ if and only if } \mathfrak{D}' \models \varphi(\bar{b}).$$

The converse does not necessarily hold, i.e., two structures may be indistinguishable by formulae of L but there nevertheless may exist a weight term that separates them. This may be the case when $\mathfrak{R}$ contains unreachable elements which do not appear as values of any closed Υ_r-term.

Logics with k variables. We first recall the definitions of some logics with bounded number of variables that are of great importance in finite model theory. L^k is the fragment of first-order logic with variables, free and bound, among $x_1, \ldots, x_k$. The infinitary logic $L^k_{\infty\omega}$ is the closure of L^k under conjunctions and disjunctions applied to arbitrary sets of formulae. Further, $L^\omega_{\infty\omega} = \bigcup_{k\in\omega} L^k_{\infty\omega}$. It is well-known that the familiar fixed point logics LFP, IFP and PFP are sublogics of $L^\omega_{\infty\omega}$.

The logics C^k, $C^k_{\infty\omega}$ and $C^\omega_{\infty\omega}$ are the extension of L^k, $L^k_{\infty\omega}$ and $L^\omega_{\infty\omega}$ by means of counting quantifiers $\exists^{\geq 2}$, $\exists^{\geq 3}$, etc., with the obvious semantics. One of the reasons why these logics are important is that $C^\omega_{\infty\omega}$ is an extension of fixed point logic with counting (FP + C).

Equivalence with respect to $L^k_{\infty\omega}$ has an elegant characterization in terms of the k-pebble game [6, 34, 52], an infinitary variant of Ehrenfeucht-Fraïssé games. There is a similar pebble game appropriate to $C^k_{\infty\omega}$ [38]. It is played by two players, I and II, on two structures $\mathfrak{A}$ and $\mathfrak{B}$ of the same relational signature. They have k pairs of pebbles.

A move of the game is played as follows.

1. Player I chooses $i \leq k$ and picks up the i-th pair of pebbles. He selects a nonempty subset X of either A or B. Player II chooses a subset Y in the other structure with $|Y| = |X|$. If no such set exists, the game is over and Player I has won.

2. Player I places an i-pebble on an element $y \in Y$. Player II puts the other i-pebble on an element $x \in X$.

After any move, the pebbles on the 'board' define a partial map from A to B, taking every pebbled element of A to the element of B carrying the corresponding pebble. Player II has to maintain the condition that the pebble map is a partial isomorphism. We say that Player II wins the C^k-game on $(\mathfrak{A}, a_1, \dots, a_\ell)$ and $(\mathfrak{B}, b_1, \dots, b_\ell)$ if she has a strategy to maintain this condition forever, when initially the first ℓ pairs of pebbles are placed on $(a_1, b_1), \dots, (a_\ell, b_\ell)$.

Theorem 5.2 (Immerman, Lander). *The following are equivalent*

(i) Player II wins the C^k-game on $(\mathfrak{A}, \bar{a})$ and $(\mathfrak{B}, \bar{b})$.

(ii) $\mathfrak{A} \models \varphi(\bar{a})$ iff $\mathfrak{B} \models \varphi(\bar{b})$ for every formula $\varphi(\bar{x}) \in C^k_{\infty\omega}$.

Here is another way to put and to refine this (see [24, 49]). For a tuple $\bar{a} \in (A \cup \{*\})^k$ (where $*$ serves as a dummy value in the case that not all k variables are actually used) we write $\bar{a}\frac{c}{j}$ for the tuple obtained by substituting (or adding) c at position j to $\bar{a}$.

We write $(\mathfrak{A}, \bar{a}) \sim_i (\mathfrak{B}, \bar{b})$ if Player II has a strategy to maintain the winning condition for at least i moves of the C^k-game, starting at position $(\mathfrak{A}, \bar{a})$ and $(\mathfrak{B}, \bar{b})$. Note that $(\mathfrak{A}, \bar{a}) \sim_0 (\mathfrak{B}, \bar{b})$ if and only if $p : \bar{a} \longmapsto \bar{b}$ is a partial isomorphism from $\mathfrak{A}$ to $\mathfrak{B}$.

Theorem 5.3. *$(\mathfrak{A}, \bar{a}) \sim_{i+1} (\mathfrak{B}, \bar{b})$ if and only if $(\mathfrak{A}, \bar{a}) \sim_i (\mathfrak{B}, \bar{b})$ and for every $\sim_i$-equivalence class C and every $j \leq k$ we have that*

$$\#\{c \in A : (\mathfrak{A}, \bar{a}\tfrac{c}{j}) \in C\} = \#\{d \in B : (\mathfrak{B}, \bar{b}\tfrac{d}{j}) \in C\}.$$

Since $C^k_{\infty\omega}$-equivalence is the intersection of all equivalence relations $\sim_i$ one obtains the following characterization.

Theorem 5.4. *$C^k_{\infty\omega}$-equivalence is the coarsest equivalence relation $\sim$ with the following property: If $(\mathfrak{A}, \bar{a}) \sim (\mathfrak{B}, \bar{b})$ then*

(i) The function $p : \bar{a} \longmapsto \bar{b}$ is a partial isomorphism from $\mathfrak{A}$ to $\mathfrak{B}$;

(ii) for every $\sim$-equivalence class C and every $j \leq k$ we have that

$$\#\{c \in A : (\mathfrak{A}, \bar{a}\tfrac{c}{j}) \in C\} = \#\{d \in B : (\mathfrak{B}, \bar{b}\tfrac{d}{j}) \in C\}.$$

An infinitary k-variable term calculus for metafinite structures with multiset operations. $L_{\infty\omega}$ generalizes first-order logic. In a similar vein, we generalize the first-order term calculus FOT(Υ) given by Definition 3.9 and the subsequent remark (because Υ_a is not necessarily empty). Let FOT^k(Υ) be FOT(Υ) with terms using only the variables $x_1, \ldots, x_k$.

Define a *set operation* on a set R to be a unary operation from subsets of R to R. Let Υ^* be the extension of Υ with names for all multiset operations over R, and let $\mathfrak{R}^*$ be the corresponding expansion of $\mathfrak{R}$.

Definition 5.5. The term calculus $T^k_{\infty\omega}(\Upsilon, R)$ is the extension of FOT^k(Υ^*) (with the secondary part $\mathfrak{R}^*$) by the following rule: If S is a set operation on R and Φ is a set (any set) of terms, then $S(\Phi)$ is a term. The rank of $S(\Phi)$ is the supremum of the ranks of terms in Φ (which may be an infinite ordinal). The semantics is as follows: Given an evaluation of the variables, compute the set $X \subseteq R$ of the values of terms in Φ under that evaluation, and then apply S to X.

Remark. The relation of F being a proper subterm of a term G is well founded.

Remark. Let us see that the characteristic function of every $C^k_{\infty\omega}$ formula φ about the primary part is given by some term t_φ in $T^k_{\infty\omega}$. The characteristic functions of the primary relations are always available. If $\varphi = \neg\psi$ then the desired $t_\varphi = S(\{t_\psi\})$ where S is any set operation such that $S(\{0\}) = 1$ and $S(\{1\}) = 0$. If φ is a disjunction of formulas φ_i where $i \in I$ then $t_\varphi = S(\{t_{\varphi_i} : i \in I\})$ where S is any operation that coincides with max on nonempty subsets of $\{0, 1\}$. To handle counting quantifiers, let Γ^i be a multiset operation such that $\Gamma^i(m) = 1$ if m contains at least i occurrences of 1 and $\Gamma^i(m) = 0$ otherwise. If $\psi = \exists^{\geq i} x \varphi$ then $t_\psi = (\Gamma^i)_x t_\varphi$.

Example 5.6. Suppose that a metafinite structure $\mathfrak{D} = (\mathfrak{A}, \mathfrak{N}, W)$ is such that every element $a \in A$ is definable in $\mathfrak{A}$ by some formula $\varphi_a(x)$, and W contains a unary weight function w. Let S be a set function such that $S(X) = 1$ if and only if every number in X is prime. The term

$$S(\{\chi[\varphi_a](x) \cdot w(x) : a \in A\})$$

evaluates to 1 in $\mathfrak{D}$ if and only if the range of w consists of primes.

Remark. In the remainder of this section, we prove various theorems about the term calculus $T^k_{\infty\omega}$. The developed theory is quite robust with respect to the definition of $T^k_{\infty\omega}$. It does not change if the $T^k_{\infty\omega}$ is further enriched by means of even fancier super-operations over R; for example we may require that, for every finitary or infinitary operation $f(r_1, r_2, \ldots)$ over R and terms $t_i \in T^k_{\infty\omega}$, the possibly infinitary expression $f(t_1, t_2, \ldots)$ is a term in $T^k_{\infty\omega}$. On the other hand, as the remark above shows, we actually use only very simple set operations.

5.2 Partial isomorphisms and the multiset pebble game.

Consider a metafinite structure $\mathfrak{D} = (\mathfrak{A}, \mathfrak{R}, W) \in M_\Upsilon(\mathfrak{R})$. We associate with $\mathfrak{D}$ a finite structure $\mathrm{fin}(\mathfrak{D})$ with universe A, by expanding $\mathfrak{A}$ with relations

$$P_{w,r} := \{\bar{a} : w^{\mathfrak{D}}(\bar{a}) = r\}$$

for every function $w \in W$ and every element $r \in R$. Although the set of these new predicates is infinite, only finitely many relations are nonempty for each $w \in W$.

Definition 5.7. Let $\mathfrak{D} = (\mathfrak{A}, \mathfrak{R}, W)$ and $\mathfrak{D}' = (\mathfrak{B}, \mathfrak{R}, W')$ belong to $M_\Upsilon(\mathfrak{R})$. A *partial isomorphism* from $\mathfrak{D}$ to $\mathfrak{D}'$ is an injective function $p : A_0 \to B$ whose domain is $A_0 \subseteq A$ such that

- for every relation symbol $R \in \Upsilon_a$ and all elements $a_1, \dots, a_m \in A_0$

  $$\mathfrak{D} \models R(a_1, \dots, a_m) \text{ if and only if } \mathfrak{D}' \models R(pa_1, \dots, pa_m).$$

- for every function symbol $w \in \Upsilon_w$ and all elements $a_1, \dots, a_m \in A_0$ we have that

  $$w^{\mathfrak{D}}(a_1, \dots, a_m) = w^{\mathfrak{D}'}(pa_1, \dots, pa_m).$$

Thus, the partial isomorphisms from $\mathfrak{D}$ to $\mathfrak{D}'$ are precisely the partial isomorphisms from $\mathrm{fin}(\mathfrak{D})$ to $\mathrm{fin}(\mathfrak{D}')$.

We now describe the 'obvious' pebble game appropriate to the logic $T^k_{\infty\omega}$. Given two metafinite structures $\mathfrak{D} = (\mathfrak{A}, \mathfrak{R}, W)$ and $\mathfrak{D}' = (\mathfrak{B}, \mathfrak{R}, W')$ in $M_\Upsilon(\mathfrak{R})$, the T^k-game on $(\mathfrak{D}, \mathfrak{D}')$ is played with k pairs of pebbles on the 'board' (A, B). A move of the T^k-game is played as follows:

1. Player I selects $\ell \le k$ pairs of pebbles and selects a function $f : A^\ell \to R$. Player II chooses a function $g : B^\ell \to R$ such that $\mathrm{mult}(f) = \mathrm{mult}(g)$. (Recall that $\mathrm{mult}(f) = \{\!\{ f(\bar{a}) : \bar{a} \in A^\ell \}\!\}$.) If no such function exists, the game is over and Player I has won.
2. Player I puts the selected pebbles on elements $b_1, \dots, b_\ell \in B$. Player II puts the corresponding pebbles on $a_1, \dots, a_\ell$ such that $f(a_1, \dots, a_\ell) = g(b_1, \dots, b_\ell)$.

Remark. It might seem that there is an asymmetry here, since Player I always selects a function on the first structure and always pebbles elements on the second one, and that instead, he should be allowed to choose on which structure he defines a function. However, this would not change the game in an essential way. The condition that Player II answers with a function defining the same multiset is very restrictive and makes it unnecessary to let Player I choose the structure first. In particular, I wins immediately if the primary parts of the two structures do not have the same cardinality. It should be noted that if two structures $\mathfrak{A}$ and $\mathfrak{B}$ are known to have the same number of elements, then also the C^k-game on $\mathfrak{A}$ and $\mathfrak{B}$ can be restricted such that Player I always chooses his sets in $\mathfrak{A}$ and pebbles elements of $\mathfrak{B}$, but never vice versa.

The moves in the T^k-game simulate the use of the multiset operations. However, it turns out that the T^k-game is equivalent to the C^k-game of Immerman and Lander. We prove this by way of two Lemmata.

Lemma 5.8. *Let $F(\bar{x})$ be a weight term in $T^k_{\infty\omega}$ of rank α such that $F^{\mathfrak{D}}(\bar{a}) \neq F^{\mathfrak{D}'}(\bar{b})$. Then*

(i) Player I wins the C^k-game on $(\mathrm{fin}(\mathfrak{D}), \bar{a})$ and $(\mathrm{fin}(\mathfrak{D}'), \bar{b})$. Furthermore if α is finite then he wins the game in at most α moves.

(ii) Player I wins the T^k-game on $(\mathfrak{D}, \bar{a})$ and $(\mathfrak{D}', \bar{b})$. Furthermore if α is finite then he wins the game in at most α moves.

Proof. Obviously, *(i)* implies *(ii)*. We prove *(i)* by induction on α, the case $\alpha = 0$ being trivial. Let $F^{\mathfrak{D}}(\bar{a}) \neq F^{\mathfrak{D}'}(\bar{b})$ for some term F of rank $\alpha > 0$. If $F = S(\Phi)$ for some operation S and set of terms Φ, then $G^{\mathfrak{D}}(\bar{a}) \neq G^{\mathfrak{D}'}(\bar{b})$ for at least one $G \in \Phi$; similarly, if $F = g(F_1, \ldots, F_m)$ then at least one subterm F_i separates $(\mathfrak{D}, \bar{a})$ and $(\mathfrak{D}', \bar{b})$.

Since the process of descending to proper subterms is well-founded, F contains at least one subterm separating $(\mathfrak{D}, \bar{a})$ and $(\mathfrak{D}', \bar{b})$ which either is of rank zero, in which case we are done, or of the form

$$\Gamma_{\bar{y}}(G(\bar{x}, \bar{y}) : H(\bar{x}, \bar{y}) = 1)$$

where G and H have ranks $< \alpha$. For ease of notation, we assume that $\bar{x}$ and $\bar{y}$ are disjoint tuples of variables among $x_1, \ldots, x_k$. In the case of finite α, the ranks of G and H are bounded by $\alpha - \ell$ where ℓ is the length of $\bar{y}$.

Thus, G and H define distinct multisets on the two structures:

$$\{\!\{G^{\mathfrak{D}}(\bar{a}, \bar{c}) : \bar{c} \in A^\ell, H^{\mathfrak{D}}(\bar{a}, \bar{c}) = 1\}\!\} \neq \{\!\{G^{\mathfrak{D}'}(\bar{b}, \bar{d}) : \bar{d} \in B^\ell, H^{\mathfrak{D}'}(\bar{b}, \bar{d}) = 1\}\!\}.$$

As a consequence there exists $r \in R$ such that

$$\#\{\bar{c} \in A^\ell : G^{\mathfrak{D}}(\bar{a}, \bar{c}) = r \wedge H^{\mathfrak{D}}(\bar{a}, \bar{c}) = 1\} \neq \#\{\bar{d} \in B^\ell : G^{\mathfrak{D}'}(\bar{b}, \bar{d}) = r \wedge H^{\mathfrak{D}'}(\bar{b}, \bar{d}) = 1\}.$$

This implies that there exist natural numbers $m_1, \ldots, m_\ell$ such that

$$\exists^{\geq m_1} y_1 \cdots \exists^{\geq m_\ell} y_\ell [G^{\mathfrak{D}}(\bar{a}, \bar{y}) = r \wedge H^{\mathfrak{D}}(\bar{a}, \bar{y}) = 1] \quad \text{but}$$

$$\textbf{not} \quad \exists^{\geq m_1} y_1 \cdots \exists^{\geq m_\ell} y_\ell [G^{\mathfrak{D}'}(\bar{b}, \bar{y}) = r \wedge H^{\mathfrak{D}'}(\bar{b}, \bar{y}) = 1]$$

(or vice versa). Player I wins by the following strategy: in his first ℓ moves he selects appropriate sets $A_1, \ldots, A_\ell \subseteq A$ of cardinalities $m_1, \ldots, m_\ell$ so that $G^{\mathfrak{D}}(\bar{a}, \bar{c}) = r$ and $H^{\mathfrak{D}}(\bar{a}, \bar{c}) = 1$ for the tuples $\bar{c} = c_1, \ldots, c_\ell$ with $c_i \in A_i$. By induction on ℓ it follows easily that whatever sets $B_1, \ldots B_\ell \subseteq B$ are chosen by Player II in these first ℓ moves, Player I can pebble elements $d_1, \ldots, d_\ell$ such that $G^{\mathfrak{D}'}(\bar{b}, \bar{d}) \neq r$ or $H^{\mathfrak{D}'}(\bar{b}, \bar{d}) \neq 1$. Since both G and H have ranks $< \alpha$, the induction hypothesis implies that Player I wins the remaining game, and, in the case of finite α, that he wins the remaining game in $\alpha - \ell$ moves. □

Lemma 5.9. *If Player II wins the C^k-game on $(\mathrm{fin}(\mathfrak{D}), \bar{a})$ and $(\mathrm{fin}(\mathfrak{D}'), \bar{b})$, then she also wins the T^k-game on $(\mathfrak{D}, \bar{a})$ and $(\mathfrak{D}', \bar{b})$.*

Proof. For fixed structures $\mathfrak{D}$, $\mathfrak{D}'$, the positions in both games are given by the tuples $\bar{a}, \bar{b}$ of pebbled elements. Since the winning conditions of the two games are identical it suffices to show the following: Suppose that Player II has a winning strategy for the C^k-game from position $(\bar a, \bar b)$. Then Player II has a strategy for *one move of the T^k-game* from position $(\bar a, \bar b)$ to reach a position from which she again has a *winning strategy for the C^k-game.* It then follows that also in the T^k-game, Player II can forever maintain the condition that the pebbled elements define a partial isomorphism between the primary parts.

Suppose that Player I, in the T^k-game from position $(\bar a, \bar b)$, starts by selecting pebbles $j_1, \dots, j_\ell$ and defining a function $f : A^\ell \to R$. By the assumption, Player II wins the C^k-game from $(\bar a, \bar b)$. Thus $(\mathrm{fin}(\mathfrak{D}), \bar a) \equiv_{C^k_{\infty\omega}} (\mathrm{fin}(\mathfrak{D}'), \bar b)$. By Theorem 5.4, this implies that, for every $C^k_{\infty\omega}$-equivalence class C and every $j \le k$, we have that

$$\#\{c \in A : (\mathrm{fin}(\mathfrak{D}), \bar a\tfrac{c}{j}) \in C\} = \#\{d \in B : (\mathrm{fin}(\mathfrak{D}'), \bar b\tfrac{d}{j}) \in C\}.$$

Repeating the argument, we get that for every equivalence class C and every $\boldsymbol{j} = j_1, \dots, j_\ell$

$$\#\{\bar c \in A^\ell : (\mathrm{fin}(\mathfrak{D}), \bar a\tfrac{\bar c}{\boldsymbol{j}}) \in C\} = \#\{\bar d \in B^\ell : (\mathrm{fin}(\mathfrak{D}'), \bar b\tfrac{\bar d}{\boldsymbol{j}}) \in C\}.$$

Thus, there exists a bijection $\pi : A^\ell \to B^\ell$ such that for all $\bar c \in A^\ell$

$$(\mathrm{fin}(\mathfrak{D}), \bar a\tfrac{\bar c}{\boldsymbol{j}}) \equiv_{C^k_{\infty\omega}} (\mathrm{fin}(\mathfrak{D}'), \bar b\tfrac{\pi\bar c}{\boldsymbol{j}}).$$

Now, Player II defines $g : B^\ell \to R$ as $g := f \circ \pi$, and, if Player I pebbles $\bar d \in B^\ell$, she answers with the unique tuple $\bar c \in A^\ell$ such that $\pi\bar c = \bar d$. The resulting positions are in the same $C^k_{\infty\omega}$-equivalence class, so Player II has again reached a winning position. ☐

Thus, we have established the following result.

Theorem 5.10. *Let $\mathfrak{D} = (\mathfrak{A}, \mathfrak{R}, W)$ and $\mathfrak{D}' = (\mathfrak{B}, \mathfrak{R}, W')$ be structures in $M_\Upsilon(\mathfrak{R})$ and $\bar a$ and $\bar b$ be ℓ-tuples of elements of $\mathfrak{A}$ and $\mathfrak{B}$, respectively. The following are equivalent*

(i) Player II wins the T^k-game on $(\mathfrak{D}, \bar a)$ and $(\mathfrak{D}', \bar b)$.

(ii) $(\mathfrak{D}, \bar a)$ and $(\mathfrak{D}', \bar b)$ are $T^k_{\infty\omega}$-equivalent.

(iii) Player II wins the C^k-game on $(\mathrm{fin}(\mathfrak{D}), \bar a)$ and $(\mathrm{fin}(\mathfrak{D}'), \bar b)$.

(iv) $(\mathrm{fin}(\mathfrak{D}), \bar a)$ and $(\mathrm{fin}(\mathfrak{D}'), \bar b)$ are $C^k_{\infty\omega}$-equivalent.

5.3 Invariants

The descriptions of $L^k_{\infty\omega}$- or $C^k_{\infty\omega}$-equivalence in terms of the k-pebble games give rise to invariants that represent in a compact way, by means of an *ordered finite structure*, the complete $L^k_{\infty\omega}$- or $C^k_{\infty\omega}$-theory of a given finite structure.

The first such invariants were found by Abiteboul and Vianu [2]. They were formulated in terms of computability by relational machines rather than $L^k_{\infty\omega}$-definability, but the notions are very closely related. With these invariants, Abiteboul and Vianu could prove that the logics FP and PFP coincide (with respect to expressive power) if and only if PTIME = PSPACE. We refer to [17] for a very nice exposition in terms of $L^k_{\infty\omega}$-equivalence.

Invariants for $C^k_{\infty\omega}$-equivalence have been defined in [24] and have been extensively studied by Otto [48, 49] who used them to prove a number of results on the structure of fixed point logic with counting, on the relationship of (FP + C) with other logics and on the canonization problem with respect to $C^2_{\infty\omega}$-equivalence.

We give an informal description of $C^k_{\infty\omega}$-invariants. For k-tuples $\bar{a}, \bar{a}'$ from a fixed structure $\mathfrak{A}$, we write $\bar{a} \sim \bar{a}'$ to denote that $(\mathfrak{A}, \bar{a})$ and $(\mathfrak{A}, \bar{a}')$ are $C^k_{\infty\omega}$-equivalent. We write $[\bar{a}]$ for the $\sim$-equivalence class of $\bar{a}$, also called the $C^k_{\infty\omega}$-type of $\bar{a}$.

The desired $C^k_{\infty\omega}$-invariant of a structure $\mathfrak{A}$ has the form

$$I^k(\mathfrak{A}) = (\mathfrak{B}, v_1, \dots, v_k)$$

where $\mathfrak{B} = (A^k/\sim, \prec, \dots)$ is an *ordered* structure over the set of $C^k_{\infty\omega}$-equivalence classes in A^k, and where weight functions $v_j : (A^k/\sim) \longrightarrow \mathbb{N}$ associate with every type $[\bar{a}]$ the number

$$v_j([\bar{a}]) := \#\{b \in A : \bar{a} \sim \bar{a}\tfrac{b}{j}\}.$$

With the game characterization of $C^k_{\infty\omega}$-equivalence it can be shown that both $\sim$ and a total order $\prec$ on $A^k/\sim$ (which is a pre-order on A^k) can be inductively defined. One starts with an arbitrary ordering $\prec_0$ of the atomic types in k variables. At every stage a pre-order $\prec_i$ on A^k is defined such that the associated equivalence relation $\sim_i$ (i.e. $\bar{a} \sim_i \bar{a}'$ iff neither $\bar{a} \prec_i \bar{a}'$ nor $\bar{a}' \prec_i \bar{a}$) describes that Player II can maintain her winning condition for at least i moves. The refinement step can be derived from Theorem 5.3: $\bar{a} \prec_{i+1} \bar{a}'$ if either $\bar{a} \prec_i \bar{a}'$, or $\bar{a} \sim_i \bar{a}'$ and the following condition holds:

> For the sequence $C_1 \prec_i C_2 \prec_i \cdots \prec_i C_r$ of $\sim_i$-equivalence classes, there exist $m \le r$ and $j \le k$ such that $\#\{b \in A : \bar{a}\tfrac{b}{j} \in C_m\} < \#\{b \in A : \bar{a}'\tfrac{b}{j} \in C_m\}$ and for all pairs $(\ell, i) <_{\text{lex}} (m, j)$ we have that $\#\{b \in A : \bar{a}\tfrac{b}{i} \in C_\ell\} = \#\{b \in A : \bar{a}'\tfrac{b}{i} \in C_\ell\}$.

Note that this refinement process is a variant of the colour refinement method leading to the stable colouring of a graph (see Example 4.23).

It follows from this description that the limits $\prec$ and $\sim$ of this inductive process are definable in (FP + C). In fact, a weaker logic is sufficient, namely fixed point logic together with a simple form of *cardinality comparison* which is captured by the so-called Rescher quantifier.

Definition 5.11. The *Rescher quantifier* is a generalized quantifier which combines two given formulae together, binding a single variable in each of the two formulae. From $\psi(x, \bar{z})$ and $\varphi(y, \bar{z})$, the new formula [Resch xy $\psi(x, \bar{z}), \varphi(y, \bar{z})$] is formed. Its semantics is defined by the equivalence

$$\models [\text{Resch } xy\ \psi(x, \bar{z}), \varphi(y, \bar{z})] \longleftrightarrow \Big(\#_x[\psi(x, \bar{z})] < \#_y[\varphi(y, \bar{z})]\Big).$$

We write FP[Resch] for the logic obtained by adjoining the Rescher quantifier to FP.

Besides the relation $\sim$ (for equality), $\prec$ (for the linear order), and the already described weight functions $v_1, \dots, v_k$, the structure $I^k(\mathfrak{A})$ is endowed with some additional relations to make sure that it encodes the entire $C^k_{\infty\omega}$-theory of $\mathfrak{A}$.

Atomic types: For every atomic type $t(x_1, \ldots, x_k)$ of vocabulary Υ_a, $I^k(\mathfrak{A})$ contains a unary relation $P_t := \{[\bar{a}] \in A^k/\sim : \mathfrak{A} \models t(\bar{a})\}$.

Reachability relations: For $j = 1, \ldots, k$, $I^k(\mathfrak{A})$ contains a binary relation $E_j([\bar{a}][\bar{a}'])$ which indicates that the type $[\bar{a}']$ can be obtained from $[\bar{a}]$ by changing the j-th coordinate. In other words,

$$E_j := \{([\bar{a}][\bar{a}']) : (\exists b \in A)\ \bar{a}\tfrac{b}{j} \sim \bar{a}'\} = \{([\bar{a}][\bar{a}\tfrac{b}{j}]) : b \in A\}.$$

Permutations: For every permutation $\sigma \in S_k$, we incorporate a binary relation $T_\sigma := \{([\bar{a}][\bar{a}']) : \sigma(\bar{a}) \sim \bar{a}'\}$ where $\sigma(a_1, \ldots, a_k) := a_{\sigma(1)}, \ldots, a_{\sigma(k)}$.

Obviously these additional relations are easily definable from $\mathfrak{A}$ and $\sim$. Further, it should be noted that there is some redundancy in this description in the sense that some relations are definable from others.

We can summarize the result on $C^k_{\infty\omega}$-invariants as follows.

Theorem 5.12. [24, 48, 49] *For every k and every finite relational vocabulary Υ_a, there exists a function I^k associating with every structure $\mathfrak{A} \in \mathrm{Fin}(\Upsilon_a)$ the $C^k_{\infty\omega}$-invariant $I^k(\mathfrak{A}) = (\mathfrak{B}, v_1, \ldots, v_k)$ such that the following hold:*

(i) The mapping $\mathfrak{A} \mapsto \mathfrak{B}$ is definable in FP[Resch].

(ii) For every $j \leq k$, the weight function $v_j : (A^k/\sim) \longrightarrow \mathbb{N}$ is definable from $\mathfrak{A}$ by a counting term $v(\bar{x}) = \#_y[\varphi(\bar{x}, y]$ with $\varphi \in$ FP[Resch].

(iii) $\mathfrak{A}$ and $\mathfrak{A}'$ are $C^k_{\infty\omega}$-equivalent if and only $I^k(\mathfrak{A}) \cong I^k(\mathfrak{A}')$.

Corollary 5.13. [24, 48] *For every class $\mathcal{K} \subseteq \mathrm{Fin}(\Upsilon_a)$, the following are equivalent*

(i) $\mathcal{K}$ is definable in (FP + C).

(ii) For some $k \in \mathbb{N}$, $\{I^k(\mathfrak{A}) : \mathfrak{A} \in \mathcal{K}\}$ is decidable in polynomial time.

Since the distinguishing power of the infinitary term calculus $T^k_{\infty\omega}$ can be reduced to $C^k_{\infty\omega}$-inequivalence of the corresponding finite structures, we obtain a notion of $T^k_{\infty\omega}$-invariants. It turns out that the $T^k_{\infty\omega}$-invariant $J^k(\mathfrak{D})$ of an arithmetical structure $\mathfrak{D}$ can be represented by a single natural number, and that J^k actually is an FP*-definable global function.

Theorem 5.14. *For every k and every vocabulary Υ of arithmetical structures there exists a numerical invariant $J^k : M_\Upsilon[\mathfrak{N}] \to \mathbb{N}$ with the following properties*

(i) J^k is FP**-definable.*

(ii) For all $\mathfrak{D}, \mathfrak{D}' \in M_\Upsilon[\mathfrak{N}]$

$$\mathfrak{D} \equiv_{T^k_{\infty\omega}} \mathfrak{D}' \ Longleftrightarrow\ J^k(\mathfrak{D}) = J^k(\mathfrak{D}').$$

We sketch a proof. From Theorem 5.10 we know that $\mathfrak{D}$ and $\mathfrak{D}'$ are $T^k_{\infty\omega}$-equivalent, if and only if the corresponding finite structures $\text{fin}(\mathfrak{D})$ and $\text{fin}(\mathfrak{D}')$ are $C^k_{\infty\omega}$-equivalent. We cannot directly use the invariant $I^k(\text{fin}(\mathfrak{D}))$, due to the infinite vocabulary of $\text{fin}(\mathfrak{D})$. However, an inductive process, similar the the one defined above, can be used to work directly with $\mathfrak{D}$, rather than with $\text{fin}(\mathfrak{D})$. It is obvious that FP[Resch] and the simple applications of counting needed for defining the weight functions can be simulated in FP* with secondary part $\mathfrak{N}$. Further an ordering (or pre-ordering) on the primary part induces a ranking (or pre-ranking) of points: just assign to a point the number of smaller points. We thus obtain an FP* definable function, mapping every $\mathfrak{D} \in M_\Upsilon[\mathfrak{N}]$ to a ranked arithmetical structure that characterizes $\mathfrak{D}$ up to $T^k_{\infty\omega}$-equivalence. Finally we can use the same techniques as in the proof of the Coding Lemma in the previous section to encode this structure by a natural number.

With these invariants, we easily get a converse for Proposition 4.25 for the case that $\mathfrak{N}$ = PTA.

Theorem 5.15. *A class $\mathcal{K} \subseteq \text{Fin}(\Upsilon_a)$ is* FP**-definable over PTA, if and only if $\mathcal{K}$ is* (FP + C)*-definable.*

Proof. The only-if direction has already been established. Suppose $\mathcal{K}$ is (FP + C)-definable. This and the FP*-definability of J^k imply that the class $\{J^k(\mathfrak{A}_\mathfrak{N}) : \mathfrak{A} \in \mathcal{K}\} \subseteq \mathbb{N}$ is decidable in polynomial-time and therefore expressible by a basic PTA-predicate. Since J^k is FP*-definable, the result follows. □

6 Asymptotic probabilities

Among the most beautiful results in finite model theory are the limit laws (in particular 0-1 laws) for various logics and probability distributions (see [14] for a survey).

We consider similar questions for metafinite structures, with fixed secondary part. It turns out, that limit laws hold only in rather restricted cases. Nevertheless, it is interesting to investigate and classify these cases.

Probability distributions. Fix a vocabulary $\Upsilon = (\Upsilon_a, \Upsilon_r, \Upsilon_w)$ where Υ_a and Υ_w are finite. Furthermore, fix a Υ_r-structure $\mathfrak{R}$, together with a probability distribution ν on the universe R. Finally, fix for every $n \in \mathbb{N}$ a probability distribution, over the finite set of Υ_a-structures with universe $\mathbf{n} = \{0, \dots, n-1\}$. In this paper, μ_n will always be the uniform distribution, giving equal probability to all structures with universe $\mathbf{n}$.

We define, for every $n \in \mathbb{N}$, a measure λ_n on the space S_n of metafinite structures $\mathfrak{D} \in M_\Upsilon[\mathfrak{R}]$ whose primary part has universe $\mathbf{n}$. The measure is defined by means of the following experiment:

- The primary part $\mathfrak{A}$ of $\mathfrak{D}$ is chosen according to the distribution μ_n.
- For every function symbol $w \in \Upsilon_w$ and every tuple $\bar{a}$, the value $w^\mathfrak{D}(\bar{a})$ is selected according to distribution ν.

Thus the measure λ_n defined in this way on S_n is the product measure of the uniform distribution μ_n over the finite set of primary parts with the product of $\sum_{w\in\Upsilon_w} n^{\text{arity}(w)}$

copies of ν. We denote the sequence $\lambda_1, \lambda_2, \ldots$ by $\boldsymbol{\lambda}$. For any class $C \subseteq M_\Upsilon[\mathfrak{R}]$ of metafinite structures we let

$$\lambda_n(C) := \lambda_n(C \cap S_n).$$

We now can define the corresponding probabilities of a sentence φ in any logic L of metafinite structures as follows:

$$\lambda_n(\varphi) = \lambda_n(\{\mathfrak{D} \in S_n : \mathfrak{D} \models \varphi)\}.$$

If the limit $\boldsymbol{\lambda}(\varphi) = \lim_{n\to\infty} \lambda_n(\varphi)$ exists, we call it the *asymptotic probability of* φ. If this limit exists for every sentence of L, then we say that the *convergence law* holds for L with respect to $\boldsymbol{\lambda}$. If, in addition, every sentence has asymptotic probability either 0 or 1, we say that the *0-1 law* holds for L with respect to $\boldsymbol{\lambda}$.

There are also other, weaker notions of limit laws, such as the existence of *Cesaro limits*

$$\lim_{n\to\infty} (\lambda_1(\varphi) + \lambda_2(\varphi) + \cdots + \lambda_n(\varphi))/n$$

or the *weak convergence law* (introduced by Shelah who called it the very weak 0-1 law [53]), saying that

$$\lim_{n\to\infty} \lambda_{n+1}(\varphi) - \lambda_n(\varphi) = 0.$$

It is clear that already very little of arithmetic present in $\mathfrak{R}$ suffices to refute the convergence law. If $\mathfrak{R}$ contains the natural numbers and parity is definable, and if we have summation over multisets then we can say that the number of elements of $\mathfrak{A}$ is even. This holds even for the trivial situation that $\Upsilon_a = \Upsilon_w = \varnothing$.

Thus, the question whether a convergence law or a 0-1 law holds, is interesting only for rather limited secondary parts $\mathfrak{R}$. In the sequel, we consider classes of *simple metafinite structures*, with various cases of $\mathfrak{R}, \Upsilon$ and ν.

6.1 The uncountable case

It should be noted that $\lambda_n(\varphi)$ is not defined in all situations. In fact, if φ is infinitary, then the set $\{\mathfrak{D} \in S_n : \mathfrak{D} \models \varphi\}$ need not be measurable. We show this by means of an example (that uses the axiom of choice and the continuum hypothesis).

Proposition 6.1. *Let $\mathfrak{R} = (R, 0, 1, \frac{1}{2}, +, \cdot, \leq)$, where R is the real interval $[0, 1]$ and $+$ is addition modulo 1. Let $\Upsilon_a = \varnothing, \Upsilon_w = \{c\}$ where c is nullary. Then, even for $n = 1$ and all $k \geq 0$, there is no probability distribution on $[0, 1]$ under which every sentence of $L^k_{\infty\omega}$ defines a measurable subset of S_n and every singleton has probability 0.*

Proof. It is known that, on the basis of the axiom of choice and the continuum hypothesis, there exists no probability distribution on $[0, 1]$, giving probability 0 to singletons, such that all subsets of $[0, 1]$ are measurable.

It therefore suffices to show that for every set $X \subseteq [0, 1]$ there exists a sentence $\psi_X \in L^k_{\infty\omega}$ such that for every structure $\mathfrak{D} \in M_\Upsilon[\mathfrak{R}]$

$$\mathfrak{D} \models \psi_X \text{ if and only if } c^{\mathfrak{D}} \in X.$$

Every real number $r \in [0,1]$ can be approximated by sequences $(a_n)_{n\in\omega}$ and $(b_n)_{n\in\omega}$ of dyadic rational numbers (i.e. rationals whose denominators are powers of two) such that $a_n \leq r \leq b_n$ and

$$\lim_{n\to\infty} a_n = \lim_{n\to\infty} b_n = r.$$

Every dyadic rational in $[0,1]$ is representable by a basic weight term in our language. Thus, in $L^k_{\infty\omega}$, we can form the sentence

$$\varphi_r := \bigwedge_{n\in\omega} (a_n \leq c \wedge c \leq b_n)$$

expressing that that $c = r$. Now the sentence $\psi_X := \bigvee_{r\in X} \varphi_r$ asserts that $c \in X$, which is what we wanted to prove. $\square$

Even though atomic formulae over $\mathfrak{R}$ define very simple sets, measurability need not be preserved under unrestricted conjunctions and disjunctions available in $L^\omega_{\infty\omega}$. Fortunately, there exist reasonable conditions on a logic L and a secondary part $\mathfrak{R}$ such that all L-definable model subclasses in S_n are measurable.

Definition 6.2. Let $\mathfrak{R}$ be a structure over Υ_r and ν a probability distribution on R. We say that $\mathfrak{R}$ has measurable atoms with respect to ν if every (first-order) atomic formula $\varphi(z_1,\dots,z_t)$ of vocabulary Υ_r defines a measurable set, i.e. $\nu(\{\bar{u} \in R^t : \mathfrak{R} \models \varphi(\bar{u})\})$ is defined.

Proposition 6.3. *If $\mathfrak{R}$ has measurable atoms with respect to ν, then every $L^\omega_{\omega_1\omega}$-definable model class in S_n (for every n) is measurable with respect to λ_n.*

Proof. Fix a primary part $\mathfrak{A}$ with universe $\mathbf{n}$ and let $S(\mathfrak{A})$ be the set of structures $\mathfrak{D} \in S_n$ with primary part $\mathfrak{A}$. Since, for fixed n, there are only finitely many primary parts, it suffices to show that the set $\{\mathfrak{D} \in S(\mathfrak{A}) : \mathfrak{D} \models \psi\}$ is measurable for every fixed $\mathfrak{A}$ and every sentence $\psi \in L^\omega_{\omega_1\omega}$. Then $\{\mathfrak{D} \in S_n : \mathfrak{D} \models \psi\}$ is a finite union of measurable sets and thus measurable.

It suffices to prove the claim for the expansion of the structure $\mathfrak{A}$ with names for all elements of A. We therefore suppose without loss of generality, that every element of $\mathfrak{A}$ is an individual constant. On $S(\mathfrak{A})$, the logic $L^\omega_{\omega_1\omega}$ then admits the elimination of quantifiers and of all primary relation and function symbols, except the constants: Every quantifier $\exists x\beta$ is replaced by $\bigvee_{a\in A} \beta(a/x)$, every primary term by the name of its value and every primary atomic subformula $Q(\bar{a})$ by its truth value. Thus the given sentence ψ is equivalent to a quantifier-free sentence φ. Since weight terms $w(\bar{a})$ are random variables with respect to the distribution ν and since $\mathfrak{R}$ has measurable atoms with respect to ν, it follows that for every atomic formula $\alpha = P(w_1(\bar{a}_1),\dots,w_k(\bar{a}_k))$ that may occur in φ, the set $\{\mathfrak{D} : \mathfrak{D} \in S(\mathfrak{A}) \wedge \mathfrak{D} \models \alpha\}$ is measurable. Since the measurable sets are closed under complementation and under countable unions and intersections, the claim follows. $\square$

Examples. We now consider some specific examples for $\mathfrak{R}$, ν, Υ_a and Υ_w such that the existence of a convergence law or a 0-1 law for first-order logic can be easily reduced to known results in finite model theory. We write FO for first-order logic in the classical sense, and FO^* for its extension to first-order logic of metafinite structures.

One unary weight function into an uncountable linear order. Let $\mathfrak{R} = ([0,1],<)$ with the uniform (Lebesgue) measure on $[0,1]$, let Υ_a be an arbitrary finite relational vocabulary and let $\Upsilon_w = \{w\}$ with w unary.

For any metafinite structure $\mathfrak{D} = (\mathfrak{A}, \mathfrak{R}, \{w\}) \subseteq M_\Upsilon[\mathfrak{R}]$, the weight function w defines a partial order on A by

$$a < b \text{ iff } \mathfrak{D} \models w(a) < w(b).$$

If $\mathfrak{D}$ is chosen randomly, then almost surely $\mathfrak{D} \models \forall x \forall y \; w(x) \neq w(y)$, so $<$ is in fact a random total order on $\mathfrak{A}$. Replacing $w(x)$ by x we can translate every sentence $\psi \in \mathrm{FO}^*$ to a sentence $\varphi \in \mathrm{FO}$ such that, almost surely, $\mathfrak{D} \models \psi$ if and only if $(\mathfrak{A}, <) \models \varphi$.

The problem is thus reduced to a problem on a class of *random finite ordered structures.*

For specific results, we distinguish several cases according to the vocabulary Υ_a of the primary part:

$\Upsilon_a = \varnothing$**:** In this case the structures have the form $\mathfrak{D} = (\mathbf{n}, \mathfrak{R}, \{w\})$ and the reduction gives a pure linear order $(\mathbf{n}, <)$. It is well-known that no first-order sentence φ can distinguish between linear orders $(\mathbf{n}, <)$ and $(\mathbf{m}, <)$ if both n and m are larger than a constant n_0 that depends only on the quantifier-rank of φ. Thus, we have a 0-1 law for FO.

However, in logics with recursion, such as transitive closure logic or fixed point logic, the presence of a linear order suffices to express that the structure has an even number of elements, and we therefore do not have any convergence law for these stronger logics. The same applies to monadic second-order logic MSO.

Υ_a **is monadic:** Clearly, we no longer have a 0-1 law. The sentence

$$\forall x([\forall y \; w(x) \leq w(y)] \rightarrow Px)$$

expresses, that the elements with minimal weights satisfy P. This is true with probability $1/2$ in all cardinalities.

However, we still have the convergence law, because of the convergence law for the first-order logic of random monadic structures with a linear order. This results appears in [44] but is attributed there to Ehrenfeucht.

Υ_a **contains at least one binary predicate.** Here we have non-convergence, due to the result of Compton, Henson and Shelah [15], according to which, on the class of random ordered graphs there exist first-order sentences without an asymptotic probability.

Two unary functions into an uncountable linear order. For structures of the form $\mathfrak{D} = (\mathfrak{A}, \mathfrak{R}, \{v, w\})$ with two unary weight functions into $\mathfrak{R} = ([0,1],<)$, it is easy to see that we no longer have a 0-1 law. For instance, the sentence

$$\exists x \exists y (\forall z (v(x) \leq v(z) \land w(y) \leq w(z)) \land v(x) < w(y)),$$

expressing that the minimal v-weight is smaller than the minimal w-weight, is true with probability $1/2$ in all cardinalities. In fact, we don't even have the convergence law. With two weight functions we can almost surely interpret two-dimensional partial orders (i.e. the intersection of two linear orders), and it is a result of Spencer [54], that there exist first-order sentences without asymptotic probabilities for k-dimensional partial orders, whenever $k \geq 2$.

Field of reals as secondary part. A different class of examples is obtained by taking for the secondary part the field of reals $\mathfrak{R} = (\mathbb{R}, +, \cdot, 0, 1)$. Here we have a 0-1 law for arbitrary relational Υ_a and arbitrary Υ_w. This might come as a surprise, but it is true for rather trivial reasons: Take any pair of basic weight terms $F(\bar{x}), G(\bar{y})$. Then almost surely either $\mathfrak{D} \models \forall \bar{x} \forall \bar{y}\ F(\bar{x}) = G(\bar{y})$ or $\mathfrak{D} \models \forall \bar{x} \forall \bar{y}\ F(\bar{x}) \neq G(\bar{y})$. Thus, the secondary part almost surely provides no information at all, so the 0-1 law holds whenever it holds on finite structures.

6.2 The countable case

The other interesting case is when the secondary part is countable. We may assume that its universe is the set of natural numbers. Then ν is given by a sequence p_n of nonnegative reals such that $\sum_{n=0}^{\infty} p_n = 1$ and $p_n = \nu(\{n\})$. We first show that one gets a strong form of non-convergence even in very simple cases. As above, $\boldsymbol{\lambda} = \lambda_1, \lambda_1, \ldots$ is the sequence of distributions induced by $\boldsymbol{\nu}$.

Definition 6.4. A distribution $\boldsymbol{\nu}$ *decreases rapidly* if $\lim_{n \to \infty} \frac{p_{n+1}}{p_n} = 0$.

An example of a rapidly decreasing distribution is the *Poisson distribution* $p_n := e^{-\mu} \mu^n / n!$ with the mean value μ.

Proposition 6.5. *Suppose that $\Upsilon_a = \Upsilon_r = \varnothing$ and Υ_w consists of one unary function name w, and let $\boldsymbol{\lambda}$ be induced by a rapidly decreasing distribution ν. Then the sentence*

$$\varphi = \exists x \forall y (y \neq x \to w(x) \neq w(y)).$$

has no asymptotic probability with respect to $\boldsymbol{\lambda}$. Even the Cesaro probabilities

$$\chi_k(\varphi) = [(\lambda_1(\varphi) + \cdots + \lambda_k(\varphi))]/k$$

do not converge.

Proof. We start with preliminary observations. Since $p_{n+1}/p_n = 0$ tends to 0, for every $c < 1$, there exists $m = m(c)$ such that $p_{n+1}/p_n < c$ for all $n > m$. Thus we may assume without loss of generality that $p_{n+1} < p_n/4$ for all n.

The sum $\sum_{j \geq n} p_j \Big/ p_n$ converges to 1 as n grows to infinity. Indeed, for every $\varepsilon > 0$, there exists a positive $c < 1$ such that $(c/(1-c)) < \varepsilon$. Let $m = m(c)$ be as above and suppose that $n > m$. We have

$$\sum_{j \geq n} \frac{p_j}{p_n} < \sum_{j \geq n} c^{j-n} = 1 + c/(1-c) < 1 + \varepsilon.$$

Finally, $e^{-2} < (1-p)^{1/p} < e^{-1}$ if $0 < p < 1/2$. Indeed, apply the Mean Value Theorem to the function $f(t) = -\log(1-t)$ on the interval $[0,p]$. There is a point $t \in (0,p)$ such

$$f(p) - f(0) = -\log(1-p) = (p-0)f'(t) = p/(1-t).$$

Since $p < p/(1-t) < p/(1-p) < p/(1-1/2) = 2p$, we have $p < -\log(1-p) < 2p$ and therefore $e^{-2p} < 1-p < e^{-p}$. Now raise the terms to power $1/p$.

Now we are ready to prove the proposition. The idea is as follows. Let $p = p_i$ and $M = \lfloor 1/p \rfloor$, so that $Mp \to 1$ and M grows much faster than i. We will check that the probabilities $\lambda_M(\exists!\, x[w(x) = i])$ converge to a positive number and therefore the probabilities $\lambda_M(\varphi)$ have a positive limes inferior. Further, let $N = \lfloor 1/\sqrt{p_{i+1}p_i} \rfloor$, so that $Np_i \to \infty$ and $Np_{i+1} \to 0$. We will check that the probabilities $\lambda_N(\exists x[w(x) > i])$ converge to zero and the probabilities $\lambda_N(\bigvee_{j \le i} \exists!\, x[w(x) = j])$ converge to zero. Therefore probabilities $\lambda_N(\varphi)$ converge to zero, because, for every n,

$$\lambda_n(\varphi) \le \lambda_n(\bigvee_{j \le i} \exists!\, x[w(x) = j]) + \lambda_n(\exists x[w(x) > i]).$$

Now let us do the necessary computations.

Part 1. Let n range over the interval $[M, 2M]$ and $\epsilon(p) = (1-p)^{1/p} - e^{-1}$, so that $\epsilon(p) = o(1)$ as p tends to 0. We have

$$\begin{aligned} \lambda_n(\varphi) &\ge \lambda(\exists! x[w(x) = i]) \ge np(1-p)^{n-1} > np[(1-p)^{1/p}]^{np} \\ &= np[e^{-1} + \epsilon(p)]^{np} > Mp[e^{-1} + \epsilon(p)]^{2Mp} = e^{-2} + o(1). \end{aligned}$$

It follows that

$$\liminf_k \chi_k(\varphi) \ge \liminf \chi_{2M}(\varphi) \ge \frac{1}{2M}[0 + M(e^{-2} + o(1))] > \frac{1}{18}.$$

Part 2. Let n range over $[N+1, 18N]$. We have

$$\begin{aligned} \lambda_n(\exists x[w(x) > i]) &\le 18N \sum_{j=i+1}^{\infty} p_j \\ &\le 18 \frac{1}{\sqrt{p_i p_{i+1}}} \sum_{j=i+1}^{\infty} p_j \\ &= \frac{1}{\sqrt{p_i p_{i+1}}} p_{i+1}(1 + o(1)) \\ &= \sqrt{\frac{p_{i+1}}{p_i}}\,(1 + o(1)) = o(1). \end{aligned}$$

Further, let j range over natural numbers $\le i$. We have

$$
\begin{aligned}
\lambda_n(\bigvee_j \exists!\, x[w(x)=j]) &\leq \sum_j \lambda_n(\exists!\, x[w(x)=j]) \\
&= \sum_j np_j(1-p_j)^{n-1} \\
&< \sum_j 18Np_j(1-p_j)^N \\
&= 18N\sum_j p_j\left[\left(1-\frac{1}{p_j}\right)^{1/p_j}\right]^{Np_j} \\
&< 18N\sum_j p_j e^{-Np_j} \\
&\leq 18\sum_j Np_j e^{-Np_j}.
\end{aligned}
$$

Notice that $Np_j - Np_{j+1} > 1$ if $j < i$. Indeed, $Np_j > Np_i \geq \sqrt{p_i/p_{i+1}} > 2$ because every $p_{m+1} < p_m/4$. Further, $Np_{j+1} < Np_j/4$. Hence $Np_j - Np_{j+1} > (3/4)Np_j > 1$. Therefore

$$
\lambda_n(\bigvee_j \exists!\, x[w(x)=j]) \leq 18 \cdot \sum_{m=\lfloor Np_i \rfloor}^{\infty} me^{-m},
$$

which converges to 0 when i grows to infinity because the series $\sum_{m=0}^{\infty} me^{-m}$ is convergent and $Np_i \leq p_i/\sqrt{p_ip_{i+1}} = \sqrt{p_i/p_{i+1}} \to \infty$. Consequently, $\lambda_n(\varphi) = o(1)$ and therefore

$$
\liminf_k(\chi_k(\varphi)) \leq \chi_{18N}(\varphi) \leq \frac{1}{18N}\big([\sum_{m=1}^{N} \lambda_n(\varphi)] + 17N\, o(1)\big) \leq 1/18.
$$

□ □

However, there is a weaker form of limit law, introduced by Shelah, which is of interest for this case.

Definition 6.6. We say that a class of sentences L satisfies the *weak convergence law* with respect to $\boldsymbol{\lambda} = (\lambda_n)_{n\in\mathbb{N}}$ if for all $\psi \in L$ we have that

$$
\lim_{n\to\infty} \lambda_{n+1}(\psi) - \lambda_n(\psi) = 0.
$$

For instance, it has been proved by Shelah [53], that first-order logic satisfies the weak convergence law on ordered random graphs and also on a random binary function. We can prove a similar result for monadic classes of metafinite structures with an arbitrary countable secondary part.

Theorem 6.7. *Let $\mathfrak{R}$ be any structure with universe $\mathbb{N}$, endowed with an arbitrary probability distribution ν, and let Υ_a and Υ_w be unary. Then for the induced sequence $\boldsymbol{\lambda}$ of probability distributions, first-order logic satisfies the weak convergence law.*

Proof. Let $\mathfrak{D} = (\mathfrak{A}, \mathfrak{R}, W)$ and $\mathfrak{D}' = (\mathfrak{B}, \mathfrak{R}, W')$ be two structures in $M_\Upsilon[\mathfrak{R}]$. Recall that for $a \in A$ and $b \in B$,

$$(\mathfrak{D}, a) \sim_0 (\mathfrak{D}', b)$$

means that the function $p : a \mapsto b$ is a partial isomorphism from $\mathfrak{D}$ to $\mathfrak{D}'$, i.e. that a and b satisfy the same Υ_a-relations over $\mathfrak{A}$ and $\mathfrak{B}$, respectively, and that the weight functions map a and b to the same values of $\mathbb{N}$. For every $m \in \mathbb{N}$, we say that a $\sim_0$-equivalence class C is m-bounded, if $w^{\mathfrak{D}}(a) \leq m$ for all $w \in \Upsilon_w$ and $(\mathfrak{D}, a) \in C$.

The structures $\mathfrak{D}$ and $\mathfrak{D}'$ are k-equivalent, i.e. cannot be distinguished by formulae of quantifier depth k, if every $\sim_0$-equivalence class C contains the same number of elements in $\mathfrak{D}$ and $\mathfrak{D}'$, or more than k elements in both structures. This can be proved by a straightforward application of the Ehrenfeucht-Fraïssè game. For every $\varepsilon > 0$ take a large enough natural number m so that $\sum_{i=0}^{m} p_i \geq 1 - \varepsilon$. Given k, choose n_0 large enough such that for every $n > n_0$, a random $\mathfrak{D} \in S_n$ contains, with probability at least $1 - \varepsilon$, more than k elements in every m-bounded $\sim_0$-equivalence class.

The process of drawing a random structure $\mathfrak{D} \in S_{n+1}$ can be described as follows: first we choose a random structure $\mathfrak{D}' \in S_n$; then we add a new element a and determine at random the truth values of atoms Pa for $P \in \Upsilon_a$ and the values of the weight terms $w(a)$ for $w \in \Upsilon_w$. With probability at least $(1 - \varepsilon)^\ell$ (where $\ell = |\Upsilon_w|$), the $\sim_0$-equivalence class of a is m-bounded. As a consequence, if $n > n_0$, then $\mathfrak{D}$ and $\mathfrak{D}'$ differ by an element that almost surely belongs to a class with more than k representants in both structures. Thus, $\mathfrak{D}$ is almost surely k-equivalent to $\mathfrak{D}'$.

Since k was arbitrary, it follows that that for every first-order formula ψ

$$\lim_{n\to\infty} \lambda_n(\psi) - \lambda_{n+1}(\psi) = 0.$$

$\square$ $\square$

Remark. With the same argument, the weak convergence law also holds for $L^\omega_{\omega_1\omega}$.

Acknowledgements

We would like to thank David Harel, Martin Otto, Jurek Tyszkiewicz, Moshe Vardi and Victor Vianu for valuable comments and suggestions.

The results in Sect. 5 were strongly influenced by discussions with Martin Otto. Some of the results in Sect. 6 were contributed by or proved in collaboration with Jurek Tyszkiewicz [57].

References

[1] S. Abiteboul, R. Hull and V. Vianu, *Foundations of Databases*, Addison Wesley (1994).

[2] S. Abiteboul and V. Vianu, *Generic Computation and Its Complexity*, Proceedings of 23rd ACM Symposium on Theory of Computing (1991), 209–219.

[3] L. Adleman and K. Manders, *Computational complexity of decision problems for polynomials*, Proceedings of 16th IEEE Symposium on Foundations of Computer Science (1975), 169–177.

[4] L. Adleman and K. Manders, *Diophantine Complexity*, Proceedings of 17th IEEE Symposium on Foundations of Computer Science (1976), 81–88.

[5] S. Arora, C. Lund, R. Motwani, M. Sudan and M. Szegedy, *Proof verification and intractability of approximation problems*, Proceedings of 33rd IEEE Symposium on Foundations of Computer Science (1992), 210–214.

[6] J. Barwise, *On Moschovakis closure ordinals*, Journal of Symbolic Logic **42** (1977), 292–296.

[7] E. Börger, E. Grädel and Y. Gurevich, *The Classical Decision Problem*, Springer 1997.

[8] O. Belegradek, A. Stolboushkin and M. Taitslin, *Extended order-generic queries*, preprint 1996.

[9] M. Benedikt, G. Dong, L. Libkin and L. Wong, *Relational expressive power of constraint query languages* Proc. 15th ACM Symp. on Principles of Database Systems PODS (1996), 5–16.

[10] L. Blum, M. Shub, S. Smale, *On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines*, Bull. Amer. Math. Soc.**21** (1989), 1–46.

[11] E. Börger, *Annotated Bibliography on Evolving Algebras*, in: E. Börger (Ed.), Specification and Validation Methods, Oxford University Press (1995).

[12] J. Cai, M. Fürer and N. Immerman, *An Optimal Lower Bound on the Number of Variables for Graph Identification*, Proceedings of 30th IEEE Symposium on Foundations of Computer Science (1989), 612–617.

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

[14] K. Compton, *0-1 Laws in Logic and Combinatorics*, in: NATO Adv. Study Inst. on Algorithms and Order, I. Rival (Ed.), 1988, 353–383.

[15] K. Compton, C. Henson and S. Shelah, *Nonconvergence, undecidability and intractability in asymptotic problems*, Annals of Pure and Applied Logic **36** (1987), 207–224.

[16] P. Crescenzi and V. Kann, *A compendium of NP optimization problems*, Technical Report SI/RR-95/-02, Dipartimento di Scienze dell'Informazione, Universitá di Roma "La Sapienza" (1995).

[17] A. Dawar, *Feasible Computation through Model Theory*, PhD thesis, University of Pennsylvania (1993).

[18] M. de Rougemont, *The reliability of queries*, Proc. 14th ACM Symp. on Principles of Database Systems PODS (1995).

[19] H.-D. Ebbinghaus and J. Flum, *Finite Model Theory*, Springer 1995.

[20] R. Fagin, *Generalized first-order spectra and polynomial-time recognizable sets*, SIAM-AMS Proceedings **7** (1974), 43–73.

[21] R. Fagin, *Finite model theory – A personal retrospective*, Theoretical Computer Science **116** (1993), 3-31.

[22] E. Grädel and A. Malmström, *0-1 laws for recursive structures*, to appear in Archive of Mathematical Logic (1997).

[23] E. Grädel and K. Meer, *Descriptive Complexity Theory over the Real Numbers*, Proceedings of 27th ACM-Symposium on Theory of Computing, STOC '95, Las Vegas (1995). An expanded version appeared in: J. Renegar, M. Shub, S. Smale (Eds.), *Mathematics of Numerical Analysis: Real Number Algorithms*, AMS Lectures in Applied Mathematics **32** (1996), 381–403.

[24] E. Grädel and M. Otto, *Inductive Definability with Counting on Finite Structures*, in: Selected Papers, 6th Workshop on Computer Science Logic CSL 92, San Miniato 1992, Lecture Notes in Computer Science Nr. 702, Springer (1993), 231–247.

[25] J. Gross and T. Tucker, *Topological Graph Theory*, Wiley, New York (1987).

[26] S. Grumbach and J. Su, *Finitely representable databases*, Proceedings of 13th ACM Symposium on Principles of Database Systems (1994).

[27] Y. Gurevich, *Toward logic tailord for computational complexity*, in: M. M. Richter et al. (Eds), Computation and Proof Theory, Springer Lecture Notes in Mathematics Nr. 1104 (1984), 175–216.

[28] Y. Gurevich, *Logic and the Challenge of Computer Science*, in: E. Börger (Ed), Trends in Theoretical Computer Science, Computer Science Press (1988), 1–57.

[29] Y. Gurevich, *Evolving Algebras 1993: Lipari Guide*, in: E. Börger (Ed.), Specification and Validation Methods, Oxford University Press (1995).

[30] Y. Gurevich and S. Shelah, *Fixed Point Extensions of First Order Logic*, Annals of Pure and Applied Logic **32** (1986), 265–280.

[31] T. Hirst and D. Harel, *Completeness Results for Recursive Databases*, Journal of Computer and System Sciences, to appear. (Also: 12th ACM Symp. on Principles of Database Systems (1993), 244–252.)

[32] T. Hirst and D. Harel, *More about Recursive Structures: Zero-One Laws and Expressibility vs. Complexity*, Proceedings of 11th IEEE Symposium on Logic in Computer Science (1996).

[33] B. Hodgson and C. Kent, *A Normal form for Arithmetical Representation of NP-sets*, Journal of Computer and System Sciences **27** (1983), 378–388.

[34] N. Immerman, *Upper and lower bounds for first-order expressibility*, Journal of Computer and Systems Sciences **25** (1982), 86–104.

[35] N. Immerman, *Relational Queries Computable in Polynomial Time*, Information and Control **68** (1986), 86–104.

[36] N. Immerman, *Expressibility as a Complexity Measure: Results and Directions*, Proc. of 2nd Conf. on Structure in Complexity Theory (1987), 194–202.

[37] N. Immerman, *Descriptive and Computational Complexity*, in: J. Hartmanis (Ed.), Computational Complexity Theory, Proc. of AMS Symposia in Appl. Math. **38** (1989), 75–91.

[38] N. Immerman and E. Lander, *Describing Graphs: A First Order Approach to Graph Canonization*, in: A. Selman (Ed), Complexity Theory Retrospective. (In Honor of Juris Hartmanis), Springer, New York 1990, 59–81.

[39] J. Jones and Y. Matijasevich, *Register machine proof of the theorem of exponential diophantine representation of enumerable sets*, Journal of Symbolic Logic **49** (1984), 818–829.

[40] F. Kabanza, J. Stevenne and P. Wolper, *Handling Infinite Temporal Data*, to appear in Journal of Computer and System Sciences.A preliminary version appeared in Proceedings of 9th ACM Syposium on Principles of Database Systems (1990).

[41] P. Kanellakis, *Elements of Relational Database Theory*, in: J. van Leeuwen (Ed.), Handbook of Theoretical Computer Science, vol. B, North Holland, Amsterdam 1990, pp. 1073–1156.

[42] P. Kanellakis, G. Kuper and P. Revesz, *Constraint Query Languages*, Proceedings of 9th ACM Symposium on Principles of Database Systems (1990), 299–313.

[43] C. Kent and B. Hodgson, *An arithmetical characterization of NP*, Theoretical Computer Science **12** (1982), 255–267.

[44] J. Lynch, *Almost sure theories*, Annals of Mathematical Logic **18** (1980), 91–135.

[45] A, Malmström, *Logic and Approximation*, PhD thesis, RWTH Aachen (1996).

[46] Y. Matijasevich, *Hilbert's Tenth Problem*, MIT Press, Cambridge (1993).

[47] M. Otto, *Generalized Quantifiers for Simple Properties*, Proceedings of IEEE Symposium on Logic in Computer Science (1994), 30–39.

[48] M. Otto, *The Expressive Power of Fixed-Point Logic with Counting*, Journal of Symbolic Logic **61** (1996), 147–176.

[49] M. Otto, Bounded Variable Logic with Counting. A Study on Finite Models, Lecture Notes in Logic Nr. 9, 183 pages, 1997.

[50] C. Papadimitriou, *Computational Complexity*, Addison-Wesley (1994).

[51] C. Papadimitriou and M. Yannakakis, *Optimization, approximization and complexity classes*, Journal of Computer and System Sciences **43** (1991), 425–440.

[52] B. Poizat, *Deux ou trois choses que je sais de* L_n, Journal of Symbolic Logic **47** (1982), 641–658.

[53] S. Shelah, *The very weak zero-one law for random graphs with order and random binary functions*, preprint (1994).

[54] J. Spencer, *Nonconvergence in the theory of random orders*, Order **7** (1991), 341–348.

[55] A. Stolboushkin, *Towards recursive model theory*, preprint (1996).

[56] A. Stolboushkin and M. Taitslin, *Linear vs. order constraint queries over rational databases*, Proc. 15th ACM Symp. on Principles of Database Systems PODS (1996), 17–27.

[57] J. Tyszkiewicz, private communication.

[58] J. D. Ullman, *Database and Knowledge-Base Systems, Vol. I and II*, Computer Science Press (1989).

[59] L. Valiant, *The complexity of enumeration and reliability problems*, SIAM J. Computing **8** (1979), 410–421.

[60] M. Vardi, *Complexity of Relational Query Languages*, Proc. of 14th Symposium on Theory of Computing (1982), 137–146.