The Kolmogorov Expression Complexity of Logics

Jerzy Tyszkiewicz*

Mathematische Grundlagen der Informatik,
RWTH Aachen, D-52064 Aachen, Germany.
jurek@informatik.rwth-aachen.de

March 3, 1997

*Supported by the Polish KBN grant 2 P301 009 06 and by the German DFG.

Proposed running head: *Kolmogorov Expression Complexity*

Proofs:
Prior to sending proofs, please send an e-mail to
`jurek@informatik.rwth-aachen.de`
or send a fax to
(+49 241) 88 88 215.

I'm going to change my employer and do not know yet the new one. It is particularly important if the proofs should be urgent.

Keywords: Kolmogorov complexity, finite-model theory, limit laws fixed point logic, infinitary logic, descriptive complexity.

Abstract

The purpose of the paper is to propose a completely new notion of complexity of logics in finite-model theory. It is the Kolmogorov variant of the Vardi's *expression complexity.* We define it by considering the value of the Kolmogorov complexity $C(L[\mathfrak{A}])$ of the infinite string $L[\mathfrak{A}]$ of all truth values of sentences of L in $\mathfrak{A}$. The higher is this value, the more expressive is the logic L in $\mathfrak{A}$.

If $\mathcal{D}$ is a class of finite models, then the value of $C(L[\mathfrak{A}])$ over all $\mathfrak{A} \in \mathcal{D}$ is a measure of expressive power of L in $\mathcal{D}$. Unboundedness of $C(L[\mathfrak{A}]) - C(L'[\mathfrak{A}])$ for $\mathfrak{A} \in \mathcal{D}$ implies nonexistence of a recursive interpretation of L in L'. A version of this statement with complexities modulo oracles implies the nonexistence of any interpretation of L in L'. Thus the values $C(L[\mathfrak{A}])$ modulo oracles constitute an invariant of the expressive power of logics over finite models, depending on their real (absolute) expressive power, and not on the syntax.

We investigate our notion for fragments of the infinitary logic $\mathcal{L}^{\omega}_{\infty\omega}$: least fixed point logic (LFP) and partial fixed point logic (PFP). We prove a precise characterization of 0-1 laws for these logics in terms of a certain boundedness condition placed on $C(L[\mathfrak{A}])$. We get an extension of the notion of a 0-1 law by imposing an upper bound on the value of $C(L[\mathfrak{A}])$ growing not too fast with cardinality of $\mathfrak{A}$, which still implies inexpressibility results similar to those implied by 0-1 laws.

We also discuss classes $\mathcal{D}$ in which $C(\mathrm{PFP}^k[\mathfrak{A}])$ is very high. It appears that then PFP or its simple extension can define all the PSPACE subsets of $\mathcal{D}$.

1 Introduction

1.1 Kolmogorov Expressive Power

In a formal way, the notion of *Kolmogorov expressive power* (KE for short) of a Boolean query language L in a finite model $\mathfrak{A}$ has been defined in Tyszkiewicz (1995), by considering two values: the Kolmogorov complexity $C(\mathfrak{A})$ of the isomorphism type of $\mathfrak{A}$, and the number of bits of this description that can be reconstructed from truth values of all sentences of L in $\mathfrak{A}$. This value we denote by $I_L(\mathfrak{A})$. The closer it is to $C(\mathfrak{A})$, the more expressive is L. In this paper we consider another value, the *Kolmogorov expression complexity* $C(L[\mathfrak{A}])$, i.e., the Kolmogorov complexity of the sequence $L[\mathfrak{A}]$ of truth values of all sentences of L in $\mathfrak{A}$.

In this paper KE is used to refer to $C(L[\mathfrak{A}])$.

Intuitively, the value $C(L[\mathfrak{A}])$ expresses how much of the complete information about $\mathfrak{A}$ is really necessary to reconstruct the L-theory of $\mathfrak{A}$. If **it is not** the complete information about $\mathfrak{A}$, then certainly L loses some of the information about $\mathfrak{A}$. Indeed, we could then change $\mathfrak{A}$ a little, getting a new structure $\mathfrak{A}' \not\cong \mathfrak{A}$, in which the results of evaluation of all sentences from L are identical. So at least the information corresponding to the difference between the isomorphism types of $\mathfrak{A}$ and $\mathfrak{A}'$ is invisible for L. In turn, if all the information about $\mathfrak{A}$ is necessary to reconstruct the theory, then L describes $\mathfrak{A}$ up to isomorphism.

We can turn the above qualitative distinction into a quantitative one in two ways:

- By considering how much of the isomorphism type of $\mathfrak{A}$ can be reconstructed from results of query evaluation; formally it is $I_L(\mathfrak{A}) = C(\mathfrak{A}) - C(\mathfrak{A}|L[\mathfrak{A}])$. (In our notation $C(\mathfrak{A}|L[\mathfrak{A}])$ is the Kolmogorov complexity of the isomorphism type of $\mathfrak{A}$, assuming the L-theory of $\mathfrak{A}$ is given.)

- By considering how much of the Kolmogorov complexity $C(\mathfrak{A})$ of $\mathfrak{A}$ is reflected by the Kolmogorov complexity $C(L[\mathfrak{A}])$ of the L-theory of $\mathfrak{A}$.

Both methods give rise to stratification of expressive powers of logics with respect to a fixed finite $\mathfrak{A}$. The unit of measure for this kind of expressive power is *bit* in the sense of *Kolmogorov complexity.*

The first choice, made in Tyszkiewicz (1995), is more natural in the realm of database theory, since it reflects the natural intuition of retrieving information from a database by querying it with Boolean queries from L. The second works better, when we want to create an abstract tool to compare expressive powers of logics, which is our goal in this paper.

1.2 KE in the Picture of Finite Model Theory

A nice introduction to the finite-model theory, covering all that we need here, can be found in Fagin (1993). The book of Ebbinghaus and Flum (1995) is also worth recommending. The following definition is the key one for us:

Definition 1.1. For two logics L, L' we say that L' *is at least as expressive as* L *over finite models,* in symbols $L \leqslant_{fin} L'$, iff there exists a map (interpretation) $i : L \to L'$ such that $\varphi \in L$ is equivalent in all finite models to $i(\varphi)$. It can be easily relativized to $L \leqslant_{\mathcal{D}} L'$ for any class $\mathcal{D}$ of finite models, by requiring that the equivalence of φ and $i(\varphi)$ holds in members of $\mathcal{D}$ only. Then if such an $i : L \to L'$ exists, we call it a $\mathcal{D}$*-interpretation.*

If $L \leqslant_{\mathcal{D}} L'$ and $L' \leqslant_{\mathcal{D}} L$, we say that L *and* L' *are of equal expressive power in* $\mathcal{D}$, and denote it symbolically $L \equiv_{\mathcal{D}} L'$.

The above two notions are absolute. I.e., they do not even refer to the syntax of L and L' or encoding of the structures. These (in)equivalences can be proved or disproved by various methods. Some of them rely on creating an invariant of the expressive power of logics.

We will not define precisely what an invariant is. This would move us to the next level of abstraction, while we intend to deal with just a few particular notions. But we hope that the several examples we describe will allow the reader to get an impression of what an invariant is.

We will confine ourselves to sentences, hence a given logic L can be represented in a form of an infinite binary matrix, whose rows correspond to all sentences of L and columns to all finite structures in $\mathcal{D}$. A 1 in row n and column m means that $\mathfrak{A}_m \models \varphi_n$, and 0 the opposite.

	$\mathfrak{A}_0$	$\mathfrak{A}_1$	$\mathfrak{A}_2$	$\mathfrak{A}_3$	$\mathfrak{A}_4$	$\mathfrak{A}_5$	$\dots$
φ_0	0	1	0	1	1	0	$\dots$
φ_1	1	1	1	0	0	1	$\dots$
φ_2	1	1	0	1	1	1	$\dots$
φ_3	1	0	1	1	1	1	$\dots$
φ_4	1	1	0	0	0	0	$\dots$
φ_5	0	0	1	0	1	1	$\dots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Figure 1: Logic represented as a binary matrix

Now according to the well-known terminology from Vardi (1982), the *data complexity* is the computational complexity of rows of the matrix, which are treated as encodings of decision problems. The *expression complexity* is the computational complexity of the columns. The *combined complexity* is the computational complexity of the whole matrix.

The expression complexity and the combined complexity are syntax dependent, as noted already by Vardi (1982) — this observation is attributed there to Chandra. Indeed, two logics of the same absolute expressive power can differ with respect to their complexity, since one of them can allow more succinct representation of the same semantical properties. The data complexity depends in turn on how we encode structures. This is however less disturbing, because we can meaningfully compare logics for some fixed encoding.

Complexity-theoretic measures of expressiveness. From what we have just said it follows that among the three notions discussed by Vardi in his paper only the *data complexity* is an invariant (for each fixed encoding of structures as inputs separately). But this already suffices to create an extremely rich theory. In some cases the set of rows of the matrix coincides in a precise way with the set of all problems computable in some complexity class. The prototype for this method was the Fagin's famous result that existential second order definable properties of graphs are precisely all the NP computable ones, Fagin (1974). We say that Σ^1_1 *captures* NP. On ordered structures many other interesting logics capture certain complexity classes. A large part of this work has been summarized by Immerman (1989). All these methods, and the area of research they belong to, are referred to as *descriptive complexity.*

Limit laws are quite a different tool for measuring expressive power of logics. A typical theorem in the theory of limit laws, a so called 0-1 *law*, asserts that for every sentence of the logic under consideration, the fraction of structures in which it is true, among all structures of cardinality n in $\mathcal{D}$, tends either to 0 or to 1, as n tends to infinity. A *convergence law* holds iff the above fraction always approaches a limit, but not necessarily 0 or 1. A *very weak* 0-1 *law* introduced by Shelah (1996), asserts that the differences between fractions computed for n and $n+1$ approach 0 for every sentence of the logic[1].

Referring again to Vardi's typology and matrix in Fig. 1, we notice that limit laws are of *data* type, since they assert certain properties of the density of 1's in the rows of the matrix. But when compared to Vardi's approach, they use combinatorial/analytic properties of the rows, instead of complexity theoretic ones, to define the invariant.

Limit laws can be thus seen as another method to measure the expressive power of logics on classes of finite models. It offers however only 4 levels of expressiveness: 0-1 law, convergence and nonconvergence, and within the last there is a possibility of the very weak 0-1 law. Some other weak forms of convergence laws have also been considered in the literature, but the spectrum of possible results still remains very small.

In the case of purely relational structures and first order logic, the first 0-1 law has been proven independently by Glebskiĭ et al. (1969) and by Fagin (1972, 1976). Since then, many similar theorems have been proven for various logics and various classes of structures.

Other methods. It can be seen that the descriptive complexity is applicable when the expressive power of L in $\mathcal{D}$ is very high. Conversely, limit laws apply in situations, when this power is very low. The middle has been so far no man's land. Essentially all we know about problems for which neither limit laws not the descriptive complexity can be used, has been proven by showing, often in a very clever way, that certain particular problems can be expressed in certain particular logics and cannot in other ones. Some properties have become even

[1] Sometimes this law is called a *slow oscillation law*, which seems closer to what it really states.

standard "separators", like graph connectivity, which has been used to separate many fragments of Σ_1^1 from dual fragments of Π_1^1.

And again each *ad hoc* method is of the *data* type, since it asserts that some particular row does/does not appear in the matrix at all.

Kolmogorov expressive power. All the invariants we have discussed so far have been of the *data* type. And it is not surprising, since the absolute notion from Def. 1.1 is of the *data* type itself. But now a little surprise is that there is an invariant of the *expression* type. It is our KE, and is defined, let us remind, by looking at the Kolmogorov complexity of columns of the matrix. (The name KE stands for **K**olmogorov **E**xpression).

This notion is in our opinion a new natural measure of expressiveness, substantially different from the descriptive complexity and the limit laws. The next surprise is that, though KE is in a precise sense orthogonal to the methods of the descriptive complexity and of the limit laws, it has nontrivial connections to both. It appears that for some logics it "almost suffices" to give a necessary and sufficient condition of a 0-1 law, and for other logics it "almost suffices" to detect when they capture complexity classes.

Connections between KE and other methods. We show that for every recursive class $\mathcal{D}$ of finite structures, if for every k the Kolmogorov expression complexity $C(\mathrm{LFP}^k[\mathfrak{A}])$ of LFP^k, the k-variable fragment of least fixed point logic LFP, is almost surely bounded in $\mathcal{D}$ by a constant if and only if for each k the class $\mathcal{D}$ can be represented as a union of finitely many disjoint subclasses, and for each of them the 0-1 law for LFP^k holds. The "limit" version is an assertion that the LFP 0-1 law holds for a recursive class $\mathcal{D}$ if and only if the first order 0-1 law holds for $\mathcal{D}$ and $C(\mathrm{LFP}^k[\mathfrak{A}])$ is almost surely bounded in $\mathcal{D}$ by a constant for each k.

An immediate generalization, which amounts to imposing a bound on the expressible information, growing slowly enough with cardinality of structures, leads to a pleasant extension of the 0-1 laws, still allowing one to prove inexpressibility results.

KE overlaps nontrivially with the descriptive complexity as well. It appears that the sentences of PFP augmented with one Lindström quantifier can define all the PSPACE subsets of the class of all finite structures in which the bounded variable fragment PFP^k for some k has a sufficiently high (with respect to the size of structures) value of $C(\mathrm{PFP}^k[\mathfrak{A}])$. Moreover, the same can be achieved in any PSPACE computable class $\mathcal{D}$ of structures even without using any Lindström quantifier, if we require in addition that the PFP^k theory distinguishes any two structures in $\mathcal{D}$, and the latter condition can be seen as a "limit" assertion about KE. (This is one of the two possibilities which have led us to introducing KE.)

Hopes for the future. Many theorems in recursion theory, formal language theory, automata theory, etc., can be proven by applications of the Kolmogorov complexity tools, mainly of the incompressibility method; see e.g. the contents list of Li and Vitányi (1993). But the mentioned list contains only one problem from logic. A version of the Gödel's famous incompleteness theorem can be proven by a Kolmogorov complexity argument. But it seems unbelievable that the rest of logic is immune to applications of the Kolmogorov complexity. We hope that this paper is the first (but not the last!) step towards introducing this tool to finite-model theory and logic. We also hope that this will create a new area of applications of Kolmogorov complexity.

2 Notation and Definition of KE

2.1 Logics and Structures

A *signature* (typically σ) is a finite collection of relation symbols, each one with a fixed arity. We fix one σ and work exclusively with finite structures (typically $\mathfrak{A}$) over it. The universe set $|\mathfrak{A}|$ of $\mathfrak{A}$ is always an initial segment of natural numbers. Let Σ be the set of all isomorphism classes of finite σ-structures. Since isomorphic structures are logically indistinguishable, we often write $\mathfrak{A} \in \Sigma$, meaning that $\mathfrak{A}$ is a representative of one of the isomorphism classes in Σ. We extend this convention also to classes $\mathcal{D} \subseteq \Sigma$, which therefore can be understood both as sets of structures, closed under isomorphisms, and as sets of isomorphism classes. If it is particularly important that we mean the isomorphism class of $\mathfrak{A}$ and not $\mathfrak{A}$ itself, we use the symbol $[\mathfrak{A}]$.

The cardinality of any set X is denoted $|X|$; consequently, $\|\mathfrak{A}\|$ denotes the cardinality of $|\mathfrak{A}|$. If $\mathcal{D} \subseteq \Sigma$, then $\|\mathcal{D}\|$ stands for the set of cardinalities of structures in $\mathcal{D}$; i.e., $\|\mathcal{D}\|$ is the spectrum of $\mathcal{D}$.

We say that L is a recursive logic if the following requirements are met. First of all, the set of sentences of L is recursive. Secondly, there is a fixed recursive function $eval : L \times \Sigma \to \{0,1\}$ such that $eval(\varphi, \mathfrak{A}) = 1$ iff $\mathfrak{A} \models \varphi$.

Having a recursive logic L, we fix some recursive bijective enumeration $\ell : \mathbb{N} \to L$ of all sentences of L (note that the converse function ℓ^{-1} of ℓ is also recursive), and a recursive bijective encoding $\mathrm{enc} : \Sigma \to \mathbb{N}$ of isomorphism classes of structures in Σ as natural numbers. We write $\varphi[\mathfrak{A}] \in \{0,1\}$ for the truth value of a sentence $\varphi \in L$ in $\mathfrak{A} \in \Sigma$. The second requirement for L to be recursive is then equivalent to the existence of a recursive semantic function $\mathrm{eval} : \mathbb{N}^2 \to \{0,1\}$ such that $\varphi[\mathfrak{A}] = \mathrm{eval}(\ell^{-1}(\varphi), \mathrm{enc}(\mathfrak{A}))$. Sometimes we use the traditional logical notation $\mathfrak{A} \models \varphi$ instead of $\varphi[\mathfrak{A}] = 1$.

For functions we use the lambda notation, i.e., $\lambda x.f(x)$ is the name of the function, which for argument x assumes the value $f(x)$.

If $\mathfrak{A} \in \Sigma$, then $L[\mathfrak{A}] \in \{0,1\}^{\mathbb{N}}$ is the function (or, equivalently, the infinite binary sequence) $\lambda n.\ell(n)[\mathfrak{A}]$. The sequence $L[\mathfrak{A}]$ is an ordered version of the *L-theory of* $\mathfrak{A}$.

2.2 Kolmogorov Complexity

We recall briefly the main definitions and notions of Kolmogorov complexity, using notation from Li and Vitányi (1993).

Proviso. $\mathbb{N}$ is the set of nonnegative integers, identified with the set $\{0,1\}^*$ of finite binary strings, ordered first by length, and then lexicographically. Thus 0 is the empty word, but we prefer to denote it by ϵ. We will use $lh(x)$ to denote the length of the word x.

We often use the asymptotic notation, such as O "big oh", o "small oh", etc. $\log n$ throughout the paper stands for the greatest $m \in \mathbb{N}$ such that $2^m \leqslant n$. The symbol $\{0,1\}^{\leqslant\infty}$ stands for the union $\{0,1\}^{\mathbb{N}} \cup \{0,1\}^*$.

Definition 2.1. Let $\psi(\cdot,\cdot)$ be a partial recursive function

$$\psi : \{0,1\}^* \times \{0,1\}^{\leqslant\infty} \to \{0,1\}^{\leqslant\infty},$$

computed by a Turing machine M. This means that the two inputs for M are provided on two input tapes, and there is no problem if the second of them is infinitely long. The output is then written on a write-only output tape, and the machine either halts after writing some output, which is then finite, or computes forever. If it continues writing output forever, then the resulting infinite sequence is the output, otherwise the value of ψ is undefined.

The *Kolmogorov complexity of a string* $x \in \{0,1\}^{\leqslant\infty}$ *relative to a string* $y \in \{0,1\}^{\leqslant\infty}$ *via a decoding function* ψ is

$$C_\psi(x|y) = \min\{lh(z) : \psi(z,y) = x\}.$$

According to a widely used convention we assume $\min\emptyset = \infty$.

$C_\psi(x|y)$ says how many bits we must add to y in order to describe x uniquely, where the method of understanding descriptions is given by ψ.

Let $\phi(\cdot,\cdot)$ be a universal partial-recursive function $\{0,1\}^* \times \{0,1\}^{\leqslant\infty} \to \{0,1\}^{\leqslant\infty}$, computed by a Turing machine M. I.e., for every partial recursive $\chi : \{0,1\}^* \times \{0,1\}^{\leqslant\infty} \to \{0,1\}^{\leqslant\infty}$ computed by a Turing machine N there is $n \in \{0,1\}^*$, which can be determined from a description of N and such that for every $z \in \{0,1\}^*$ and every $y \in \{0,1\}^{\leqslant\infty}$ holds

$$\phi(\langle n,z\rangle, y) = \chi(z,y),$$

where $\langle \_,\_ \rangle$ denotes some fixed recursive pairing function on $\{0,1\}^*$.

Theorem 2.2. *For every partial-recursive function* $\psi : \{0,1\}^* \times \{0,1\}^{\leqslant\infty} \to \{0,1\}^{\leqslant\infty}$ *there exists a constant* c_ψ *such that for all* $x, y \in \{0,1\}^{\leqslant\infty}$

$$C_\phi(x|y) \leqslant C_\psi(x|y) + c_\psi.$$

□

The above theorem, called The Invariance Theorem, justifies the following definition:

Definition 2.3. The *Kolmogorov complexity of a string* $x \in \{0,1\}^{\leqslant\infty}$ *relative to a string* $y \in \{0,1\}^{\leqslant\infty}$ is defined as $C(x|y) = C_\phi(x|y)$.

The *Kolmogorov complexity of a string* $x \in \{0,1\}^{\leqslant\infty}$ is defined as $C(x) = C(x|\epsilon)$. (ϵ is the empty word.)

The point here is that the Kolmogorov complexity of a string depends on that string, and *the choice of the function* ϕ. Theorem 2.2 says that for every other possible choice the value of the complexity does not increase more than by an additive constant. Therefore $C(x|y)$ captures the intuitive notion of the shortest possible description of x given y, up to an additive constant. This means as well that the complexity is determined "up to an additive constant term", and we can take any universal partial-recursive function in place of ϕ.

The above two notions can be extended in a strightforward manner by allowing Turing machines to use a fixed oracle R. The resulting complexities are denoted $C^R(x|y)$ and $C^R(x)$.

Proposition 2.4. *Let* R *be any oracle. For every* $y \in \{0,1\}^{\leqslant\infty}$ *and arbitrary finite set* $X \subseteq \{0,1\}^{\leqslant\infty}$ *there exists* $x \in X$ *such that*

$$C^R(x|y) \geqslant \log|X|.$$

In particular, for $X = \{0,1\}^n$, *it means that for every* $y \in \{0,1\}^{\leqslant\infty}$ *and every* $n \in \mathbb{N}$ *there is* $x \in \{0,1\}^*$ *of length* n *and with* $C^R(x|y) \geqslant n$.

The second part of the above proposition is often rephrased as "there are incompressible strings", i.e., strings which cannot be described in any way shorter then the string itself. Such incompressible objects are often used as "difficult" inputs in lower bound proofs of various kinds. Many examples can be found in the book Li and Vitányi (1993). We should mention that the existence of incompressible strings can be proven in a nonconstructive way, only.

3 KE and Invariance Theorem

Definition 3.1. The *Kolmogorov complexity of a finite structure* $\mathfrak{A} \in \Sigma$ *modulo oracle* R is defined as $C^R(\mathfrak{A}) = C^R(\mathrm{enc}(\mathfrak{A}))$.

The *Kolmogorov expression complexity of* L *in* $\mathfrak{A}$ *modulo oracle* R is $C^R(L[\mathfrak{A}])$.

If we omit the oracle in the complexity, the oracle is empty.

It has been already shown in Tyszkiewicz (1995) that the definitions we have given are correct. I.e., the values of $C^R(\mathfrak{A})$ and $C^R(L[\mathfrak{A}])$ do not depend on the choices we have made: of the encoding function enc to represent finite structures as words, of the enumeration ℓ of sentences of L, etc., *by more than an additive constant.* (The proof there does not mention oracles, which are meaningless in the context of databases, considered there. But the proof with oracles is essentially identical.) On the other hand, the Kolmogorov complexity of strings can itself change by an additive constant, depending on the choice of the universal partial-recursive function, so the additional indeterminacy introduced by our choices does not spoil more than has been already spoiled by the indeterminacy of the Kolmogorov complexity itself.

The next theorem follows almost directly from the definitions, but it can be seen as one of the main results in the paper, therefore we give the proof. It establishes that KE is an invariant of expressive power of logics, or, in other words, that it is syntax-independent to the maximal extent possible for notions based on the Kolmogorov complexity.

Theorem 3.2 (Invariance). *Let L, L' be any two recursive logics and let $\mathcal{D} \subseteq \Sigma$.*

1. *Let R be an oracle. Suppose that $C^R(L'[\mathfrak{A}]) - C^R(L[\mathfrak{A}])$ is unbounded in $\mathcal{D}$. Then there is no $\mathcal{D}$-interpretation $i : L' \to L$ which is recursive w.r.t. R.*

2. *If $C^R(L'[\mathfrak{A}]) - C^R(L[\mathfrak{A}])$ is unbounded in $\mathcal{D}$ for every oracle R, then $L' \not\leq_{\mathcal{D}} L$.*

Proof. 1. Let us suppose to the contrary, that there exists a $\mathcal{D}$-interpretation $i : L' \to L$, recursive w.r.t. R. Let for $\mathfrak{A} \in \mathcal{D}$ there be a program p (i.e., an input for ϕ), which computes consecutive bits of $\lambda n.\ell(n)[\mathfrak{A}]$ forever, possibly accessing the oracle R. In the course of contradiction, we modify p increasing its length by a constant to get a program p', which computes consecutive bits of $\lambda n.\ell'(n)[\mathfrak{A}]$, possibly accessing oracle R. Indeed, for each $n \in \mathbb{N}$ the program p' first computes $\psi = i(\ell'(n))$, which can be done effectively w.r.t. R. ψ is already a sentence of L, and $\mathcal{D} \models \psi \leftrightarrow \ell'(n)$. Then p' computes further the value $m = \ell^{-1}(i(\ell'(n)))$ and simulates the computation of p until it prints 0 or 1 in the m-th cell of the output tape. This value is indeed $\psi[\mathfrak{A}] = \ell'(n)[\mathfrak{A}]$. Then p' outputs itself the same value, and starts considering $n + 1$.

Item 2. follows from Item 1. and two trivial observations: first that if every sentence of L' is expressible in L then there exists a $\mathcal{D}$-interpretation $i : L' \to L$, and second that this i is recursive w.r.t. some oracle R. ☐

Several comments are in order:

First, it is particularly important that Theorem 3.2 is independent of the computational complexity. I.e., it is independent of the most commonly used and studied invariant, based on the Vardi's data complexity. This suggests that some problems which are out of reach of the latter invariant can be resolved by applications of KE. E.g., we are able to prove inexpressibility results, like Corollary 6.7 in Section 6.2, without any assumptions about complexity of evaluation of sentences of L. Also, we hope that some theorems having only difficult proofs, so far, can be proven in a more natural way by use of KE combined with the incompressibility method.

However, we should not expect too much. If we remind ourselves that any result stating that $C(L'[\mathfrak{A}]) - C(L[\mathfrak{A}])$ is unbounded in $\mathcal{D}$ means roughly, that there are many L-sentences inexpressible in L' over $\mathcal{D}$, then it is quite probable that any proof of it must lead to more or less direct construction of inexpressible sentences, and thus to a substantially simpler proof of $L' \not\leq_{\mathcal{D}} L$.

But the second observation is that even if it is difficult to get new separation and inexpressibility results with KE, we can still profit a lot from it. E.g., the difference $C(L'[\mathfrak{A}]) - C(L[\mathfrak{A}])$ over $\mathfrak{A} \in \mathcal{D}$ gives a numerical estimate, to what extent L' is more expressive than L. And this value can be of interest even for logics which are already known to be of different expressive power. With it we can move from a black-and-white picture of the situation we have now, to a full grey-scale one.

The third comment is that it is certainly not the case that the converse of the Invariance Theorem holds. There are several uninteresting reasons for it, and some interesting ones, too. Let us name one of them. Another one will be presented in Theorem 5.1 below.

Let L be any recursive logic and R be any oracle. Let us denote by L^{Bool} the closure of L under Boolean operations, i.e., let L^{Bool} consist of all finite Boolean combinations of sentences from L. Then for every $\mathfrak{A} \in \Sigma$ the quantities $C^R(L[\mathfrak{A}])$ and $C^R(L^{Bool}[\mathfrak{A}])$ differ at most by an additive constant independent of $\mathfrak{A}$. The reason is that there is a simple algorithm reconstructing $L^{Bool}[\mathfrak{A}]$ from $L[\mathfrak{A}]$, and hence $C^R(L^{Bool}[\mathfrak{A}]) - C^R(L[\mathfrak{A}])$ is bounded. Boundedness of the other difference is obvious. Thus KE cannot distinguish between L and L^{Bool}. The same effect appears if we consider a query language L in which queries are defined as sequences of accepting devices, such as Boolean circuits, which are given separately for structures of each cardinality. Then the nonuniform and various uniform versions of L cannot be separated by means of KE. Indeed, we consider the theories for different structures separately, therefore it is irrelevant what is the complexity of constructing the circuits as a function of size of structures.

So the hopes expressed in the two previous comments must be necessarily limited.

The fourth and final comment is about a serious drawback of KE, that the quantity $C(L[\mathfrak{A}])$ we are speaking about is noncomputable, like most of other kinds of Kolmogorov complexity. However, Theorem 3.2 requires only existential premises. E.g., the almost sure values of $I_L(\mathfrak{A})$ for several classes of finite models, which in these cases are close to the almost sure values of $C(L[\mathfrak{A}])$, have been established in Tyszkiewicz (1995). And this already allows one to apply Theorem 3.2.

Proposition 3.3. *For any recursive logic L there exists a constant d such that for all $\mathfrak{A} \in \Sigma$*

$$C(L[\mathfrak{A}]) \leqslant C(\mathfrak{A}) + d.$$

Proof. Let p be a program of length $C(\mathfrak{A})$ for ϕ such that $\phi(p, \epsilon) = \mathrm{enc}(\mathfrak{A})$. Then the program which computes forever consecutive bits of $L[\mathfrak{A}]$ performs the following algorithm: for each n it computes the value $\mathrm{eval}(n, \phi(p, \epsilon))$ and writes it in the n-th cell of the output tape. The only necessary observation is that enc is recursive and therefore can be described by a finite portion of code for ϕ. □

Thus the values of $C(L[\mathfrak{A}])$, irrespective of the logic L, can vary between 0 and $C(\mathfrak{A}) + d$, where the constant can depend on L.

4 FO, LFP, PFP and Infinitary Logic

4.1 Syntax and Semantics

In this paper we intend to deal with some specific logics: least fixed point logic LFP and partial fixed point logic PFP, and their bounded variable fragments. All of them are fragments of the infinitary logic $\mathcal{L}^{\omega}_{\infty\omega}$.

We assume that the reader is already familiar with first-order logic FO. LFP and PFP have been introduced to remedy an important weakness of FO—the lack of any recursion mechanism. E.g., FO fails to express the transitive closure of a graph, or that a graph is connected. The logics we will deal with have been introduced by Chandra (1982) and Chandra and Harel (1980), in a different notation. Both of them allow iterating an FO formula up to a fixed point. The difference is in the form of iteration.

Definition 4.1 (LFP and PFP).

First we define PFP, and then by restricting the syntax we get also LFP. PFP is an extension of first order logic FO, and we present the syntax and semantics of PFP by giving the only formula formation rule of it, which is missing in first order logic, together with the semantics of this construct.

Let $\varphi(R, \vec{x})$ be a PFP formula with k free variables over $\sigma' = \sigma \cup \{R\}$, where R is k-ary and does not occur in σ.

Then the formula $[\text{fp } \varphi(R), R(\vec{x})](\vec{x})$ is in PFP over σ; its semantics is as follows:

Let $\mathfrak{A}$ be a finite structure over σ, and let $\mathfrak{A}[R := S]$ be the structure over σ' resulting from $\mathfrak{A}$ by assigning to R the relation $S \subseteq |\mathfrak{A}|^k$. Then let $\Phi^0 = \emptyset$, and $\Phi^{i+1} = \{\vec{a} \in |A|^k \ : \ \mathfrak{A}[R := \Phi^i] \models \varphi(\vec{a})\}$.

The sequence Φ^i need not be convergent. **If it is so,** then the limit is denoted by Φ^∞ (and is equal to $\Phi^{2^{\|\mathfrak{A}\|^k}}$); otherwise the default value for Φ^∞ is $\emptyset$. Finally,

$$\mathfrak{A} \models [\text{fp } \varphi(R), R(\vec{x})](\vec{a}) \iff \vec{a} \in \Phi^\infty.$$

To get LFP we restrict the use of fp constructor: it can be applied only if all the occurrences of the relation variable R in φ are *positive,* i.e., under an even number of negations. This restriction ensures that the sequence Φ^i is ascending, so it must converge to a limit Φ^∞ (and in fact $\Phi^\infty = \Phi^{\|\mathfrak{A}\|^k}$).

The sets of formulas we have defined are denoted by PFP and LFP, respectively. FO^k, LFP^k and PFP^k stand for the sets of those formulas in FO, LFP and PFP, respectively, in which only k variables are used.

Definition 4.2 (Infinitary logic $\mathcal{L}^{\omega}_{\infty\omega}$). The logic $\mathcal{L}^{k}_{\infty\omega}$ is the closure of FO^k under infinitary conjunctions and disjunctions of arbitrary sets of formulae, and $\mathcal{L}^{\omega}_{\infty\omega} = \bigcup_{k\in\mathbb{N}} \mathcal{L}^{k}_{\infty\omega}$.

4.2 Expressive Power of LFP, PFP and Infinitary Logic

The following theorem summarizes some of the most important facts about expressive power of LFP and PFP.

Theorem 4.3. *1. For every $k \in \omega$* $\mathrm{LFP}^k \subseteq \mathrm{PFP}^k \leqslant_{fin} \mathcal{L}^k_{\infty\omega}$, *Dawar et al. (1995).*

2. *Over ordered finite structures* LFP *expresses precisely all* PTIME *properties, Immerman (1986) and Vardi (1982).*

3. *Over ordered finite structures* PFP *expresses precisely all* PSPACE *properties, Abiteboul and Vianu (1991a) and Vardi (1982).*

4. *On arbitrary structures neither* LFP *nor* PFP *can express that the cardinality of the universe is even, Chandra and Harel (1980).*

5. PFP $\equiv_{fin}$ LFP *if and only if* PTIME = PSPACE, *Abiteboul and Vianu (1991b).*

4.3 Normal Form

We are going to present a very powerful normal form theorem for PFP logic, proved first by Abiteboul and Vianu (1991b).

For every signature σ and every natural k there exists a sequence ϖ_k of LFP^{2k+2} formulae, which in arbitrary $\mathfrak{A} \in \Sigma$ define: a pre-ordering $\preccurlyeq_k$ of $|\mathfrak{A}|^k$, and a tuple of additional binary and unary relations $\vec{R}$ over $|\mathfrak{A}|^k$, such that the equivalence relation $\equiv_k$ on k-tuples of elements of $|\mathfrak{A}|$ defined by

$$\vec{x} \equiv_k \vec{y} \ :\Leftrightarrow\ \vec{x} \preccurlyeq_k \vec{y} \wedge \vec{y} \preccurlyeq_k \vec{x}$$

is a congruence with respect to the remaining relations (considered over domain $|\mathfrak{A}|^k$).

The quotient structure $\langle |\mathfrak{A}|^k, \preccurlyeq_k, \vec{R}\rangle / \equiv_k$ we call $\pi_k(\mathfrak{A})$. It appears that $\pi_k(\mathfrak{A})$ contains already all the information about truth of PFP^k sentences in $\mathfrak{A}$. Moreover, one can easily speak about $\pi_k(\mathfrak{A})$ in the language of $\mathfrak{A}$, since all of the relations of $\pi_k(\mathfrak{A})$ are definable in $\mathfrak{A}$.

Theorem 4.4 (Abiteboul and Vianu (1991b)). *Every formula of* PFP^k *over* $\mathfrak{A}$ *is expressible by a* PFP *formula over* $\pi_k(\mathfrak{A})$, *uniformly for all* $\mathfrak{A}$.

More precisely, for every k-ary formula φ in PFP^k *there exists a unary formula φ^{π_k} in* PFP *such that for arbitrary $\mathfrak{A} \in \Sigma$ the subset of $\|\mathfrak{A}\|^k$ defined by φ is equal to $\bigcup \varphi^{\pi_k}(\pi_k(\mathfrak{A}))$. (Recall that $\varphi^{\pi_k}(\pi_k(\mathfrak{A}))$ is a set of $\equiv_k$-equivalence classes of k-tuples over $|\mathfrak{A}|$.)*

Moreover, if φ is in LFP *then φ^{π_k} is in* LFP *too.* □

The converse of this theorem is obvious: whatever can be defined by a formula ψ of LFP about $\pi_k(\mathfrak{A})$ can be as well defined by an LFP formula of $\mathfrak{A}$ — we just have to replace signature symbols in φ by their LFP definitions over $\mathfrak{A}$. Certainly the same is true for PFP, too.

The construction of π_k, which reduces the question of definability over any possibly unordered structure $\mathfrak{A}$ to definability over the ordered invariant $\pi_k(\mathfrak{A})$ is a very powerful technique. It serves as a basis for very important theoretical results, e.g., the proof of the Item 5 of Theorem 4.3 is based on this approach. For the cardinality of $\pi_k(\mathfrak{A})$ we write $\|\pi_k(\mathfrak{A})\|$.

5 Basic Properties of KE of Extensions of First-Order Logic

The following results have been proven in Tyszkiewicz (1995), without oracles. However, the presence of them does not change anything in the proofs.

Theorem 5.1. *For any oracle R the Kolmogorov complexities $C^R(\mathrm{FO}^k[\mathfrak{A}])$, $C^R(\mathrm{LFP}^k[\mathfrak{A}])$ and $C^R(\mathrm{PFP}^k[\mathfrak{A}])$ are equal up to an additive constant, independent of $\mathfrak{A}$.* □

The essential reason is that the well-known result that $\mathrm{PFP}^k \leqslant_{fin} \mathcal{L}^k_{\infty\omega}$ has an effective proof, which actually yields a uniform translation of $\mathrm{PFP}^k[\mathfrak{A}]$ into $\mathrm{FO}^k[\mathfrak{A}]$, for each finite $\mathfrak{A}$.

Theorem 5.2. *Suppose that $\mathfrak{A}$ is a finite structure over σ, R any oracle and $\|\pi_k(\mathfrak{A})\| = n$. Then $C^R(\mathrm{LFP}^k[\mathfrak{A}]) \leqslant 4n \log n + O(n)$.* □

This theorem says to what extent the Abiteboul and Vianu invariant $\pi_k(\mathfrak{A})$ compresses the information about $\mathfrak{A}$. It can be proven by a careful analysis of the invariant, which allows removing relations definable from the remaining ones, and analyzing the complexity of those which are necessary.

6 KE as Extension of 0-1 Laws

6.1 0-1 Laws and Their Characterization in KE

The results below show that the notion of KE can be understood as an extension of the notion of a 0-1 law for sublogics of $\mathcal{L}^\omega_{\infty\omega}$. In order to present them, we need a slight modification of the notion of asymptotic probability often found in literature, to cover those $\mathcal{D}$, which do not have elements of arbitrary cardinality.

Definition 6.1. Let L be a logic, let $\mathcal{D} \subseteq \Sigma$ be a class of finite models.

1. Let for a sentence $\varphi \in L$ the value $\nu_n^{\mathcal{D}}(\varphi)$ be defined as

$$\frac{|\{[\mathfrak{A}] \in \mathcal{D} : \|\mathfrak{A}\| = n \ \&\ \mathfrak{A} \models \varphi\}|}{|\{[\mathfrak{A}] \in \mathcal{D} : \|\mathfrak{A}\| = n\}|},$$

   i.e., it is the fraction of the isomorphism classes of n element structures in Σ in which φ is true among all such isomorphism classes.

2. We say that a 0-1 *law* holds for L and $\mathcal{D}$, iff for every sentence φ of L the following limit exists and is equal to either 0 or 1:

$$\nu^{\mathcal{D}}(\varphi) = \lim_{\substack{n\to\infty \\ n\in\|\mathcal{D}\|}} \nu_n^{\mathcal{D}}(\varphi).$$

   In the case that $\|\mathcal{D}\|$ is finite, the 0-1 law holds by default. We emphasize the fact that our probabilities are defined by counting *isomorphism classes* of structures in $\mathcal{D}$. Thus we deal with so called *unlabeled uniform probabilities.*

3. We say that for a logic L and a class $\mathcal{D}$ of finite models a *mixed 0-1 law holds,* iff $\mathcal{D}$ can be represented as a finite disjoint union of classes of finite structures, and a 0-1 law holds for each block of this partition and L.

4. We say that a *convergence law holds for L and $\mathcal{D}$* iff the limit $\nu^{\mathcal{D}}(\varphi)$ exists for every $\varphi \in L$.

5. If we say that *almost every* $[\mathfrak{A}] \in \mathcal{D}$ *has property* φ, we mean that $\nu^{\mathcal{D}}(\varphi) = 1$. Yet another phrasing for it is φ *holds almost surely in* $\mathcal{D}$.

Let us make an observation that in the case when a mixed 0-1 law holds, for every sentence $\varphi \in L$, $\nu_n^{\mathcal{D}}(\varphi)$ is asymptotically equal to a sum of densities of some of the blocks of the partition of $\mathcal{D}$, in whole $\mathcal{D}$. Therefore if $\mathcal{D}$ is recursive it is not difficult to give a decidable property dependent on the size of structures, inexpressible in L. If $\|\mathcal{D}\| = \mathbb{N}$, this property can be even chosen to be of the form $\|\mathfrak{A}\| \equiv 0 \pmod{p}$ for some p.

In the following two theorems we drop the oracles from the complexities. We are going to speak about bounded values of the complexities, therefore adding/dropping an oracle does not make any difference. First what we can get in KE:

Theorem 6.2. *Suppose that $\mathcal{D}$ is a recursive class of finite models. Then the following statements are equivalent:*

1. *For every fixed k, there is a constant c such that $C(\mathrm{LFP}^k[\mathfrak{A}]) \leqslant c$ holds for almost every $\mathfrak{A} \in \mathcal{D}$.*

2. *For every fixed k, there is a constant c such that $\|\pi_k(\mathfrak{A})\| \leqslant c$ holds for almost every $\mathfrak{A} \in \mathcal{D}$.*

3. *For every fixed k, a mixed* 0-1 *law holds for* LFP^k *and $\mathcal{D}$, and the blocks of the partition of $\mathcal{D}$ are recursive.*

4. *For every fixed k, a mixed* 0-1 *law holds for $\mathcal{L}^k_{\infty\omega}$ and $\mathcal{D}$, and the blocks of the partition of $\mathcal{D}$ are recursive.*

Proof. $3 \Leftrightarrow 4$ has been shown in Tyszkiewcz (1993).

$2 \Rightarrow 1$ follows from Theorem 5.2.

$1 \Rightarrow 2$. Suppose to the contrary that $\|\pi_k(\mathfrak{A})\|$ assumes with asymptotic nonzero probability arbitrarily large values. Since $\pi_k(\mathfrak{A})$ is definable in LFP^{2k+2}, in this logic we can express the cardinality of $\pi_k(\mathfrak{A})$, as well. Therefore we get $C(\mathrm{LFP}^{2k+2}[\mathfrak{A}]) \geqslant C(\|\pi_k(\mathfrak{A})\|) - O(1)$. But, as $\|\pi_k(\mathfrak{A})\|$ can be arbitrarily large with asymptotic nonzero probability, $C(\mathrm{LFP}^{2k+2}[\mathfrak{A}])$ can be arbitrarily large too, which contradicts 1, and therefore finishes the proof of this implication.

$2 \Rightarrow 3$. Since $\|\pi_k(\mathfrak{A})\|$ is asymptotically bounded in $\mathcal{D}$, we can assume that it is bounded in $\mathcal{D}' \subseteq \mathcal{D}$, and $\mathcal{D} - \mathcal{D}'$ is asymptotically vanishing in $\mathcal{D}$. So $\pi_k(\mathfrak{A})$ assumes only finitely many values for $\mathfrak{A} \in \mathcal{D}'$. Let the division $\mathcal{D}' = \mathcal{D}'_1 \sqcup \ldots \sqcup \mathcal{D}'_m$ consist of all the equivalence classes of the equivalence relation $\{\langle \mathfrak{A}, \mathfrak{B} \rangle \mid \mathfrak{A}, \mathfrak{B} \in \mathcal{D}',\ \pi_k(\mathfrak{A}) \cong \pi_k(\mathfrak{B})\}$. Since for every $\varphi \in \mathrm{LFP}^k$ the value $\varphi[\mathfrak{A}]$ depends on $\pi_k(\mathfrak{A})$, only, each of these classes has a LFP^k 0-1 law. What remains to be done, is to spread $\mathcal{D}-\mathcal{D}'$ over $\mathcal{D}'_1, \ldots, \mathcal{D}'_m$, in order to get recursive $\mathcal{D}_1, \ldots, \mathcal{D}_m$, such that the 0-1 law for each of them still holds. We achieve this adding all the n-element structures of $\mathcal{D}-\mathcal{D}'$ to this $\mathcal{D}'_i$, which has the maximal number of n-element structures among $\mathcal{D}'_1, \ldots, \mathcal{D}'_m$, and i is minimal possible with that property. It is left for the reader to check that the resulting division is recursive, and that the requested 0-1 laws hold.

The proof of $3 \Rightarrow 2$ is similar to the proof of the main theorem in Tyszkiewicz (1993). The main technical result, which is implicitly stated there, and which we need here explicitly, is as follows.

Lemma 6.3. *For a uniform unlabeled probability distribution over a recursive class $\mathcal{D}$ of finite structures, such that $\|\pi_k(\mathfrak{A})\|$ is almost surely unbounded with respect to this distribution, there exists a* LINSPACE *computable function $u : \mathbb{N} \to \mathbb{N}$ satisfying*

$$\textit{for every } r > 1 \textit{ there is } n \in \mathbb{N} \textit{ such that } \quad \nu_n^{\mathcal{D}}(u(\|\pi_k(\mathfrak{A})\|) = r) > 2/3. \qquad (1)$$

Intuitively, this can be explained in the following way: the distribution we speak about has almost surely unbounded value of $\|\pi_k(\mathfrak{A})\|$, i.e. for every fixed m, $\lim_{n\to\infty} \nu_n^{\mathcal{D}}(\|\pi_k(\mathfrak{A})\| \leqslant m) = 0$. If we look at the probabilities of values of $\|\pi_k(\mathfrak{A})\|$, this means that asymptotically almost the whole mass of probability will pass structures with $\|\pi_k(\mathfrak{A})\| \leqslant m$. Again intuitively, this mass must then form a "wave" moving towards structures of greater values of $\|\pi_k(\mathfrak{A})\|$. Now if we rescale our picture, replacing $\|\pi_k(\mathfrak{A})\|$ by $u(\|\pi_k(\mathfrak{A})\|)$, this wave becomes very "concentrated": the condition says that for each r there is n such that at time n at least $2/3$ of the wave is located precisely at r.

Sketch of proof of Lemma 6.3. Define function $g : \mathbb{N} \to \mathbb{N}$ by the condition

$$g(n) = \left[\min\{m \ :\ \nu_m^{\mathcal{D}}(\|\pi_k(\mathfrak{A})\| > n) > 2/3\}\right]^k. \qquad (2)$$

g is recursive because $\mathcal{D}$ is recursive and because $\|\pi_k(\mathfrak{A})\|$ is almost surely unbounded. Let $g^{(m)}(n) = \underbrace{g \dots g}_{m}(n)$.

Note that $u : \mathbb{N} \to \mathbb{N}$ defined by

$$u(n) = \min\{r \;:\; g^{(r)}(1) \geqslant n\}$$

satisfies already (1), which is half of what we require. Indeed, for each $r > 1$ there is m satisfying $g^{(r-1)}(1) < m < g^{(r)}(1)$ (namely $m = \sqrt[k]{g^{(r)}(1)}$) such that the following conditions hold:

$$\nu_m^{\mathcal{D}}(\|\pi_k(\mathfrak{A})\| > g^{(r-1)}(1)) > 2/3, \tag{3}$$

which follows from (2), and

$$\nu_m^{\mathcal{D}}(\|\pi_k(\mathfrak{A})\| \leqslant g^{(r)}(1)) = 1,$$

which is obvious (there are only m^k k-tuples over an m element domain). Therefore

$$\nu_m^{\mathcal{D}}(u(\|\pi_k(\mathfrak{A})\|) = r) = \nu_m^{\mathcal{D}}(g^{(r-1)}(1) < \|\pi_k(\mathfrak{A})\| \leqslant g^{(r)}(1)) > 2/3.$$

Now, in order to improve u to a LINSPACE computable $\tilde{u}$, we repeat the same construction with any *space constructible* function $\tilde{g}$ which majorizes g defined by (2). Everything works for $\tilde{g}$ exactly as it has worked for g — only choosing m to satisfy (3) we have to take $m = \sqrt[k]{g\tilde{g}^{(r-1)}(1)}$. Now it is readily verified that $\tilde{u}$ resulting from this construction is LINSPACE computable. Indeed, in order to compute $\tilde{u}(n)$ we simulate a machine computing $\tilde{g}$ and witnessing its space constructibility on input 1, then on the output it has given on 1, and so on, as long as these computations fit in space $n-1$. When finally some step of the simulation requires more space, we know that the output is going to be at least n (by space constructibility), and we output the number of successful iterations plus one. $\square$

Now, turning back to the proof of Theorem 6.2, we proceed as follows:

Let us suppose to the contrary that $\|\pi_k(\mathfrak{A})\|$ is not almost surely bounded in $\mathcal{D}$.

In the division of $\mathcal{D}$, corresponding to the case with $2k+2$ variables, there must be a block $\mathcal{D}_j$ in which $\|\pi_k(\mathfrak{A})\|$ is almost surely unbounded, and whose sequence of probabilities in $\mathcal{D}$ does not tend to 0. Indeed, this is so since otherwise either $\|\pi_k(\mathfrak{A})\|$ would be almost surely bounded in $\mathcal{D}$, or else in some block $\mathcal{D}_i$ some of the properties $\|\pi_k(\mathfrak{A})\| \leqslant m$ for $m \in \mathbb{N}$, expressible in LFP^{2k+2}, would not have asymptotic probability 0 or 1.

In what follows, in order to derive a contradiction, it suffices to show that there is l such that $\mathcal{D}_j$ cannot be further partitioned into $\mathcal{D}_{j1}, \dots, \mathcal{D}_{jm}$ with a 0-1 law for LFP^l in each $\mathcal{D}_{ji}$.

Let us consider the function $u : \mathbb{N} \to \mathbb{N}$ satisfying (1) with respect to the uniform distribution $\nu^{\mathcal{D}_j}$ on $\mathcal{D}_j$. All the properties ψ_p defined as $u(\|\pi_k(\mathfrak{A})\|) \equiv 0 \pmod{p}$ for prime $p > 2$ are expressible in LFP^l for some fixed l.

Indeed, u is computable in LINSPACE as a *numeric* function, which means its input and output are written in the binary notation. But the cardinality of $\|\pi_k(\mathfrak{A})\|$ in structures is represented in the unary notation, i.e., the length of $\|\pi_k(\mathfrak{A})\|$ written in binary is logarithmic with respect to $\|\pi_k(\mathfrak{A})\|$. Therefore u in structures is computable even in LOGSPACE. Thus there is a single formula $\xi \in$ LFP, which computes $u(\|\pi_k(\mathfrak{A})\|)$, the argument and value being represented in $\|\pi_k(\mathfrak{A})\|$ — recall that LFP captures PTIME on ordered structures (Theorem 4.3 Item 2). Now it suffices to observe that all the tests ψ_p can be expressed in LFP^l for some fixed l.

According to the properties of u, the set of structures satisfying ψ_3 has no asymptotic probability on $\mathcal{D}_j$. So the structures in which ψ_3 is true and false, respectively, must be kept in separate blocks of the division of $\mathcal{D}_j$. But then ψ_5 has no asymptotic probability in the set of the structures in which ψ_3 is true. So the blocks have to be further split. Now immediate induction shows that no finite number of blocks suffices, and this yields a contradiction, which finishes the proof. □

The limit version is then the following.

Theorem 6.4. *Let us suppose that $\mathcal{D}$ is a recursive class of finite models and that the first order 0-1 law holds. Then the following statements are equivalent:*

1. *For every fixed k there is a constant c such that $C(\mathrm{LFP}^k[\mathfrak{A}]) \leqslant c$ for almost every $[\mathfrak{A}] \in \mathcal{D}$.*

2. *For every fixed k there is a constant c such that $\|\pi_k(\mathfrak{A})\| \leqslant c$ for almost every $\mathfrak{A} \in \mathcal{D}$.*

3. *A* 0-1 *law holds for* LFP *and $\mathcal{D}$.*

4. *A* 0-1 *law holds for $\mathcal{L}^\omega_{\infty\omega}$ and $\mathcal{D}$.*

Proof. The equivalences of 2, 3 and 4 have been proven in Tyszkiewicz (1993), while the equivalence of 1 and 2 can be shown exactly as in the previous proof. □

We have just shown that those 0-1 laws for first order logic which are accompanied by almost sure equality $C(\mathrm{FO}^k[\mathfrak{A}]) = O(1)$ for all k are exactly those, which extend to 0-1 laws for LFP and $\mathcal{L}^\omega_{\infty\omega}$. (Recall that $C(\mathrm{FO}^k[\mathfrak{A}])$ and $C(\mathrm{LFP}^k[\mathfrak{A}])$ are equal up to an additive constant independent of $\mathfrak{A}$, Theorem 5.1.) It should be mentioned that there are known 0-1 laws for first order logic with $C(\mathrm{FO}^4[\mathfrak{A}]) = C(\mathfrak{A}) - O(1)$ for all structures in $\mathcal{D}$. An example appears below. Thus 0-1 laws for first order logic are not tightly connected to KE, unlike those for LFP and PFP, since both extremes of the Kolmogorov expression complexity can be achieved for such classes.

Proposition 6.5. *There is a recursive class $\mathcal{D}$ of finite structures with a* 0-1 *law for first order logic and such that there is a constant c with $C(\mathrm{FO}^4[\mathfrak{A}]) \geqslant C(\mathfrak{A}) - c$ for all structures $\mathfrak{A} \in \mathcal{D}$.*

Proof. Let $\mathcal{D}$ be the class of all graphs with an underlying modular successor relation, i.e., structures $\mathfrak{A} = \langle |\mathfrak{A}|, R^{\mathfrak{A}}, S^{\mathfrak{A}} \rangle$, where $R^{\mathfrak{A}}$ is an arbitrary binary relation on $n = |\mathfrak{A}|$ and $S^{\mathfrak{A}}$ satisfies $S(i, i+1)$ for $i = 0, \dots, n-1$ and additionally $S^{\mathfrak{A}}(n, 0)$. It is then a result of Lynch (1980) that a 0-1 law holds for FO and $\nu^{\mathcal{D}}$. The equality $C(\mathrm{FO}^4[\mathfrak{A}]) = C(\mathfrak{A}) - O(1)$ for $\mathfrak{A} \in \mathcal{D}$ is easily seen. The reason is that we can describe each structure in $\mathcal{D}$ with 4 variables up to isomorphism, fixing one variable as a reference point and "walking around" with the remaining ones and saying for each pair of vertices if they are joined by an edge or not. $\square$

6.2 Low Value of KE as Extension of 0-1 Laws

Let us discuss, what kinds of inexpressibility results can be proven by tools provided by KE. The classical limit laws can be used for it: Generally, even very weak 0-1 laws exclude the possibility of expressing that $\|\mathfrak{A}\|$ is even, if $\|\mathcal{D}\| = \mathbb{N}$. One expects that theorems asserting a low, but not necessarily constant, value of KE for L should still lead to similar inexpressibility results for L. And they lead, but such inexpressibility results do not assert that some particular property is inexpressible, like the limit laws do. Instead, they assert that there is no interpretation of *all* properties from some family of them in L, similarly as mixed 0-1 laws do. The following theorem gives an example of such an inference, generalizing the observation we have made after the definition of the mixed 0-1 laws.

Of course, this is the place where the incompressibility method plays an important role. The intuitive idea is quite simple: we have a family of properties which, if expressible in a given logic L, would enforce $C(L[\mathfrak{A}])$ to be greater than in fact it can be. We will use the incompressibility argument to prove existence of structures in which this complexity would be indeed so large.

Theorem 6.6. *Let L be any recursive logic and let $\|\mathcal{D}\| = \mathbb{N}$. Let* DIV *be the logic consisting of sentences $(\|\mathfrak{A}\| \equiv 0 \pmod q)$ for $q \in \mathbb{N}$.*

1. *If for an oracle R holds*

$$\lim_{\substack{\|\mathfrak{A}\| \to \infty \\ \mathfrak{A} \in \mathcal{D}}} \log \|\mathfrak{A}\| - C^R(L[\mathfrak{A}]) = \infty, \tag{4}$$

   then there is no $\mathcal{D}$-interpretation $i : \mathrm{DIV} \to L$, recursive w.r.t. R.

2. *If* (4) *holds for every oracle R, then some sentence of* DIV *is inexpressible in L over $\mathcal{D}$.*

Proof. For Item i we use Theorem 3.2, Item i, $i = 1, 2$. In both cases it is enough to to show that $C^R(\mathrm{DIV}[\mathfrak{A}]) \geqslant \log \|\mathfrak{A}\|$ for infinitely many $\mathfrak{A}$ in $\mathcal{D}$.

Consider the set $\mathcal{D}_r = \{\mathfrak{A} \ : \ \mathfrak{A} \in \mathcal{D},\ 2^r \leqslant \|\mathfrak{A}\| < 2^{r+1}.\}$ $\mathcal{D}_r$ contains structures of 2^r different cardinalities, and $\log \|\mathfrak{A}\| = r$ for $\mathfrak{A} \in \mathcal{D}_r$. Now let

$X_r = \{\mathrm{DIV}[\mathfrak{A}] \;:\; \mathfrak{A} \in \mathcal{D}_r\}$. Since $\mathrm{DIV}[\mathfrak{A}] \neq \mathrm{DIV}[\mathfrak{B}]$ whenever $\|\mathfrak{A}\| \neq \|\mathfrak{B}\|$, it follows $|X_r| = 2^r$. Applying Proposition 2.4 we are guaranteed to have $\mathfrak{A} \in \mathcal{D}_r$ with $C^R(\mathrm{DIV}[\mathfrak{A}]) \geqslant r$, which finishes the proof. □

Note that the above proof is an example of the situation we have already mentioned after introducing the incompressible strings (Proposition 2.4). Again all structures $\mathfrak{A}$ with high complexity of $\mathrm{DIV}[\mathfrak{A}]$ appear to be "difficult cases" for L. Comparing the above proof with the proof of Theorem 3.2, we can realize that precisely for such structures it is impossible to reconstruct $\|\mathfrak{A}\|$ from $L[\mathfrak{A}]$.

In case $\|\mathcal{D}\|$ is a proper subset of $\mathbb{N}$ a similar result can be easily proven, in which $\log\|\mathfrak{A}\|$ is replaced by an expression depending on the appropriate version of *asymptotic density* of $\|\mathcal{D}\|$ in $\mathbb{N}$.

The above theorem again justifies the idea of considering KE as an extension of 0-1 laws, this time without any restriction of the logic this argument applies to.

We introduce a new "logic" now, first defined in this form by Abiteboul and Vianu (1991b). A *Loosely Coupled Generic Machine* (GM^{loose} for short) consists of a Turing machine augmented with a finite set of fixed arity relations forming the relational store. Apart from standard operations Turing machines can perform, GM^{loose} can apply a first order definable transformation to some of its relations. It can test its relational store if it satisfies some first order sentences, using the result in the computations. The transformations and test sentences it can use are encoded in its finite control. Such machines provide a theoretical model of database application programs, which use first order (i.e., SQL) queries embedded in a full programming language, such as C. The intention is that the input of GM^{loose} is a structure in the relational store, on which the machine can perform some computations, the output of which can be a structure in the relational store again, or a word written on a standard tape.

Corollary 6.7. Let $\|\mathcal{D}\| = \mathbb{N}$. Suppose that for every k the difference $\log\|\mathfrak{A}\| - C(\mathrm{FO}^k[\mathfrak{A}])$ is unbounded in $\mathcal{D}$. Then there is no GM^{loose} machine which can compute for all $\mathfrak{A} \in \mathcal{D}$, given $\mathfrak{A}$ as its relational input, the cardinality of $\mathfrak{A}$ written on its output tape.

Proof. Let us suppose to the contrary that such a machine M exists. Let k be the largest number of variables which appear in some first order manipulation (transformation or test) which M can perform. It is then an easy induction proof that during any computation of M over input $\mathfrak{A}$, M has always in its relational store relations which are FO^k-definable in $\mathfrak{A}$, and any first order test it performs during this computation is a test in FO^k. Hence what M actually does is reconstruction of $\|\mathfrak{A}\|$ from $\mathrm{FO}^k[\mathfrak{A}]$. A simple modification of M yields then a construction of a machine M_p testing if $\|\mathfrak{A}\| \equiv 0 \pmod{p}$, for each p. But this contradicts the previous theorem, since we have recursively interpreted DIV within a recursive "logic" L of total GM^{loose} machines with k variables computing Boolean functions, even though $C(L[\mathfrak{A}]) \leqslant C(\mathrm{FO}^k[\mathfrak{A}]) + O(1)$. This finishes the proof. □

Note the reason why GM^{loose} fails to compute $\|\mathfrak{A}\|$ above. It is not any lack of computational power — it can have as much of it as we want. It is the lack of information about the structure. Further note that we could use essentially any query languege L closed under query composition in place of FO^k, as well as allow the machines to create the queries at runtime in the above corollary, and the proof would still work for such machines.

6.3 KE and Descriptive Complexity

The results in this section show that surprisingly the area of the KE applicability overlaps with the area of the descriptive complexity applicability. Over classes consisting of structures in which the complexity of $C(\mathrm{PFP}^k[\mathfrak{A}])$ is sufficiently high, the sentences of PFP itself or of its simple extension define already all they could: all PSPACE properties. This indicates that KE for PFP indeed fills a no man's land between classes in which the expressive power of PFP is extremely low (i.e., 0-1 laws hold), and those in which this expressive power achieves all of PSPACE.

Theorem 6.8. *Let $k \in \mathbb{N}$ and $\varepsilon > 0$ and the signature σ of structures be fixed.*

Then there exists a Δ_2^p-computable $c\cdot\lceil 2/\varepsilon\rceil$-ary predicate Q, where c depends on σ, only, such that the sentences of $\mathrm{PFP}(Q)$ *(i.e.,* PFP *with Q added as a language primitive) define precisely all* PSPACE *subsets of the class $\mathcal{D}$ of all finite $\mathfrak{A}$ such that $C(\mathrm{PFP}^k[\mathfrak{A}]) \geqslant \|\mathfrak{A}\|^{\varepsilon}$.*

Proof. Let k, ε and $\mathcal{D}$ be as in the statement of the theorem. By Theorem 5.2 we deduce that for all $\mathfrak{A} \in \mathcal{D}$ $\|\pi_k(\mathfrak{A})\| \geqslant \|\mathfrak{A}\|^{\varepsilon/2}$. The structure $\pi_k(\mathfrak{A})$ is definable in PFP^{2k+2}. Let $m = \lceil 2/\varepsilon\rceil$. The structure $(\pi_k(\mathfrak{A}))^m$ — the $m-$th Cartesian power of $\pi_k(\mathfrak{A})$ is definable in PFP^{2mk+2m}. It can be ordered lexicographically in PFP. Certainly, $\|(\pi_k(\mathfrak{A}))^m\| \geqslant \|\mathfrak{A}\|$. Now let the predicate Q be chosen such that it defines over $(\pi_k(\mathfrak{A}))^m$ the canonization of $\mathfrak{A}$, i.e., an ordered isomorphic copy of $\mathfrak{A}$, first in the lexicographic ordering of all ordered isomorphic copies of $\mathfrak{A}$.

E.g., let $\mathfrak{A}$ be a graph, and let $\mathfrak{A}'$ be its canonization, i.e., an ordered graph whose adjacency matrix is lexicographically first among all the graphs isomorphic to $\mathfrak{A}$. Then Q is $2m$-ary, and for two tuples $\vec{a}, \vec{b} \in |\mathfrak{A}|^m$ we have $Q(\vec{a}, \vec{b})$ iff for some $0 \leqslant i, j < \|\mathfrak{A}\|$, $\vec{a}$ is in the i-th class and $\vec{b}$ in the j-th class according to the lexicographic ordering of $(\pi_k(\mathfrak{A}))^m$, and there is an edge between i and j in $\mathfrak{A}'$.

Since canonization of finite structures is in Δ_2^p, Blass and Gurevich (1995), the predicate is computable in this complexity class, as well. Now, because $\mathrm{PFP}(Q)$ defines over some Cartesian power of $\mathfrak{A}$ the isomorphic, ordered copy of $\mathfrak{A}$, by Theorem 4.3 sentences of $\mathrm{PFP}(Q)$ define all PSPACE subsets of $\mathcal{D}$. $\square$

Concerning the last theorem, it appears from the proof that Q (which can be quite formally introduced to the logic as so called *Lindström quantifier*) is not used to provide any computational power to PFP. It just "pumps" the missing information in, and then the whole computational task is performed without any use of it.

So a similar comment as the one we have made after proof of Corollary 6.7 can be made here: what we provided PFP with introducing Q, was information about the structure rather than computational power. We had to do it, as the following simple example shows:

Example. *Let $f, g : \mathbb{N} \to \mathbb{N}$ be two functions. Let $\sigma = \langle E, \leqslant \rangle$ be the signature of directed graphs with partial ordering of the universe. The class $\mathcal{D}_{f,g} \subseteq \Sigma$ consists of all graphs G, which are disjoint unions of: a graph G_0 of cardinality n* linearly ordered *by $\leqslant$, and two cliques: G_1 of size $f(n)$ and G_2 of size $g(n)$, in which $\leqslant$ is the identity relation.*

Let $f(n) = g(n) = f'(n) = n$ and $g'(n) = n + 1$. Then it is easy to see that in $\mathcal{D} = \mathcal{D}_{f,g} \cup \mathcal{D}_{f',g'}$ holds $C(\mathrm{FO}^3[\mathfrak{A}]) = C(\mathfrak{A}) + O(1)$, and yet PFP *does not define all PSPACE subsets of $\mathcal{D}$. The inexpressible property is "the cliques are of the same cardinality", which can be demonstrated by an application of the Ehrenfeucht-Fraïssé game for $\mathcal{L}^{\omega}_{\infty\omega}$. However, $\mathcal{D}$ satisfies the premises of Theorem 6.8, and therefore all PSPACE subsets of $\mathcal{D}$ can be defined by sentences of* PFP *extended by one Lindström quantifier, which is then really necessary to tell the logic whether the cliques are of equal cardinality or not.*

The reason that PFP *alone falls short is that for every k there are in $\mathcal{D}$ infinitely many pairs of nonisomorphic structures $\mathfrak{A}, \mathfrak{B}$ such that $\pi_k(\mathfrak{A}) \cong \pi_k(\mathfrak{B})$.*

In the limit of the previous theorem we can get rid of Q. This is possible when PFP does not miss any information about structures.

Theorem 6.9. *Let us suppose that $\mathcal{D}$ is a* PSPACE *class of finite structures of some fixed signature, and that $k \in \mathbb{N}$ is such that $\lambda\mathfrak{A}.\mathrm{PFP}^k[\mathfrak{A}]$ is injective on $\mathcal{D}$. Moreover, let there exist an $\varepsilon > 0$ such that $C(\mathrm{PFP}^k[\mathfrak{A}]) \geqslant \|\mathfrak{A}\|^{\varepsilon}$ for every $\mathfrak{A} \in \mathcal{D}$. Then the sentences of* PFP *define precisely all* PSPACE *subsets of $\mathcal{D}$.*

Proof. It is enough to show, exactly as in the previous proof, that uniformly for all $\mathfrak{A} \in \mathcal{D}$, the logic PFP can interpret over some Cartesian power of $|\mathfrak{A}|$ an ordered copy of $\mathfrak{A}$ itself. Exactly as before, interpreting the ordering is not a problem. Then, having the whole computational power of PSPACE over this ordering expressible in PFP, and also the definability of $\pi_k(\mathfrak{A})$ in PFP, the formula we need expresses the result of the following PSPACE computation: consider consecutively, in the lexicographic ordering, all finite structures in $\mathcal{D}$ equipped with orderings, looking for the first one, say $\mathfrak{B}_{\leqslant} = \langle \mathfrak{B}, \leqslant \rangle$, such that $\pi_k(\mathfrak{B}) \cong \pi_k(\mathfrak{A})$. When it is found, then by our assumptions $\mathfrak{B} \cong \mathfrak{A}$. So we have indeed found an interpretation of an ordered copy of $\mathfrak{A}$ in some Cartesian power of $\mathfrak{A}$ itself. $\square$

The results in this section have been independently obtained by Seth (1995) in a slightly different formulation. His results require values of $\|\pi_k(\mathfrak{A})\|$ to be of order $\|\mathfrak{A}\|^{\varepsilon}$, while we have used an analogous requirement concerning $C(\mathrm{PFP}^k[\mathfrak{A}])$. In virtue of Theorem 5.2 our assumption implies that of Seth (but not vice versa, so our results are somewhat weaker).

6.4 What Is so Special About PFP?

The question from the title of this section is quite legitimate. We have noticed a surprising explanation of both the 0-1 laws and capturing of PSPACE by this logic in terms of $C(\mathrm{PFP}^k[\mathfrak{A}])$. Is it a special feature of PFP or can this happen for other logics?

The first candidate is LFP. The 0-1 law explanation holds. Are the counterparts of Theorems 6.8 and 6.9, but for LFP and PTIME instead of PFP and PSPACE, also true? This is rather unlikely, since the computations we have used in the proofs and encoded in PFP do not seem to be doable in PTIME. But it is easy to see that refutation of them would imply separation of PTIME and PSPACE. Indeed, PTIME=PSPACE implies LFP $\equiv_{fin}$ PFP, Theorem 4.3, Item 5. So assuming the first equality, we can replace in both theorems all occurrences of PTIME by PSPACE and all occurrences of LFP by PFP, getting the desired counterparts. Now immediately we ask if the converse of this observation is also true?

For other logics even the explanation of 0-1 laws may fail, as we have shown in Proposition 6.5.

7 Questions

This paper reports an ongoing research. There are relatively few known facts, and many questions. Let us present some of them:

1. The Kolmogorov expressive complexity $C(L[\mathfrak{A}])$ has been defined as the number of bits necessary to describe the L-theory of $\mathfrak{A}$. The Kolmogorov expressive power $I_L(\mathfrak{A})$ is the number of bits of the description of (the isomorphism type of) a structure $\mathfrak{A}$ we can learn by having access to the L-theory of $\mathfrak{A}$. So $I_L(\mathfrak{A})$ says how much of $C(L[\mathfrak{A}])$ is really used to express properties of $\mathfrak{A}$, while the latter measures merely the complication of the theory.

   The connections between the two notions are unclear (see Tyszkiewicz (1995)), and it seems an intriguing question, if $C(L[\mathfrak{A}])$ can be much larger than $I_L(\mathfrak{A})$, because it is to some extent a question, how much inaccessible or useless information can there be in an L-theory of a structure.

2. Definitions similar to those investigated in this paper can be given for almost any of over a dozen versions of Kolmogorov complexity. What are the natural areas of applicability for the choices other than the plain complexity considered here?

3. What are the methods to estimate $C(L[\mathfrak{A}])$? How are they related to Ehrenfeucht-Fraïssé games, which are used in the proof of the theorem in Tyszkiewicz (1995), mentioned in the comment after the proof of Theorem 3.2?

4. What is the relationship between 0-1 laws and convergence laws on the one hand (cf. Definition 6.1) and KE on the other hand, especially for

logics other than sublogics of $\mathcal{L}^{\omega}_{\infty\omega}$? How KE, viewed as a generalization of 0-1 laws, relates to other generalizations of them, defined by imposing less restrictive conditions on the asymptotic behavior of sequences $\nu_n^{\mathcal{D}}(\varphi)$ for $\varphi \in L$? E.g., such a condition is the notion of a very weak 0-1 law of Shelah (1996), mentioned already in Section 1.2.

5. It is easy to see that, similarly to KE, the *data complexity* in the Kolmogorov version (KD) can also be defined. In the naive approach, if we just take the Kolmogorov complexity of rows of the matrix, we get an uninteresting notion: for every logic able to express infinitely many different semantical notions the upper bound of complexities of rows is ∞. To make comparisons finer than just the trivial one between logics which can/cannot define infinitely many different semantical properties, we have to compare the Kolmogorov complexities of rows, suspected to be identical. But now it is not more complicated to verify directly the identity of rows rather than their Kolmogorov complexity, which is noncomputable.

   But there is a reasonable solution of this problem, and the notion appears even to be useful as a tool in proving inexpressibility results. In fact, most of the hierarchy results for Lindström quantifiers, shown by Hella, Luosto and Väänänen (1996), have been proven by arguments equivalent to the incompressibility method combined with (an appropriate version of) KD, as shown by Tyszkiewicz (1996).

8 Conclusion

We have defined the Kolmogorov expression complexity KE of a given logic. Our main intention was to create an invariant of expressive power of logics which has the following properties, unlike those already existing.

- KE is of the expression type.
- KE depends on the ability to express information rather than computations.

By investigating KE for least fixed point and partial fixed point logics we have shown that we have really achieved these goals.

Acknowledgment I would like to thank Katarzyna Benowska and the anonymous referees for their efforts to correct my English. One of the referees has suggested an improved version of Theorem 6.6.

References

Abiteboul, S., and Vianu, V. (1991a), Datalog extensions for database queries and updates, *Journal of Computer and System Sciences,* **43**, pp. 62–124.

Abiteboul, S., and Vianu, V. (1991b), Generic computation and its complexity, *in* “Proc. ACM SIGACT Symp. on the Theory of Computing”, pp. 209–219.

Blass, A., and Gurevich, Y. (1995), Equivalence relations, invariants and normal forms, *SIAM Journal of Computing* **123**, pp. 172–184.

Chandra, A. (1982), Programming primitives for database languages, in: *Proc. ACM Symp. on Principles of Programming Languages* pp. 50–62.

Chandra, A., and Harel, D. (1980), Computable queries for relational data bases, *Journal of Computer and System Sciences* **21**, pp. 156–178.

Dawar, A., Lindell, S., and Weinstein, S. (1995), Infinitary logic and inductive definability over finite structures, *Inf. Computation* **119**, pp. 160–175.

Ebbinghaus, H.-D., and Flum, J. (1995), “Finite model theory”, Perspectives in Mathematical Logic, Springer Verlag, Berlin.

Fagin, R. (1972), Probabilities on finite models, *Notices of the American Mathematical Society*, October 1972, Abstract number 72T-E90, p. A714.

Fagin, R. (1974), Generalized first-order spectra and polynomial-time recognizable sets, *in* “Complexity of Computation, SIAM-AMS Proc.” **7**, (Karp, R.M., Ed.), pp. 43–73.

Fagin, R. (1976), Probabilities on finite models, *Journal of Symbolic Logic* **41**, pp. 50–58.

Fagin, R. (1993), Finite-model theory — a personal perspective, *Theoretical Computer Science* **116**, pp. 3–31.

Glebskiĭ, J.V., Kogan, D.I., Liogon'kiĭ M.I., and Talanov V.A. (1969), Range and degree of realizability of formulae in the restricted predicate calculus, *Kibernetika* **2**, pp. 17–28.

Hella, L., Luosto, K., and Väänänen, J. (1996), The hierarchy theorem for generalized quantifiers, *Journal of Symbolic Logic* **61**, pp. 802–817.

Immerman, N. (1986), Relational queries computable in polynomial time, *Inf. Control* **68**, pp. 86–104.

Immerman, N. (1989), Descriptive and computational complexity, *in* “Computational Complexity Theory, Proc. Symp. Applied Math.” **38** (Hartmanis, J., Ed.), pp. 75–91, American Mathematical Society.

Li, M., and Vitányi, P.M.B. (1993), “An introduction to Kolmogorov complexity and its applications”, Springer Verlag, New York.

Lynch, J.F. (1980), Almost sure theories, *Annals Math. Logic* **18**, pp. 91–135.

Seth, A. (1995), When do fixed point logics capture complexity classes?, *in* “Proc. 10th IEEE Symp. Logic in Computer Science”, pp. 353–363, IEEE Press.

Shelah, S. (1996), On the very weak 0-1 law for random graphs with orders, *J. Logic Computat.* **6** No. 1, pp. 137-159.

Tyszkiewicz, J. (1993), On asymptotic probabilities in logics that capture DSPACE($\log n$) in presence of ordering, *in* "Proc. TAPSOFT'93", (Gaudel, M.-C., and Jouannaud, J.-P., Eds.), pp. 569-583, Lecture Notes in Computer Science No. 668, Springer Verlag.

Tyszkiewicz, J. (1995), On the Kolmogorov expressive power of Boolean query languages, to appear in *Theoretical Computer Science,*
preliminary version in "Proc. Database Theory—ICDT'95" (Gottlob G. and Vardi, M.Y., Eds.), pp. 97–110, Lecture Notes in Computer Science No. 893, Springer Verlag.

Tyszkiewicz, J. (1996), A note on the Kolmogorov data complexity and nonuniform logical definitions, *submitted.*

Vardi, M.Y. (1982), Complexity of relational query languages, *in* "Proc. 14th ACM Symp. on Theory of Computing", pp. 137–146.