More about Recursive Structures: Descriptive Complexity and Zero-One Laws

Tirza Hirst* David Harel†

Abstract: This paper continues our work on infinite, recursive structures. We investigate the descriptive complexity of several logics over recursive structures, including first-order, second-order, and fixpoint logic, exhibiting connections between expressibility of a property and its computational complexity. We then address 0–1 laws, proposing a version that applies to recursive structures, and using it to prove several non-expressibility results.

0 Introduction

Infinite recursive structures, with recursive graphs as a special case, have been studied quite extensively in the past. Most interesting properties of recursive graphs have been shown to be undecidable, and many are actually outside the arithmetic hierarchy; see, e.g., [AMS, Be1, Be2, BG, H, HH1]. In [HH2] we considered recursive structures to be generalizations of finite relational data bases, and investigated the class of computable queries over them, the motivation being borrowed from [CH1]. A computable query is a (partial) recursive function that is also generic, i.e., it preserves isomorphisms. One of the results of [HH2] is that quantifier-free first-order logic is complete for recursive data bases, meaning that it expresses precisely the recursive generic queries. Part 1 of this paper deals with languages that can express non-recursive queries.

If we return for a moment to the world of finite structures, classical complexity classes can often be characterized in terms of their *descriptive complexity*, i.e., the ability of certain logics to express properties in the class. The first important result was that of Fagin [F1], who showed that existential second order logic (that is, Σ^1_1 formulas) expresses exactly the properties in NP. An analogous match holds between each level of the second-order hierarchy and the corresponding level of the polynomial-time hierarchy. In Part 1 of the paper we carry out an analogous investigation over recursive structures, analyzing the expressibility of several logics, such as first-order logic, second-order logic and fixpoint logic. We are able to exhibit many connections between descriptive and computational complexity, such as the following result, which is analogous to that of Fagin: the properties of recursive structures expressible by a Σ^1_k formula, for $k \geq 2$, are exactly the generic properties in the complexity class Σ^1_k of the analytic hierarchy.

*The Weizmann Institute of Science, Rehovot, Israel. Current email: hirst-tirzah@cs.yale.edu. Part of this author's work was carried out during a postdoctoral fellowship at Bar-Ilan University, Ramat-Gan, Israel.

†The Weizmann Institute of Science, Rehovot, Israel. Email: `harel@wisdom.weizmann.ac.il`. Part of this author's work was carried out during a sabbatical at the Department of Computer Science, Cornell University, Ithaca, NY.

In Part 2 of the paper we address the issue of whether there are useful 0–1 laws for recursive structures. Some classical results and methods developed for general models, such as completeness and compactness, fail for finite and recursive structures, while others, such as Ehrenfeucht/Fraisse games, hold for these too. However, one tool that appears to be unique to finite model theory is the 0–1 law, that asserts that certain properties (such as those expressible in first-order logic) are either almost surely true or almost surely false [F3]. This notion seems to require that structures be finite, since the notion of "almost surely" relies on asymptotic behavior as the structures grow in size. We propose an extension of this notion to recursive structures, and show that the new kind of 0–1 law holds for several logics, such as first-order logic, strict Σ^1_1 and strict Π^1_1. We then use this to show non-expressibility of properties in such logics.

Each of the two parts of the paper starts with a more detailed background and overview section.

1 Descriptive Complexity

1.1 Background and Overview

First, some definitions taken from [HH2]. Throughout the paper we replace the term "data base" by "structure", although for all our purposes they are synonymous.

Definition 1.1 *Let D be a countably infinite recursive set, and let $R_1, \ldots, R_k$, $k > 0$, be relations, such that for all $1 \leq i \leq k$, $R_i \subseteq D^{a_i}$. We say that $B = (D, R_1, \ldots, R_k)$ is a recursive relational structure of type $a = (a_1, \ldots, a_k)$ (an r-st for short[1]) if each R_i is a recursive relation. We often use $D(B)$ to denote D, the domain of B.*

Each R_i is *recursive relation*, and can be represented by a Turing machine, which on input u decides whether $u \in R$. Throughout the paper we view an r-st B as being represented by a sequence of Turing machines that represent its relations.

Definition 1.2 *Let $B_1 = (D_1, R_1, \ldots, R_k)$ and $B_2 = (D_2, R'_1, \ldots, R'_k)$ be two r-st's of type $a = (a_1, \ldots, a_k)$, and let $u \in D_1^n$ and $v \in D_2^n$. We say that B_1 and B_2 are isomorphic by isomorphism h, if $h : D_1 \to D_2$ is a bijection, and $h(R_i) = R'_i$ for each $1 \leq i \leq k$. B_1 and B_2 are isomorphic, written $B_1 \cong B_2$, if B_1 and B_2 are isomorphic by some isomorphism. We also write $(B_1, u) \cong (B_2, v)$, for tuples u and v if B_1 and B_2 are isomorphic by an isomorphism that maps u to v, and $u \cong_B v$ if $(B, u) \cong (B, v)$.*

Definition 1.3 *An r-st query of type a (or an r-query for short) is a partial function Q, which, for each r-st B of type a, yields an output (if any) that is a recursive relation over $D(B)$. We let $Gr(Q)$ denote the set $\{(B, u) \mid u \in Q(B)\}$.*

We now want to define recursive queries over r-st's. There are various possibilities for this. One is to require the existence of a Turing machine, which, on an input containing (codes for) the Turing machines of the relations in the input structure, produces as output the (code of) the Turing machine for the output relation. We prefer the following oracle-based definition:

[1] In [HH2] an r-st is called an r-db.

Definition 1.4 *An r-query Q is recursive if there is an oracle Turing machine which, given a tuple u, uses oracles for the relations of the input structure B to decide whether $u \in Q(B)$. If $Q(B)$ is undefined, the machine does not halt on any input tuple.*

Definition 1.5 *An r-query Q is called generic if it preserves isomorphisms for all structures (not necessarily recursive ones), under the assumption that there exist oracles for their relations. In symbols, for any relational structures B_1 and B_2, if $B_1 \cong B_2$ by isomorphism h then $Q(B_1) \cong Q(B_2)$ by h, where $Q(B)$ is the result of applying Q to oracles for the relations in B.*

Definition 1.6 *An r-query is called computable if it is recursive and generic. A query language is r-complete if it expresses precisely the computable r-queries.*

Given a query language L, we adapt the following from Vardi [V1]:

Definition 1.7 *The data complexity of a language L is the computational complexity of the sets $Gr(Q_e)$, where e ranges over the expressions of L, and Q_e is the query expressed by e. A language L is data-complete (or d-complete for short) for a complexity class C if for every expression e in L, $Gr(Q_e)$ is in C, and there is an expression e_0 in L such that $Gr(Q_{e_0})$ is hard for C.*

In Section 1.2, we show that first-order logic expresses arithmetical generic queries from the entire arithmetical hierarchy, but it does not express them all. In Section 1.3, E-Σ^1_1, which consists of the existential second-order formulas over some vocabulary,[2] is shown to be d-complete for the complexity class Σ^1_1 of the analytic hierarchy, but there are even arithmetical generic queries that are not expressible in E-Σ^1_1. For $k \geq 2$, we have a stronger result in Section 1.4, which is similar to the known result of Fagin [F1] on the equivalence between NP and the existential second-order properties. Namely, E-Σ^1_k expresses precisely the generic properties that are in the complexity class Σ^1_k. This means that every generic query over some vocabulary σ that is in Σ^1_k is expressible by an uninterpreted formula in E-Σ^1_k over σ, not only by a Σ^1_k formula over interpreted recursive predicates.

It is worth saying a little more about this result. The second-order quantifiers are used in the proof to define a total order and predicates $+$ and $*$, which, in turn, are used to define the required elementary arithmetic expression. Also, the defined order must contain a minimal element for every subset of elements, a requirement that needs a universal second-order quantifier. This explains why the result requires $k \geq 2$. In fact, there are arithmetical properties that are not expressible in E-Σ^1_1, such as the one stating that a recursive structure satisfies all the extension axioms (see Part 2 of the paper and [F3]). However, if a built-in total order is added to the vocabulary, all the Σ^1_1 properties are expressible in E-Σ^1_1.

In Sections 1.5 and 1.6 we consider two special cases of E-Σ^1_1: monadic E-Σ^1_1, where the second-order quantifiers are over sets only, is d-complete for Σ^1_1, and strict E-Σ^1_1 is d-complete for Σ^0_2.

In Section 1.7 we show that the data-complexity of iterative logic, which is obtained by adding **while** loops to first-order logic [CH2, KV1], is Δ^1_1, but it does not express all the Δ^1_1 properties.

Section 1.8 deals with fixpoint logic, which is obtained by adding least fixpoint operators to first-order formulas [CH2, Mos]. FP_1 denotes positive fixpoint logic, which is obtained from first-order logic by adding the least fixpoint operator to positive formulas only. The full logic FP is

[2] We use the prefix "E-", standing for "expression", to denote syntactic classes.

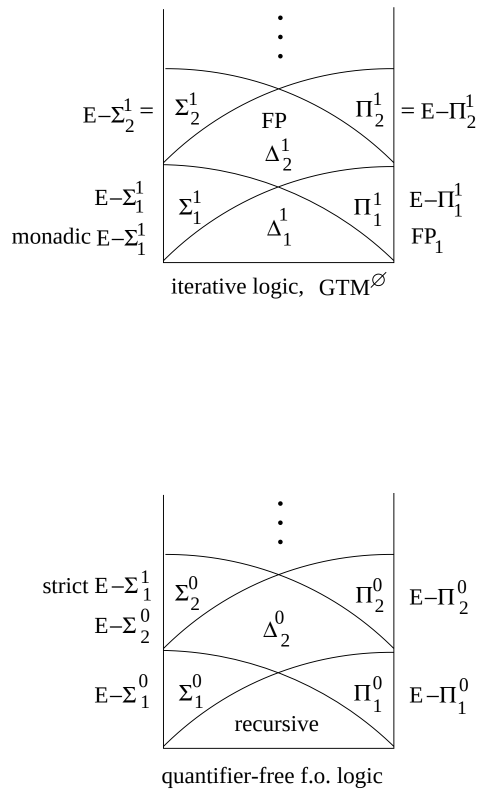

Figure 1: Summary of results in Part 1

obtained by combining the least fixpoint operator with the formation rules of first-order formulas. When fixpoint logic is applied to finite structures, the hierarchy of compositions of least fixpoint operations with first-order constructs (noticeably negation) collapses, and a single fixpoint operator suffices, as shown by Immerman [I]. This is not the case for recursive structures: it turns out that FP_1 is d-complete for Π^1_1, and hence $\neg FP_1$ (containing the negations of expressions in FP_1) is d-complete for Σ^1_1. The data complexity of FP is exactly Δ^1_2, and we exhibit a sample query that is expressible in FP but is hard for Σ^1_1 and Π^1_1.

Finally, in Section 1.9 we investigate the complexity of queries computable by generic Turing machines with oracles for detecting the emptiness of relations. These machines (related to those of [AV2]), which we call $GTM^\emptyset$'s, are shown to have data complexity of Δ^1_1. $GTM^\emptyset$'s compute all the arithmetical queries, and some non-arithmetical ones too. Also, while we do not know if there is an effective language for expressing queries computable by a $GTM^\emptyset$, we show that there is no BP-complete language for expressing the relations defined by them.

Figure 1 summarizes some of the results of this part of the paper.

1.2 First-order logic

Definition 1.8 *Let $\mathcal{L}$ be the language of first-order logic, considered as a query language with queries of the form $\{(x_1, \ldots, x_n) \mid \phi(x_1, \ldots, x_n, R_1, \ldots, R_m)\}$, where ϕ is a first-order formula over some vocabulary $\sigma = (P_1, \ldots, P_m)$. Here, $R_1, \ldots, R_m$ denote the relations of the input r-st of type $a = (a_1, \ldots, a_m)$, where each a_i is the arity of P_i, and $x_1, \ldots, x_n$ are ϕ's free variables. The allowed*

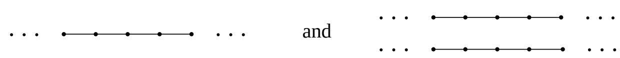

Figure 2: Two graphs

atomic formulas are $x_i = x_j$, for $1 \le i, j \le n$, and $(x_{j_1}, \ldots, x_{j_{a_i}}) \in R_i$, for $1 \le j_1, \ldots, j_{a_i} \le n$ and $1 \le i \le m$. In particular, if R is of rank 0, then $() \in R$ is a legal atomic formula.

Definition 1.9 *Denote by E-Σ_k^0, for $k \ge 1$, the restriction of $\mathcal{L}$ to bounded-quantifier formulas of the form $(\exists \overline{y}_1)(\forall \overline{y}_2) \cdots (\Phi \overline{y}_k)\ \phi(\overline{y}_1, \ldots, \overline{y}_k)$, where Φ is $\exists$ or $\forall$, depending on the parity of k, and ϕ is a quantifier-free formula over some vocabulary σ. E-Π_k^0 is defined similarly, but with the formulas starting with a universal quantifier.*

Proposition 1.1 *For each k, E-Σ_k^0 is d-complete for Σ_k^0, and E-Π_k^0 is d-complete for Π_k^0.*

Proof: Let $k > 0$. Let Q be an r-query that is expressible in E-Σ_k^0 over a vocabulary σ. $Gr(Q) = \{(B, \overline{x}) \mid \psi(\overline{x}, B)\}$, where $\psi(\overline{x}, B)$ is a formula in E-Σ_k^0 over σ. Since B is recursive, $Gr(Q)$ is in the complexity class Σ_k^0.

We now exhibit a Σ_k^0-complete r-query that is expressible in E-Σ_k^0. Let

$$A = \{x \mid (\exists z_1)(\forall z_2) \cdots (\Phi z_k)\ S(x, z_1, \ldots, z_k)\}$$

be a Σ_k^0-complete set over $\mathcal{N}$, where S is a recursive predicate and Φ is $\exists$ or $\forall$, depending on the parity of k. Define the following query Q on r-st's of type $a = (k+1)$:

$$Q(B) = \{x \mid (\exists z_1)(\forall z_2) \cdots (\Phi z_k)\ R(x, z_1, \ldots, z_k)\}.$$

We show now that $Gr(Q)$, which is clearly in Σ_k^0, is also hard for Σ_k^0. Let $B^* = (\mathcal{N}, S)$, where S is the recursive predicate from the definition of A, considered as a relation. Now $Q(B^*) = A$ and $Gr(Q_{B^*}) = \{(B^*, x) \mid x \in Q(B^*)\}$. Since B^* is fixed, $A \cong Gr(Q_{B^*})$, and hence $Gr(Q_{B^*})$ is also Σ_k^0-complete. But $Gr(Q_{B^*})$ is trivially reducible to $Gr(Q)$. Hence $Gr(Q)$ is Σ_k^0-complete too. The case of E-Π_k^0 is dual. $\triangle$

Prop. 1.1 states that $\mathcal{L}$, when applied to recursive structures, expresses queries from the entire arithmetical hierarchy. However, not all the arithmetical queries are expressible in $\mathcal{L}$. For example, connectivity of recursive graphs, which is arithmetical, is not expressible by a first-order formula. This can be shown by using Ehrenfeucht-Fraisse games (see the Appendix) to prove that no first-order formula can distinguish between the two recursive graphs in Figure 2.

Another example is the formula EA, defined in Part 2 of the paper, which states that a recursive structure satisfies all the extension axioms. We shall show that EA is not expressible even in E-Π_1^1, although it can easily be shown to be in Π_2^0.

Note that while first-order logic is closed under negation E-Σ_k^0 and E-Π_k^0, for $k > 0$, cannot be, due to their d-completeness in the appropriate complexity classes.

1.3 E-Σ_1^1 and E-Π_1^1

Definition 1.10 *Denote by E-Σ_k^1, for $k \geq 1$, the restriction of second-order logic to formulas of the form $(\exists\overline{S}_1)(\forall\overline{S}_2)\cdots(\Phi\overline{S}_k)\ \phi(S_1,\ldots,S_k)$, where Φ is $\exists$ or $\forall$, depending on the parity of k, and ϕ is a formula in first-order logic over some vocabulary σ. E-Π_k^0 is defined similarly, but with the formulas starting with a universal quantifier.*

Proposition 1.2 *E-Σ_1^1 is d-complete for Σ_1^1, and E-Π_1^1 is d-complete for Π_1^1.*

Proof: Clearly, if Q is an r-query expressible in E-Σ_1^1, then $Gr(Q) \in \Sigma_1^1$. In addition, ∞-CLIQUE, which asks whether a recursive graph has an infinite clique, and is known to be Σ_1^1-complete [HH1], is expressible in E-Σ_1^1 as follows:

$$Gr(\infty\text{-CLIQUE}) = \{(V,E) \quad | \quad (\exists <)\,(\exists C)\ \psi(<)\ \wedge \\ \forall x,y((x\in C\ \wedge\ y\in C)\ \rightarrow\ (x,y)\in E) \wedge\ \forall x\exists y\ (x<y\ \wedge y\in C)\}$$

Here $\psi(<)$ verifies that $<$ contains an infinite chain $x_1 < x_2 < \cdots$ without loops, and can be expressed as follows:

$$\forall x\forall y\forall z\ (\neg(x<x)\ \wedge(x\neq y \rightarrow x<y \vee y<x)\ \wedge\neg(x<y\wedge y<x)\ \wedge \\ ((x<y\ \wedge\ y<z)\ \rightarrow\ x<z))$$

The case of E-Π_1^1 is dual. $\triangle$

Thus, E-Σ_1^1 expresses queries that are in the class Σ_1^1 of the analytic hierarchy; but, again, not all of them, since from Example 2.2 it follows that $\overline{\text{EA}}$ is not expressible in E-Σ_1^1 although it is arithmetical. Similarly, E-Π_1^1 does not express all the Π_1^1 queries. In contrast, in the presence of a built-in successor relation S and a total order $<$, all the Σ_1^1 and Π_1^1 queries over σ are expressible by formulas in E-Σ_1^1 or E-Π_1^1, respectively, as we now show. Denoting these extensions by E-$\Sigma_1^1(\sigma,S,<)$ and E-$\Pi_1^1(\sigma,S,<)$, we have:

Proposition 1.3 *An r-query over some vocabulary σ is in the complexity class Σ_1^1 (respectively, Π_1^1) if and only-if it is expressible in E-$\Sigma_1^1(\sigma,S,<)$ (respectively, E-$\Pi_1^1(\sigma,S,<)$).*

Proof: Let Q be an r-query expressible in E-$\Sigma_1^1(\sigma,S,<)$. $Gr(Q) \in \Sigma_1^1$, since Σ_1^1 can clearly define a recursive successor and order for every recursive structure. Conversely, assume that $Gr(Q) \in \Sigma_1^1$. $Gr(Q)$ can be defined as follows:

$$Gr(Q) = \{(B,\overline{x}) \mid (\exists\overline{R})(\exists\overline{x}_1)(\forall\overline{x}_2)\cdots(\forall\overline{x}_m)\ \psi(B,\overline{R},\overline{x},\overline{x}_1,\overline{x}_2,\ldots,\overline{x}_m)\}$$

where ψ is in elementary arithmetic [R]. $Gr(Q)$ can also be written using a built-in successor and order, as follows:

$$Gr(Q) \quad = \quad \{(B,\overline{x}) \mid\ (\exists+)(\exists*)(\exists 0)(\exists\overline{R})(\exists\overline{x}_1)(\forall\overline{x}_2)\cdots(\forall\overline{x}_m) \\ (\tau(S,<,+,*,0)\ \wedge\ \ \psi'(B,S,<,+,*,0,\overline{R},\overline{x},\overline{x}_1,\overline{x}_2,\ldots,\overline{x}_m))\}$$

where τ is a first-order formula stating that $+$ and $*$ satisfy the appropriate axioms and that 0 is the minimal element. ψ' is ψ written by using the defined relation symbols instead of the elementary arithmetical operations. The entire formula is now in E-$\Sigma_1^1(\sigma,S,<)$.

The case for E-$\Pi_1^1(\sigma,S,<)$ is dual, since $Gr(Q)$ is expressible in E-$\Pi_1^1(\sigma,S,<)$ iff $\overline{Gr(Q)}$ is expressible in E-$\Sigma_1^1(\sigma,S,<)$, iff $\overline{Gr(Q)} \in \Sigma_1^1$, iff $Gr(Q) \in \Pi_1^1$. $\triangle$

1.4 E-Σ_k^1 and E-Π_k^1, for $k \geq 2$

In contrast to E-Σ_1^1 and E-Π_1^1, we now show that the higher-levels of the second-order syntactic hierarchy, E-Σ_k^1, for $k \geq 2$, express, respectively, all the Σ_k^1 generic queries. In other words, each query in the complexity class Σ_k^1 that preserves isomorphisms can be expressed by a formula in E-Σ_k^1 over its own vocabulary. The same holds for the universal classes.

Proposition 1.4 *Let Q be a generic r-query over σ, and let $k \geq 2$. $Gr(Q) \in \Sigma_k^1$ iff Q is expressible in E-Σ_k^1, and $Gr(Q) \in \Pi_k^1$ iff Q is expressible in E-Π_k^1.*

Proof: The "if" direction is clear. For the "only-if" direction, assume that $Gr(Q) \in \Sigma_k^1$, and hence it can be expressed as follows:

$$Gr(Q) = \{(B, \overline{x}) \mid (\exists \overline{R}_1)(\forall \overline{R}_2) \cdots (\Phi \overline{R}_k)(\exists \overline{x}_1)(\forall \overline{x}_2) \cdots \psi(B, \overline{R}_1, \overline{R}_2, \ldots, \overline{R}_k, \overline{x}, \overline{x}_1, \overline{x}_2, \ldots)\}$$

where Φ is $\exists$ or $\forall$, depending on the parity of k, and ψ is in elementary arithmetic. In addition, let E be the following set:

$$\begin{aligned} E \;=\; & \{(B, \overline{x}) \mid (\exists S)(\exists <)(\exists +)(\exists *)(\exists 0)(\exists \overline{R}_1)(\forall X)(\forall \overline{R}_2) \cdots (\Phi \overline{R}_k) \\ & (\exists y)(\forall z)(\exists \overline{x}_1)(\forall \overline{x}_2) \cdots \;(\tau(S, <, +, *, 0, X, y, z) \;\wedge\; \psi'(\cdots))\} \end{aligned}$$

Here, τ states that S is a successor relation, $<$ is a total linear order that agrees with S, $+$ and $*$ satisfy the appropriate axioms, 0 is the minimal element and y is the smallest element in the set X. In addition, ψ' is ψ written using the defined relation symbols instead of the the usual arithmetical operations. The entire formula is now in Σ_k^1 over σ.

We show that $Gr(Q) = E$, due to the genericity of Q. Clearly, $Gr(Q) \subseteq E$. Let $(B, u) \in E$ and let $<'$ be the order relation that exists by the formula in E. Let π be the isomorphism obtained by associating $<'$ with the standard order $<$ on $\mathcal{N}$. Let (B', v) be the structure and tuple obtained by applying π to (B, u). Now, $(B', v) \in Gr(Q)$, since all conditions appearing in the formula of $Gr(Q)$ are satisfied in the same way as the corresponding ones in E are satisfied for (B, u). Since $(B', v) \cong (B, u)$, we have $(B, u) \in Gr(Q)$.

The case for Π_k^1 follows by duality. $\triangle$

Thus, the situation for E-Σ_k^1, for $k \geq 2$, is similar to that for E-$\Sigma_1^1(\sigma, S, <)$. We use the second-order quantifiers to define the successor and order. Since E-Σ_1^1 does not express all the generic Σ_1^1 queries ($\overline{\mathrm{EA}}$, for example) a total order cannot be expressed using only existential second-order quantifiers, and requires a universal quantifier which is used to verify that every subset has a least element.

However, there is an essential difference between the way E-$\Sigma_1^1(\sigma, S, <)$ captures Σ_1^1 and the way E-Σ_k^1 captures Σ_k^1, for $k \geq 2$. E-$\Sigma_1^1(\sigma, S, <)$ expresses *all* the Σ_1^1 queries, whereas E-Σ_k^1 express only generic Σ_k^1 queries. E-$\Sigma_1^1(\sigma, S, <)$ does not preserve the isomorphisms of the original structure because of the given order. In contrast, the successor and order that are required to exist for E-Σ_k^1 can differ from one structure to another, and isomorphic r-st's will have isomorphic successor and order relations.

Note that the relationship established in Prop. 1.4 between E-Σ_k^1 and Σ_k^1 is stronger than d-completeness, which follows as a corollary:

Corollary 1.5 *E-Σ_k^1 and E-Π_k^1, for $k \geq 2$, are d-complete for Σ_k^1 and Π_k^1, respectively.*

Corollary 1.6 *Let Q be a generic r-query over σ, and let $k \geq 2$. $Gr(Q) \in \Delta_k^1$ iff Q is expressible in both E-Σ_k^1 and in E-Π_k^1.*

1.5 Monadic E-Σ_1^1

Definition 1.11 *Monadic E-Σ_1^1 is the query language containing formulas of the form $\exists A_1 \cdots \exists A_k\ \phi(A_1, \ldots, A_k)$, where each of the A_i is unary and ϕ is a first-order formula over some vocabulary σ.*

Proposition 1.7 *Monadic E-Σ_1^1 is d-complete for Σ_1^1.*

Proof: Clearly, a query expressible in monadic E-Σ_1^1 is in the complexity class Σ_1^1. To prove completeness, we use ∞-CLIQUE-2, which is a variant of ∞-CLIQUE. (∞-CLIQUE itself is not expressible in monadic E-Σ_1^1, as we show in Prop. 1.8.) Let $\sigma = (V, E, H)$, where V is unary and E and H are binary. Define ∞-CLIQUE-2 to be the class of recursive structures over σ that contain an infinite clique, and in which H is an order (not necessarily with a minimal element for every subset) satisfying:

$$\begin{aligned} \forall x, y, z \quad (\neg(x,x) \in H \ \wedge\ (x \neq y \rightarrow (x,y) \in H \vee (y,x) \in H)\ \wedge\ \neg((x,y) \in H \wedge (y,x) \in H) \wedge \\ ((x,y) \in H \ \wedge\ (y,z) \in H)\ \rightarrow\ (x,z) \in H\ ) \end{aligned}$$

If we denote this requirement by $\phi(H)$, we may now express ∞-CLIQUE-2 in monadic E-Σ_1^1 as follows:

$$\begin{aligned} Gr(\infty\text{-CLIQUE-2}) \quad = \quad & \{(V, E, H) \mid \exists C\ (\phi(H)\ \wedge\ \forall x \forall y\ ((x \in C\ \wedge\ y \in C)\ \rightarrow\ (x,y) \in C) \\ & \wedge\ \forall x \exists y\ (H(x,y)\ \wedge y \in C))\} \end{aligned}$$

It is easy to see that ∞-CLIQUE is reducible to ∞-CLIQUE-2, by applying ∞-CLIQUE-2 to a recursive graph with some recursive order. Hence ∞-CLIQUE-2 is also Σ_1^1-complete. $\triangle$

Ehrenfeucht-Fraisse-type games are a popular tool for showing non-expressibility of properties of finite structures in monadic E-Σ_1^1 (see the Appendix). The theorems appearing in the Appendix, which are adapted from other people's work on finite structures, hold also for recursive structures. In fact, the proofs can be applied here with almost no change. We now use this technique to exhibit some Σ_1^1-properties that are not expressible in monadic E-Σ_1^1.

Proposition 1.8 ∞-CLIQUE *is not expressible in monadic E-Σ_1^1.*

Proof: Assume that ∞-CLIQUE is expressible in monadic E-Σ_1^1. By Prop. A.3 of the Appendix, there are c, r such that the spoiler has a winning strategy in the Ajtai-Fagin (c, r)-game over ∞-CLIQUE.

Let c, r be given. We show that the duplicator can win this game. The duplicator chooses G_0 to be a recursive graph consisting of an infinite clique and an infinite independent set. The spoiler now colors G_0 with colors $C_1, \ldots, C_c$. Assume that i_j points from the clique have been colored with C_j, for $1 \leq j \leq c$ (here i_j can be ∞). The duplicator now chooses G_1 to be a graph consisting of a finite clique of size $c \cdot r$ and an infinite independent set. He first colors the independent set in the same way as the spoiler colored the independent set of G_0. Now, for each $1 \leq j \leq c$, if $i_j \leq r$ the duplicator colors i_j points from the clique with C_{i_j}, and if $i_j > r$ he colors only r points with C_{i_j}. The rest of the points are colored with a color that appears more than r times in the clique. They now play the r-game, and clearly the duplicator can win. $\triangle$

Proposition 1.9 $\overline{\infty\text{-CLIQUE}}$ *is not expressible in monadic E-Σ_1^1.*

Proof: Given c and r, the duplicator chooses G_0 to be a recursive graph consisting of a clique H of size $c \cdot r$ and an infinite independent set. After the spoiler colors G_0, the duplicator chooses G_1 to be a recursive graph consisting of an infinite clique and an infinite independent set, and colors it as follows. For each $1 \leq j \leq c$, the duplicator colors i_j points of the clique with C_j if the spoiler colored $i_j \leq r$ points of H with C_j. Otherwise, he colors r points of the clique with C_j. The rest of the points of the clique are colored with a color appearing at least r times in H. The independent set is colored in the same way as the independent set of G_0. The duplicator can now win the r-game. $\triangle$

Denote by ISO the problem of whether two recursive graphs are isomorphic.

Proposition 1.10 *ISO is not expressible in monadic E-*Σ_1^1.

Proof: Assume that ISO is expressible in monadic E-Σ_1^1. Let c and r be given. We use the graphs appearing in the proof of Fagin, Stockmeyer and Vardi [FSV] that connectivity in finite graphs is not in monadic NP.

Let H_0 be the long cycle chosen by the duplicator in the Ajtai-Fagin (c, r)-game appearing in their proof. In our game, the duplicator chooses the pair of isomorphic graphs to be $\langle G_0, G_0 \rangle$, where G_0 is a disjoint union of H_0 and an infinite independent set. The spoiler then colors them with colors $C_1, \ldots C_c$. Let H_1 be a disjoint union of two cycles that were chosen by the duplicator in the game described in [FSV], after the spoiler colors the first cycle. In our game, the duplicator chooses the next pair of non-isomorphic graphs to be $\langle G_1, G_0 \rangle$, where G_1 consists of a disjoint union of H_1 and an infinite independent set. The duplicator then colors H_1 in the same way as the duplicator colored it in the game of [FSV], and he colors the rest of the points in the same way as the spoiler colored the analogous ones in the first pair. Now, the duplicator has a winning strategy in the r-game over the two pairs of graphs. $\triangle$

Proposition 1.11 $\overline{ISO}$ *is not expressible in monadic E-*Σ_1^1.

Proof: Assume that $\overline{\text{ISO}}$ is expressible in monadic E-Σ_1^1. Let c and r be given. We devise a (c, r)-game that uses the graphs used by Fagin [F4] in the (c, r)-game appearing in the proof of the non-containment of connectivity on finite graphs in monadic Σ_1^1.

Let H_0 be the first cycle chosen by the duplicator in Fagin's (c, r)-game, and let H_1 be the second graph chosen by the duplicator, which contains two cycles. Our duplicator chooses a pair of non-isomorphic graphs to be $\langle G_0, G_1 \rangle$, where G_0 (respectively, G_1) is the union of H_0 (respectively, H_1) with an infinite independent set. The second pair of isomorphic graphs chosen by our duplicator is $\langle G_1', G_1'' \rangle$, where $G_1' = G_1'' = G_1$. Now, the spoiler colors the first pair $\langle G_0, G_1 \rangle$, and then the duplicator colors the cycles appearing in G_1' precisely in the same way as the duplicator colored H_1 in the previous game, and he colors the independent set of G_1' in the same way as it was colored in G_0. In addition, he colors G_1'' in the same way as the spoiler colored G_1. Now, the duplicator has a winning strategy in the r-game over the two pairs of graphs. $\triangle$

In [FSV] it is shown that connectivity of finite graphs is not in monadic Σ_1^1. We now show the same for recursive graphs.

Proposition 1.12 *Connectivity of recursive graphs is not expressible in monadic E-*Σ_1^1.

Proof: Assume that connectivity is expressible in monadic E-Σ_1^1. Let $\mathcal{C}$ be the class of connected recursive graphs and let c, r be given. We show that the duplicator wins the Ajtai-Fagin (c, r)-game over $\mathcal{C}$. Let d be the constant existing by Corollary A.5 of the Appendix for this r. The duplicator chooses G_0 to be the graph (V, E), where $V = \{a_i \mid i \in \mathcal{Z}\}$ and $E = \{(a_i, a_{i+1}), (a_{i+1}, a_i) \mid i \in \mathcal{Z}\}$.

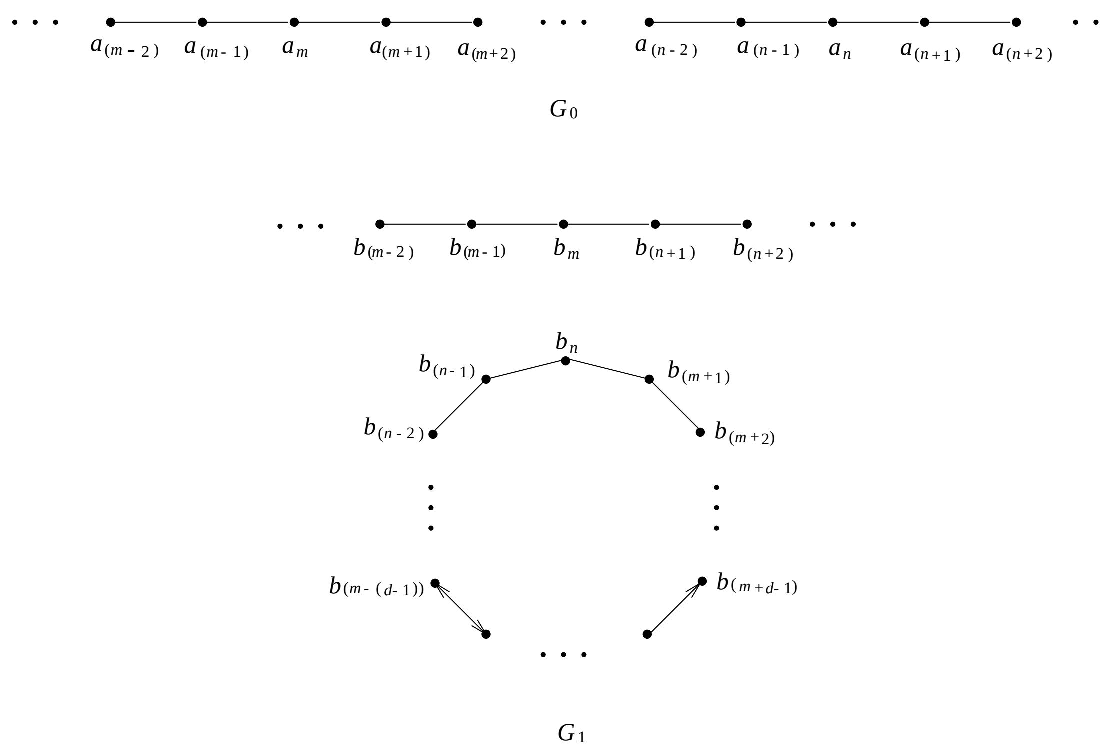

Figure 3: The graphs G_0 and G_1

The spoiler now colors G_0 with c colors. Let $\chi(a_i)$ denote the color of a_i. The d-type of the point a_i in the resulting structure is fully described by the following list of $2d-1$ colors:

$$\langle \chi(a_{i-(d-1)}), \ldots, \chi(a_{i-1}), \chi(a_i), \chi(a_{i+1}), \ldots, \chi(a_{i+(d-1)}) \rangle$$

The number of possible d-types is $c^{(2d-1)}$, which is a constant. So there must exist points a_m and a_n that have the same d-type and are at least at distance $2d$ apart. (Actually, there are infinitely many such points in G_0.) The duplicator now chooses G_1 to be the graph $G_1 = (V', E')$, where $V' = \{b_i \mid i \in \mathcal{Z}\}$ and $E' = E_1 \cup E_2$, with:

$$\begin{aligned} E_1 &= \{(b_i, b_{i+1}), (b_{i+1}, b_i) \mid i < m \ \lor \ i > n\} \cup \{(b_m, b_{n+1}), (b_{n+1}, b_m)\} \\ E_2 &= \{(b_j, b_{j+1}), (b_{j+1}, b_j) \mid m < j < n\} \cup \{(b_n, b_{m+1}), (b_{m+1}, b_n)\}. \end{aligned}$$

Figure 3 illustrates these graphs. The duplicator now colors G_1 such that $\chi(b_i) = \chi(a_i)$ for all i. Note that a_i and b_i have the same d-type for all i, so that G_0 and G_1 are d-equivalent. It follows from Corollary A.5 of the Appendix that $G_0 \sim_r G_1$, and hence the duplicator wins the r-game over the two colored graphs. $\triangle$

Disconnectivity of recursive structures can be defined in monadic E-Σ_1^1 as follows:

$$\exists C_1 \exists C_2 \exists x \exists y \forall z_1 \forall z_2 \ (x \in C_1 \ \land \ y \in C_2 \ \land \ ((z_1 \in C_1 \ \land \ z_2 \in C_2) \ \rightarrow \ (z_1, z_2) \notin E))$$

where E is the set of edges of the input graph. This example (and, independently, the d-completeness of monadic E-Σ_1^1 in Σ_1^1) implies that the collection of monadic E-Σ_1^1 properties is not closed under negation.

1.6 Strict E-Σ^1_1

Definition 1.12 *Strict E-Σ^1_1 is the query language containing formulas of the form $\exists\overline{S}\exists\overline{x}\forall\overline{y}\ \phi(\overline{S},\overline{x},\overline{y})$, where ϕ is a quantifier-free formula over some vocabulary σ.*

Proposition 1.13 *Strict E-Σ^1_1 is d-complete for Σ^0_2.*

Proof: Let Q be an r-query over σ whose graph is $Gr(Q) = \{(B,\overline{u}) \mid \exists\overline{S}\exists\overline{x}\forall\overline{y}\ \phi(B,\overline{S},\overline{u},\overline{x},\overline{y})\}$, where ϕ is a quantifier-free formula over σ. Let B be an r-st over σ and let $\overline{u}$ be a tuple. Let $\{\overline{x}_1,\overline{x}_2,\ldots\}$ and $\{\overline{y}_1,\overline{y}_2,\ldots\}$ be all the feasible values of $\overline{x}$ and $\overline{y}$. Consider the sequence consisting the formulas $\{\phi(B,\overline{S},\overline{u},\overline{x}_i,\overline{y}_j) \mid i,j \geq 1\}$.

Recall that ϕ is quantifier-free and contains terms of the forms $R(\overline{w})$, for $R \in B$, and $S(\overline{z})$, for $S \in \overline{S}$. Here $\overline{w}$ and $\overline{z}$ are projections of $\overline{u}$, $\overline{x}$ and $\overline{y}$. Since B is given, we may view ϕ as a Boolean formula with "variables" of the form $S(\overline{z})$. Now, in order to determine whether $\overline{u} \in Q(B)$, it suffices to find some $\overline{x}_i$ for which all the formulas $\{\phi(B,\overline{S},\overline{u},\overline{x}_i,\overline{y}_j)\}_{j=1}^{\infty}$ are satisfied by a common $\overline{S}$. We now show how, after fixing some i, checking the existence of a common $\overline{S}$ can be reduced to checking the existence of an infinite path in a recursive binary tree, which is co-r.e. It follows that $Gr(Q)$ is in Σ^0_2.

The reduction is carried out as follows. Let $\{\overline{z}_n\}_{n=1}^{\infty}$ be all the variables of $\overline{S}$ in some order. Each $\overline{z}_n$ belongs to some $S \in \overline{S}$, and its rank is that of S. For each n, level n of the tree is related to $\overline{z}_n$. Branching left or right represents evaluating the corresponding $S(\overline{z}_n)$ to 0 or 1. A node on level n is determined to be a leaf if evaluating $S(\overline{z}_1),\ldots,S(\overline{z}_n)$ with the values according to the corresponding path, yields a false formula among the first n formulas, $\phi(B,\overline{S},\overline{u},\overline{x}_i,\overline{y}_j)$, for $1 \leq j \leq n$. Note that the tree is recursive and contains an infinite path iff there exists a "good" evaluation for all the variables.

We claim that the completeness follows from the d-completeness of E-Σ^0_2 for Σ^0_2. Since E-Σ^0_2 is contained in strict E-Σ^1_1, the existing Σ^0_2-complete query that is expressible in E-Σ^0_2 is also expressible in strict E-Σ^1_1. In particular, let σ be a vocabulary containing a binary and a unary relation symbol. Let Q be an r-query whose graph is $Gr(Q) = \{(R_1,R_2) \mid \exists x\forall y\ (R_1(x,y) \rightarrow \neg R_2(y))\}$. Determining whether a recursive set is finite — a well known Σ^0_2-complete problem — is reducible to $Gr(Q)$, as follows. Given a recursive set A, check if $(<,A) \in Gr(Q)$, where $<$ is a recursive linear order. Clearly, $(<,A) \in Gr(Q)$ iff A is finite. $\triangle$

Corollary 1.14 *An r-query that is not in Σ^0_2 cannot be expressed in strict E-Σ^1_1.*

In particular, none of the Σ^1_1-complete problems over recursive structures (for example, those appearing in [HH1]) are expressible in strict E-Σ^1_1. The 0–1 laws for recursive structures introduced in Part 2 of the paper constitute an additional tool for showing non-expressibility of properties in strict E-Σ^1_1. Several examples of this are given later, and among them is EA, which is in Π^0_2. Actually, we will show that EA is not expressible by any E-Π^1_1 formula over σ, hence $\overline{\text{EA}}$ is not expressible in strict E-Σ^1_1, although it is in Σ^0_2.

Corollary 1.15 *Strict E-Σ^1_1 does not express all the Σ^0_2 generic r-queries.*

Kolaitis and Vardi [KV1] proved that Hamiltonicity and connectivity of finite graphs cannot be expressed in strict E-Σ^1_1. We show that this is true for recursive structures too. For Hamiltonicity, this follows from Corollary 1.14 and the fact that the problem is Σ^1_1-complete [H]. The proof for connectivity is similar to that of [KV1] for the finite case:

Proposition 1.16 *Connectivity of recursive graphs is not expressible in strict E-Σ^1_1.*

Proof: Let $\psi = \exists\overline{S}\exists\overline{x}\forall\overline{y}\phi(\overline{S},\overline{x},\overline{y})$ be a strict E-Σ_1^1 formula defining connectivity on recursive structures.

Each graph that satisfies the 2-extension axioms is connected. In particular, the countable random graph $\mathbf{A}$ is connected, and hence ψ has to be true on $\mathbf{A}$. As in the proof of Lemma 2.8, let A_0 be the finite subgraph of $\mathbf{A}$ with universe consisting of the elements $\{a_1,\ldots,a_n\}$ that witness the first-order existential quantifiers $\exists\overline{x}$ in $\mathbf{A}$.

Let G be a non-connected recursive countable graph containing A_0. G is isomorphic to a subgraph of $\mathbf{A}$ because $\mathbf{A}$ is universal. Since universal statements are preserved under subgraphs and $\mathbf{A}$ satisfies ψ, G satisfies ψ too ($\overline{x}$ is interpreted by $\overline{a}$ and $\overline{S}$ is interpreted by the restriction to G of the relations on $\mathbf{A}$ that witness the existential second-order quantifiers). This contradicts the fact that G is not connected. $\triangle$

Now, since the monadic E-Σ_1^1 formula given above for disconnectivity is also in strict E-Σ_1^1, the class of properties expressible in strict E-Σ_1^1 is not closed under negation.

1.7 Iterative Logic

Iterative logic is obtained by adding **while** loops to first-order logic [CH2, KV1]. We shall not provide a formal definition here, except to say that formulas in iterative logic are of the form $(X)S$, where S is a program and X is a relation variable containing the answer.

Proposition 1.17 *The data-complexity of iterative logic is Δ_1^1.*

Proof: Let Q be an r-query over σ expressed by a formula $(X)S$ in iterative logic. Note that $(B,\overline{x}) \in Gr(Q)$ if the program S terminates when applied to B, and upon termination $\overline{x} \in X$. If S does not terminate when applied to B, or if $\overline{x} \notin X$ upon termination, $(B,\overline{x}) \notin Gr(Q)$. These possibilities can be expressed as follows:

$$\begin{aligned} (B,\overline{x}) \in Gr(Q) \quad &\text{iff} \quad \exists\overline{P}\ (\psi(\overline{P},B)\ \wedge\ \phi(\overline{x},\overline{P},B)) \\ (B,\overline{x}) \notin Gr(Q) \quad &\text{iff} \quad \exists\overline{P}\ (\psi'(\overline{P},B)\ \vee\ (\psi(\overline{P},B)\ \wedge\ \neg\phi(\overline{x},\overline{P},B))), \end{aligned}$$

where $\psi(\overline{P},B)$ checks that $\overline{P}$ is a legal terminating computation of S on B, $\psi'(\overline{P},B)$ checks that $\overline{P}$ is a legal infinite computation of S on B, and $\phi(\overline{x},\overline{P},B)$ checks that $\overline{x}$ is contained in the answer.

Here is how $\overline{P}$ represents computations. If $X_1,\ldots,X_k$ are the variables appearing in the program S, $\overline{P}$ will be a sequence of second-order variables $P_1,\ldots,P_k$, such that each P_i represents the contents of X_i during the computation. If X_i's rank is n, P_i's will be $n+2$. Now, $(j,k,x_1,\ldots,x_n) \in P_i$ iff j is the index of a configuration, k is a location in the program S, and $(x_1,\ldots,x_n)$ is contained in X_i in the j-th configuration. The set of all tuples in P_i with the same configuration index represents the content of X_i in the relevant configuration.

Since all the internal conditions are arithmetical, $Gr(Q) \in \Delta_1^1$. $\triangle$

Iterative logic does not express all the Δ_1^1 r-queries. For example, we show in Section 2.5 that iterative logic cannot express the arithmetical property stating that a recursive graph has a clique of size k, for arbitrarily large k.

Iterative logic is closed under negation, since for every program in iterative logic we can easily construct a program that simulates the original and then computes the complement of the result.

1.8 Fixpoint logic

Fixpoint logic is obtained by adding least fixpoint operators to first-order formulas (see [CH2, I, Mos]). FP_1 denotes positive fixpoint logic, which is obtained from first-order logic by adding the least fixpoint operator to positive formulas. FP denotes the hierarchy obtained by combining the least fixpoint operator with the formation rules of first-order logic. It is clearly closed under negation.

Proposition 1.18 *FP_1 is d-complete for Π^1_1.*

Proof: Let Q be an r-query over σ expressible in FP_1. $Gr(Q)$ can be expressed as follows [Mos]:

$$Gr(Q) = \{(B, \overline{x}) \mid \forall S((\forall \overline{x}'[\phi(\overline{x}', S, B) \rightarrow \overline{x}' \in S]) \rightarrow \overline{x} \in S)\},$$

where ϕ is a first-order formula over σ. This expression is in E-Π^1_1, hence $Gr(Q)$ is in Π^1_1.

In addition, we exhibit a Π^1_1-complete query that is expressible in FP_1. Let $\sigma = \{T\}$, where T is a binary relation symbol. Let Q be a query yielding the set of nodes from which there is no infinite path. We thus have:

$$Gr(Q) = \{(T, x) \mid \ \forall y((x, y) \notin T \ \vee \ ((x, y) \in T \rightarrow (T, y) \in Gr(Q)))\}.$$

Since $(T, y) \in Gr(Q)$ appears positively, this formula is in FP_1. $Gr(Q)$ is Π^1_1-complete, since the well-foundedness of recursive trees, the quintessential Π^1_1-complete problem [R], is trivially reducible to it. $\triangle$

For the next proposition recall that by Corollary 1.6 the generic queries in the complexity class Δ^1_2 of the analytic hierarchy are precisely those queries expressible in E-Σ^1_2 and in E-Π^1_2.

Proposition 1.19 *The data complexity of FP is Δ^1_2.*

Proof: Let $\sigma = (P_1, \ldots, P_k)$. We prove the claim by induction on the number of compositions of least fixpoint operations appearing in the formula. If a query Q is expressed by a first-order formula, or by applying a single least fixpoint operation to a positive first-order formula, then Q is expressible in E-Π^1_1, and hence it is in Δ^1_2. Also, closures of E-Π^1_1 formulas under first-order operations are either in E-Π^1_1 or in E-Σ^1_1.

Assume now that any query over σ that is expressible by applying n compositions of least fixpoint operations to a first-order formula is in Δ^1_2, and that the same is true for the closure of such formulas under first-order operations. Let Q be an r-query obtained by applying a least fixpoint operation to $\tau(\overline{x}, S, B, R)$, where τ is first-order, B denotes the structure over σ and R is expressible both in E-Σ^1_2 and in E-Π^1_2. As in the previous proof, $Gr(Q)$ can be expressed as follows:

$$\begin{aligned} Gr(Q) &= \{(B, \overline{x}) \mid \forall S\ ((\forall \overline{x}'[\tau(\overline{x}', S, B, R) \rightarrow \overline{x}' \in S]) \rightarrow \overline{x} \in S)\} \\ &= \{(B, \overline{x}) \mid \psi_1(\overline{x}, B, R)\}. \end{aligned}$$

R can be expressed as follows:

$$R = \{\overline{x} \mid \forall E \exists F\ \phi_1(\overline{x}, E, F, B)\} = \{\overline{x} \mid \exists C \forall D\ \phi_2(\overline{x}, C, D, B)\},$$

where ϕ_1 and ϕ_2 are first-order. (For simplicity we refer to E and F as single second-order variables instead of tuples thereof.)

Now, in order to show that $Gr(Q)$ can be expressed in both E-Σ^1_2 and E-Π^1_2, we first translate ψ_1, the formula expressing $Gr(Q)$, into prenex disjunctive normal form, ψ_2. Now $Gr(Q)$ can be expressed in two ways as follows:

1. Each positive appearance of $\overline{x} \in R$ in ψ_2 is replaced by $\forall E \exists F\ \phi_1(\overline{x}, E, F, B)$, and every appearance of $\overline{x} \notin R$ is replaced by $\forall C \exists D\ \neg\phi_2(\overline{x}, C, D, B)$. If there is more than one appearance of $\overline{x} \in R$ or $\overline{x} \notin R$, then we use different names for E, F, C and D, such as $E_1, E_2, \ldots$ for the appearances of E, and $F_1, F_2, \ldots$ for F, etc. We then rearrange the quantification, obtaining:

   $$\forall S \forall E_1 \exists F_1 \forall E_2 \exists F_2 \cdots \forall C_1 \exists D_1 \forall C_2 \exists D_2 \cdots \psi_3(\cdots)$$

   Since the only dependencies between the variables are those of the F's on the E's and the D's on the C's, we may change the order of the quantifiers, obtaining the following E-Π^1_2 formula:

   $$\forall S \forall E_1 \forall E_2 \cdots \forall C_1 \forall C_2 \cdots \exists F_1 \exists F_2 \cdots \exists D_1 \exists D_2 \cdots \psi_3(\cdots).$$

2. Each appearance of $\overline{x} \in R$ in ψ_2 is replaced by $\exists C \forall D\ \phi_2(\overline{x}, C, D, B)$, and each appearance of $\overline{x} \notin R$ is replaced by $\exists E \forall F\ \neg\phi_1(\overline{x}, E, F, B)$. Proceeding like in step 1, we get:

   $$\forall S \exists E_1 \exists E_2 \cdots \exists C_1 \exists C_2 \forall F_1 \forall F_2 \cdots \forall D_1 \forall D_2 \cdots \psi_4(\cdots).$$

   Since no second-order variables depend on S, we can move the quantification of S, to obtain the E-Σ^1_2 formula:

   $$\exists E_1 \exists E_2 \cdots \exists C_1 \exists C_2 \forall S \forall F_1 \forall F_2 \cdots \forall D_1 \forall D_2 \cdots \psi_4(\cdots).$$

Thus, Q is expressible in E-Σ^1_2 and in E-Π^1_2, and by Corollary 1.6, $Gr(Q) \in \Delta^1_2$. Clearly, the closures of sets in Δ^1_2 under first-order operations are also in Δ^1_2. $\triangle$

Proposition 1.20 *There is a query expressible in FP that is hard for both Σ^1_1 and Π^1_1.*

Proof: Let $\sigma = \{T, E\}$ be a vocabulary for alternation trees, where T is binary and E is unary. The nodes in E are $\exists$-nodes and the others are $\forall$-nodes. Let Q_1 be a query, which, for an alternation tree, yields all $\forall$-nodes and all nodes from which there is no infinite path of $\exists$-nodes. We may thus write:

$$Gr(Q_1) = \{(T, E, x) \mid x \notin E \ \vee\ \forall y\ ((x, y) \in T \rightarrow (T, E, y) \in Gr(Q_1))\}.$$

Let Q_2 be a query that yields all the $\forall$-nodes that are leaves or all of whose successors are in $Q_2(T, E)$, and all the $\exists$-nodes that have some successor in $Q_2(T, E)$ or have an infinite path of $\exists$-nodes emanating from them.

$$\begin{aligned} Gr(Q_2) \ &= \ \{(T, E, x) \mid (x \notin E \ \wedge\ \forall y\ ((x, y) \notin T \ \vee\ ((x, y) \in T \rightarrow (T, E, y) \in Gr(Q_2)))) \\ &\qquad \vee\ (x \in E \ \wedge\ ((\exists y (x, y) \in T \ \wedge\ (T, E, x) \in Gr(Q_2)) \ \vee\ (T, E, x) \notin Gr(Q_1)))\} \end{aligned}$$

$Gr(Q_2)$ can be expressed by applying a fixpoint operator to the negation of a fixpoint operator, and hence is expressible in FP. In addition, $Gr(Q_2)$ is hard for both Π^1_1 and Σ^1_1, since a recursive tree T over $\mathcal{N}$ is well-founded iff $(T, \emptyset, root) \in Gr(Q_2)$, and it is not well-founded iff $(T, \mathcal{N}, root) \in Gr(Q_2)$. $\triangle$

In Section 2.4 we show that the 0–1 law proposed for recursive structures holds for FP, and there are properties (even arithmetical ones) for which the 0–1 law does not hold. Hence, FP does not express all the generic Δ^1_2 queries.

Over finite structures many thing are known about FP and iterative logic. For example, iterative logic in no weaker than FP. Also, FP is d-complete for PTIME and iterative logic is d-complete for PSPACE [CH2]. Actually, iterative logic is equivalent to FP iff PTIME = PSPACE [AV1, AV2]. If structures have a built-in order, then FP expresses exactly PTIME and iterative logic expresses exactly PSPACE [I, V2]. In contrast, over recursive structures, FP is stronger than iterative logic. The reason is that infinite **while** loops can yield undefined r-queries, whereas formulas in FP are everywhere defined even when the fixpoint operator is applied for ω iterations. An example is the query Q appearing in the proof of Prop. 1.18, that yields the set of nodes in a recursive tree that have no infinite path emanating from them. When Q is applied to the tree in Figure 4, it actually carries out ω iterations.

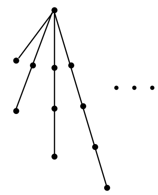

Figure 4: A simple tree

1.9 Turing machines with oracles for emptiness

Definition 1.13 *An emptiness oracle is an oracle, which, given a code for a relation R (recursive relations obviously have finite codes), decides whether $R = \emptyset$. We denote by $R^\emptyset$ the smallest set of relations that contains the recursive ones and is closed under the use of emptiness oracles. A $TM^\emptyset$ is a Turing machine, which, given codes for the Turing machines of the relations of the input structure and a tuple u, may use emptiness oracles for relations in $R^\emptyset$ to decide whether u is in the output relation. A $GTM^\emptyset$ is a $TM^\emptyset$ that computes a generic r-query.*

Proposition 1.21 *In the framework of $TM^\emptyset$'s, projections and emptiness checking are interchangable, in the sense that one can be computed given the other.*

Proof: For a recursive relation R of rank n, let $R\downarrow$ be defined as in [CH1] to project out the first component of R. In order to check whether $R = \emptyset$, simply check whether $() \in R{\downarrow}^n$, where $()$ is the unique tuple of rank 0. For the other direction, note that

$$\begin{aligned} (x_1, x_2, \ldots, x_{n-1}) \in R\downarrow \quad &\Longleftrightarrow \quad \exists x_0\ (x_0, x_1, \ldots, x_{n-1}) \in R \\ &\Longleftrightarrow \quad D \times \{(x_1, x_2, \ldots, x_{n-1})\} \cap R \neq \emptyset \qquad \triangle \end{aligned}$$

Among the operations of the relational algebra [C], it is the projections (which are really just existential quantifiers) that do not preserve recursiveness of relations [HH2]. Hence, once oracles for checking emptiness are present, relational algebra becomes computable. The equivalence of relational algebra and first-order logic [C] then yields:

Corollary 1.22 *First-order logic is computable over r-st's with an oracle for emptiness.*

Thus, a query computed by a $\mathrm{GTM}^\emptyset$ will be recursive over r-st's that have oracles to check the emptiness of relations, but without such oracles, $\mathrm{GTM}^\emptyset$'s can compute non-recursive queries. In fact, since Prop. 1.21 implies that projections are $\mathrm{GTM}^\emptyset$-computable, we have:

Corollary 1.23 *$GTM^\emptyset$'s compute all arithmetical generic r-queries.*

The non-recursive relations that $\mathrm{GTM}^\emptyset$'s may give rise to, can be represented on the tape by finite strings that describe the operations leading to them from the original recursive relations.

Proposition 1.24 *The data complexity of queries expressible by $GTM^\emptyset$'s is Δ^1_1.*

Proof: Let Q be an r-query expressible by a GTM$^{\emptyset}$ M. We may write:

$$\begin{aligned} Gr(Q) &= \{(B,\overline{x}) \mid \overline{x} \in Q(B) \;\wedge\; Q(B) \text{ is defined}\} \\ \overline{Gr(Q)} &= \{(B,\overline{x}) \mid \overline{x} \in Q(B)\} \cup \{(B,\overline{x}) \mid Q(B) \text{ is undefined}\}. \end{aligned}$$

Now, $(B,\overline{x}) \in Gr(Q)$ iff there is a finite computation of M on $(B,\overline{x})$ that accepts $\overline{x}$. In addition, $(B,\overline{x}) \notin Gr(Q)$ iff there is a finite computation of M on $(B,\overline{x})$ that does not accept $\overline{x}$ or an infinite computation of M on $(B,\overline{x})$.

The computations assumed to exist in these statements may contain infinite relations that M can check for emptiness. Since all the internal conditions are arithmetical, $Gr(Q) \in \Sigma^1_1 \cap \Pi^1_1 = \Delta^1_1$. $\triangle$

Proposition 1.25 *There is a non-arithmetical generic query that is computed by a GTM*$^{\emptyset}$.

Proof: Let A be a recursive set. Consider $A^{(\omega)} = \{\langle x,y\rangle \mid x \in A^{(y)}\}$, which is defined in [R], where $A^{(y)}$ is defined by $A^{(0)} = A$ and $A^{(n+1)} = (A^{(n)})'$. Here,

$$A' = \{x \mid \varphi^A_x(x) \text{ is convergent}\} = \{x \mid \exists z \exists w \; U^A(x,z,x,w)\},$$

where $U^A(x,z,u,w)$ is the universal function that checks whether the output of a Turing machine encoded by x is w when it runs z steps on input u using an oracle for A.

We define a query Q that works on r-st's containing a binary relation R:

$$Q(B) = \begin{cases} \emptyset^{(\omega)^*} & \text{if } R \text{ is a total order} \\ \emptyset & \text{otherwise} \end{cases}$$

Here $\emptyset^{(\omega)^*} = \{z \mid \exists\langle x,y\rangle \in \emptyset^{(\omega)} \text{ such that } z \text{ is the } \langle x,y\rangle\text{'th element in the order } R\}$.

Note that Q is generic, and it is also GTM$^{\emptyset}$-computable, since checking if R is a total order is arithmetical. (This does not contradict the fact that total order is not expressible in E-Σ^1_1.) Now, since $\emptyset^{(\omega)}$ is not arithmetical [R], neither is $\emptyset^{(\omega)^*}$, and hence $Gr(Q)$ is not arithmetical either. $\triangle$

In [HH2] we proved that quantifier-free first-order logic is r-complete, i.e., it expresses precisely all the generic and recursive r-queries. In the same spirit we may ask whether there are effective complete languages that capture precisely the arithmetical generic queries, or the Σ^1_1 queries? How about the GTM$^{\emptyset}$-computable queries?

Definition 1.14 *A query language is $\emptyset$-r-complete if it expresses precisely the generic queries that are computable by a GTM*$^{\emptyset}$.

The following is a recursive variant of the notion of completeness for expressing relations due to Bancilhon [B] and Paredaens [P], which was called BP-completeness in [CH1]:

Definition 1.15 *A language for expressing relations is BP-$\emptyset$-r-complete if for each r-st B it expresses precisely the TM$^{\emptyset}$-computable relations that preserve the isomorphisms of B.*

Proposition 1.26 *If the test $(B_1,u) \cong (B_2,v)$, for r-st's B_1, B_2 and tuples u, v, is GTM$^{\emptyset}$-computable, then there is an effective $\emptyset$-r-complete language.*

Proof: Assume that this isomorphism test is indeed GTM$^{\emptyset}$-computable. Given a Turing machine M with an oracle for emptiness, construct a GTM$^{\emptyset}$ M' as follows: on input $(B, \overline{x})$, M' uses the assumed to exist GTM$^{\emptyset}$ to compute the least $(B', \overline{y})$ (in some well ordering) that is isomorphic to $(B, \overline{x})$. M' then accepts the input $(B, \overline{x})$ iff M accepts $(B', \overline{y})$. △

Proposition 1.27 *The following claims are equivalent:*

1. *given two r-st's B_1, B_2, the test $B_1 \cong B_2$ is GTM$^{\emptyset}$-computable;*
2. *given an r-st B and tuples u and v, the test $(B, u) \cong (B, v)$ is GTM$^{\emptyset}$-computable;*
3. *given two r-st's B_1 and B_2 and tuples u and v, the test $(B_1, u) \cong (B_2, v)$ is GTM$^{\emptyset}$-computable;*
4. *there is an effective BP-$\emptyset$-r-complete language.*

Proof: It is easy to verify that (1) $\rightarrow$ (3) and (3) $\rightarrow$ (2).

We now show (2) $\rightarrow$ (1). Assuming (2), let $B_1 = (D_1, R_1, \ldots, R_k)$ and $B_2 = (D_2, R'_1, \ldots, R'_k)$ be r-st's. We may assume that the domains D_1 and D_2 are disjoint. Consider the following r-st:

$$B = (D_1 \cup D_2 \cup \{a, b\}, S, Rel_1, \ldots, Rel_k, \{a, b\})$$

where $S = (\{a\} \times D_1) \cup (\{b\} \times D_2)$ and $Rel_i = (\{a\} \times R_i) \cup (\{b\} \times R'_i)$. It follows that $(B, a) \cong (B, b)$ iff $B_1 \cong B_2$. By the assumption, there is a GTM$^{\emptyset}$ that computes B and checks that $(B, a) \cong (B, b)$.

We now show (3) $\rightarrow$ (4). By Prop. 1.26, assuming (3) is true there is an effective $\emptyset$-r-complete language; denote it by $\mathcal{L}$. We show that $\mathcal{L}$ is also BP-$\emptyset$-r-complete. Clearly, for each $\phi \in \mathcal{L}$ and r-st B, $\phi(B)$ preserves the isomorphisms of B. For the other direction, let B be some r-st, and let R be a relation that preserves the isomorphisms of B, and is computable by some TM$^{\emptyset}$ M. We have to show that there is an expression in $\mathcal{L}$ that yields R when applied to B. Define the following query:

$$Q(B') = \begin{cases} R & \text{if } B' = B \\ R' & \text{if } B' \cong B \\ \emptyset & \text{otherwise} \end{cases}$$

Here, R' is the relation computed by the following TM$^{\emptyset}$, M': given B' and input $\overline{x}$, M' computes the least $\overline{y}$ satisfying $(B', \overline{x}) \cong (B, \overline{y})$, and then accepts $\overline{x}$ iff M accepts $\overline{y}$. M' can check $(B', \overline{x}) \cong (B, \overline{y})$ by the assumption.

By statement (1) (which we can assume, since we already know (3) $\rightarrow$ (1)), the test $B' \cong B$ is GTM$^{\emptyset}$-computable, and so is the test $B' = B$, which is arithmetical. In addition, R and R' are TM$^{\emptyset}$-computable. Since Q is also generic, there is an expression in $\mathcal{L}$, denoted ϕ, that expresses Q. But then $\phi(B) = R$.

Finally to prove (4) $\rightarrow$ (2), assume that there is an effective BP-$\emptyset$-r-complete language $\mathcal{L}$. Let B be some r-st, and let u and v be tuples over $D(B)$. Define $B' = B \cup \{R\}$, where R is the relation $\{u, v\}$. Similarly to the proof of Prop. 6.1 of [HH2], in which we show that there is no BP-r-complete language, in order to determine that $(B, u) \not\cong (B, v)$, it suffices to check that there exists an expression $\psi \in \mathcal{L}$ such that $u \in \psi(B')$ but $v \notin \psi(B')$. This is GTM$^{\emptyset}$-computable, since the search is arithmetical and ψ is GTM$^{\emptyset}$-computable. △

Now, the isomorphism problem for recursive structures is known to be Σ^1_1-complete [Mo], and since GTM$^{\emptyset}$'s compute only queries in Δ^1_1, they cannot be used for the isomorphism problem. We may thus conclude:

Corollary 1.28 *There is no effective BP-$\emptyset$-r-complete language.*

The problem of whether there is an effective $\emptyset$-r-complete language remains open.

2 Zero-One Laws

2.1 Background and Overview

Many of the classical theorems of model theory fail for finite models [G1, G2], and also for recursive ones. For example, in Section 2.2 we show that this is the case for the completeness theorem and the compactness theorem. (This, and other failures of classical theorems, was shown independently also by Stolboushkin; see [S].) In contrast, the technique of Ehrenfeucht/Fraisse games [E, Fr] holds for these classes too. Another important technique, considered unique to finite model theory, is that of 0–1 laws, which concern the asymptotic probability of properties of finite structures. (In the following, the parts concerning finite structures are adapted from [KV1].)

If C is a class of finite structures over some vocabulary σ, and if P is a property of some of the structures in C, then the *asymptotic probability* $\mu(P)$ on C is the limit as $n \to \infty$ of the fraction of the n-element structures in C that satisfy P, provided that the limit exists. That is, to obtain $\mu(P)$ on C, we divide the number of n-element structures in C that satisfy P by the total number of n-element structures in C. We say that P is *true almost everywhere* on C if $\mu(P) = 1$. If $\mu(P) = 0$, we say that P is *false almost everywhere* on C.

Fagin [F3] and Glebskii et al. [GKLT] were the first to establish a fascinating connection between logical definability and asymptotic probabilities. More specifically, they showed that if C is the class of all finite structures over some relational vocabulary and if P is any property expressible in first-order logic, then $\mu(P)$ exists and is either 0 or 1. This result, which is known as the *0–1 law for first-order logic*, became the starting point of a series of investigations aimed at discovering the relationship between expressibility in a logic and asymptotic probabilities. Several additional logics, such as fixpoint logic, iterative logic and strict E-Σ^1_1, have been shown to satisfy the 0–1 law [BGK, TK, KV1, KV2, KV3]. See also the survey by Compton [Co].

A standard method for establishing 0–1 laws on finite structures, originating in Fagin [F3], is to prove that the following *transfer theorem* holds: there is an infinite structure $\mathbf{A}$ over σ such that for any property P expressible in L we have:

$$\mathbf{A} \models P \text{ iff } \mu(P) = 1 \text{ on } C.$$

It turns out that there is a common countable structure $\mathbf{A}$ that satisfies this equivalence for first-order logic, fixpoint logic, iterative logic and strict E-Σ^1_1. Moreover, this structure $\mathbf{A}$ is characterized by an infinite set of *extension axioms*, which, intuitively, assert that any *type* can be *extended* to obtain any other type. In other words, for each finite set X of points, and each possible way that a point $y \notin X$ could relate to X in terms of atomic formulas over the appropriate vocabulary, there is an extension axiom stating that indeed such a point exists. For example, the following is an extension axiom over a vocabulary containing one binary relation symbol R:

$$\forall x_1 \forall x_2 \; (\; x_1 \neq x_2 \;\Rightarrow$$
$$\exists y \; (y \neq x_1 \;\wedge\; y \neq x_2 \;\wedge\; (y, x_1) \in R \;\wedge\; (x_1, y) \notin R \;\wedge\; (y, x_2) \notin R \;\wedge\; (x_2, y) \in R) \;)$$

Fagin [F3] realized that the extension axioms are relevant to the study of probabilities on finite structures and proved that on the class C of all finite structures of vocabulary σ, $\mu(\tau) = 1$ for any extension axiom τ. The theory of all extensions axioms, denoted T, was proven to be ω-categorical (that is, every two countable models are isomorphic) by Ehrenfeucht and Ryll Nardzeweski [Ry] and by Gaifman [Ga]. Hence, $\mathbf{A}$, which is a model for T, is unique up to isomorphism. This structure is called the *random countable structure* (or the *random countable graph* if σ has just one binary relation symbol), since it is generated, with probability 1, by a random process in which we start with a countable set of points, and where each possible tuple appears with probability $\frac{1}{2}$, independently of the other tuples. The random graph was studied by Rado [Ra], and is sometimes called the *Rado graph.*

Since all countable structures of type σ are isomorphic to $\mathbf{A}$ with probability 1, the asymptotic probability of any (generic) property P on countable structures has to be 0 or 1, depending on whether $\mathbf{A}$ satisfies P or not. Hence, the subject of 0–1 laws over all countable structures is not interesting.

Turning to recursive structures, one is faced with the difficulty of defining asymptotic probabilities, since structure size is no longer applicable. In Section 2.3 we address this issue, suggesting an extension of 0–1 laws to recursive structures. First, we define a *T-sequence* $\mathcal{F}$ of recursive structures to be a sequence in which all the extension axioms are almost surely true. For a property P we then define $\mu_{\mathcal{F}}(P)$ to be the asymptotic fraction of the structures that satisfy P as we make progress along $\mathcal{F}$. We then define the 0–1 law for recursive structures relative to a logic L, if $\mu_{\mathcal{F}}(P)$ exists and is equal to 0 or 1 for all properties P expressible in L, and for any T-sequence $\mathcal{F}$. In Section 2.4 we show that this law holds for first-order logic, iterative logic, fixpoint logic, strict E-Σ_1^1 and strict E-Π_1^1, but not for general E-Σ_1^1.

Our results can be used to show non-expressibility in several logics. For example, in Section 2.5 we show that the following properties are not expressible in fixpoint logic, iterative logic, strict E-Σ_1^1 or E-Π_1^1: ∞-CLIQUE, which requires the existence of an infinite clique in a recursive structure; EA, which states that a recursive structure satisfies all the extension axioms, and ISO, which states that two recursive structures are isomorphic.

2.2 Completeness and compactness fail

The completeness theorem on general structures states that if ψ is a first-order sentence and Σ is a set of first-order sentences, then $\Sigma \models \psi \iff \Sigma \vdash \psi$. The compactness theorem on general structures, which follows from completeness, states that a set Σ of first-order sentences is satisfiable if every finite subset of Σ is satisfiable. The completeness and compactness theorems fail for finite structures (see [F4]). Grumbach and Su show in [GS] that they fail also for finitely representable models, which are different from recursive structures. Here we show that these theorems fail also for recursive structures, as proved independently by Stolboushkin [S].

Proposition 2.1 *The compactness theorem fails for recursive structures.*

Proof: As a counter example, let B be a non-recursive set over $\mathcal{N}$, and let $\sigma = \{A, <, S, U\}$, where $<$ and S are binary and A and U are unary. For each integer k, we define ψ_k to be the following first-order sentence:

$$\exists x_1 \cdots \exists x_k \forall y \forall z \ \Big((y \in A \leftrightarrow y = x_1) \ \wedge \ order(<) \ \wedge suc(S) \ \wedge$$
$$\neg(y < x_1) \ \wedge \ (x_1, x_2) \in S \ \wedge \ \cdots \ \wedge \ (x_{k-1}, x_k) \in S \ \wedge$$
$$((z \neq x_1 \ \wedge \ \ldots \ \wedge \ z \neq x_k) \rightarrow (x_k < z)) \ \wedge \ \psi(U, x_1, \ldots, x_k) \Big)$$

where $order(<)$ checks that $<$ is an order (without checking if each subset has a minimal element), $suc(S)$ checks that S is a successor relation, and $\psi(U, x_1, \ldots, x_k)$ is a conjunction of the atomic formulas $x_i \in U$ if $i \in B$, and $x_i \notin U$ if $i \notin B$, for each $1 \leq i \leq k$,

Now, each finite subset of $\{\psi_k\}_{k=1}^{\infty}$ is satisfiable by a recursive structure consisting of an order, successor and an appropriate finite set. However, $\{\psi_k\}_{k=1}^{\infty}$ itself is not satisfiable by any recursive structure, since then B would be recursive. $\triangle$

Proposition 2.2 *The completeness theorem fails for recursive structures.*

Proof: (Adapted from the proof for finite structures given in [F4].) We show that completeness implies compactness on recursive structures too, which yields the result. Assume that the completeness theorem holds for recursive structures; i.e., $\Sigma \models_r \psi$ iff $\Sigma \vdash \psi$, where $\Sigma \models_r \psi$ means that ψ is true for every *recursive* model of Σ, and $\Sigma \vdash \psi$ means that there is a finite proof of ψ from Σ in some effective proof system.

Let Σ be a set of first-order sentences that is not satisfiable in any recursive structure. Then $\Sigma \models_r false$, where $false$ is some logically false sentence such as $p \wedge \neg p$. By the completeness theorem, $\Sigma \vdash false$. Since the proof is of finite length, only a finite subset $\Sigma' \subseteq \Sigma$ is used in the proof. So $\Sigma' \vdash false$. Again, by the completeness theorem, $\Sigma' \models_r false$. Since $false$ is not satisfiable, Σ' is not satisfiable by a recursive structure.

It follows that if every finite subset Σ' of Σ is satisfiable by a recursive structure then Σ is also satisfiable by a recursive structure, which is compactness for recursive structures. $\triangle$

2.3 Defining 0–1 laws for recursive structures

Proposition 2.3 *For any given vocabulary σ there is a recursive countable random structure over σ.*

Proof: Let σ be some vocabulary. The required structure is defined in levels as follows. The first level contains the first element of the domain. Assume that levels 1 to n contain the elements $\overline{a} = a_1, \ldots, a_m$, and that we have already defined which tuples over $\overline{a}$ are in the relations of σ. Level $n+1$ is now defined inductively as follows. For every tuple $\overline{x} = (x_1, \ldots, x_n)$, with $\{x_1, \ldots, x_n\} \in \overline{a}$, and for every $(n+1)$-σ-type $s(\overline{x}, z)$ that extends the n-σ-type $t(\overline{x})$, we add a point z to level $n+1$, and define the containment of tuples over $\{x_1, \ldots, x_n, z\}$ in the relations of σ according to type $s(\overline{x}, z)$. Note that there are only finitely many types that extend a given type, so that the construction is recursive. In order to determine whether a tuple $\overline{y}$ is in some relation, it suffices to construct the structure up to a level that contains $\overline{y}$. $\triangle$

We now extend the notion of asymptotic probability to recursive structures.

Definition 2.1 *Let $\mathcal{F} = \{F_i\}_{i=1}^{\infty}$ be a sequence of recursive structures over some vocabulary, and let P be a property of some structures in $\mathcal{F}$. Then the asymptotic probability $\mu_{\mathcal{F}}(P)$ is defined as:*

$$\mu_{\mathcal{F}}(P) = \lim_{n\to\infty} \frac{|\{F_i \mid 1 \leq i \leq n,\ F_i \models P\}|}{n}.$$

We can arrange *all* the recursive structures in sequences $\mathcal{F}$ such that for every property P we will have $\mu_{\mathcal{F}}(P) = 0$ or $\mu_{\mathcal{F}}(P) = 1$. For example, let G be some non-trivial recursive graph. Arrange all the recursive graphs as follows: $G, H_1, G_1, G_2, H_2, G_3, G_4, G_5, \ldots$, where $\{G_i\}_{i=1}^{\infty}$ are all isomorphic to G, and $\{H_i\}_{i=1}^{\infty}$ are all the rest.

Definition 2.2 *Let $\mathcal{F} = \{F_i\}_{i=1}^{\infty}$ be a sequence of recursive structures over some vocabulary σ. We say that $\mathcal{F}$ is a T-sequence if $\mu_{\mathcal{F}}(\tau) = 1$ for every extension axiom τ over σ.*

Example 2.1 *The following are T-sequences:*

1. A sequence of graphs that are all isomorphic to the countable random graph **A**. We shall use U to denote one of these sequences.

2. $\mathcal{F} = \{F_n\}_{n=1}^{\infty}$, where each F_n is a graph satisfying all the n-extension axioms and is built in levels as follows. The first level contains n distinct points, which form an independent set of size n. Level k is then defined to contain a point z for every set $\{x_1, \ldots, x_n\}$ from lower levels and for every possible extension of it. The new point z is then connected to them according to the extension.

3. $\mathcal{F}' = \{\overline{F_n}\}_{n=1}^{\infty}$, where $\overline{F_n}$ is the complement of F_n (i.e., the edge sets are complements).

Definition 2.3 *Let P be a property of recursive structures. We say that the 0–1 law holds for P if for every T-sequence $\mathcal{F}$, $\mu_{\mathcal{F}}(P)$ exists and is uniformly equal to 0 or 1 (i.e., it is 0 for all $\mathcal{F}$ or 1 for all $\mathcal{F}$). In such a case, we say that P is* almost surely false *or* almost surely true, *respectively. For a logic L, we say that the 0–1 law holds for L if the 0–1 law holds for every property expressible in the logic L.*

According to these definitions, since U is a T-sequence, the transfer theorem holds whenever the 0–1 law holds for L; i.e., $\mathbf{A} \models P$ iff $\mu_{\mathcal{F}}(P) = 1$, for every T-sequence $\mathcal{F}$ and for every property P expressible in L.

2.4 0–1 laws for recursive structures

We use ∞-CLIQUE to denote the class of recursive graphs that contain an infinite clique, and ∞-IND to denote the class of recursive graphs that contain an infinite independent set.

Proposition 2.4 *The 0–1 law on recursive structures does not hold for ∞-CLIQUE and ∞-IND.*

Proof: Let $\mathcal{F} = \{F_n\}_{n=1}^{\infty}$ and $\mathcal{F}' = \{\overline{F}_n\}_{n=1}^{\infty}$ be the T-sequences defined in Example 2.1. Each F_n contains a maximal clique of size $n+1$ and an infinite independent set. Hence, each $\overline{F}_n$ contains a maximal independent set of size $n+1$ and an infinite clique. Therefore, $\mu_{\mathcal{F}}(\infty\text{-CLIQUE}) = 0$, $\mu_{\mathcal{F}}(\infty\text{-IND}) = 1$, $\mu_{\mathcal{F}'}(\infty\text{-CLIQUE}) = 1$, and $\mu_{\mathcal{F}'}(\infty\text{-IND}) = 0$. Thus, according to Def. 2.3, the 0–1 law does not hold. △

Corollary 2.5 *The 0–1 law on recursive structures does not hold for general E-Σ_1^1-properties.*

The next lemma follows from the definition of a T-sequence.

Lemma 2.6 *Let $\mathcal{F}$ be a T-sequence, and let ψ be a conjunction of a finite number of extension axioms. Then $\mu_{\mathcal{F}}(\psi) = 1$.*

Lemma 2.7 [KV1] *Let $\mathbf{A}$ be the countable random structure over the vocabulary σ, and let ϕ be an arbitrary E-Π_1^1 sentence over σ of the form $\forall \overline{S}\theta(\overline{S})$. If $\mathbf{A} \models \phi$, then there is a first-order sentence ψ over σ, such that the 0–1 law holds for ψ and*

$$\models \psi \rightarrow \phi.$$

In particular, every E-Π_1^1 sentence that is true on $\mathbf{A}$ has probability 1 on every T-sequence.

According to the proof in [KV1], ψ is a conjunction of a finite number of extension axioms, which has probability 1 on every T-sequence.

The next lemma extends a lemma of [KV1] from finite structures to countable structures. We write $\models_c \psi$ to denote that ψ is true on all countable structures.

Lemma 2.8 *Let* $\mathbf{A}$ *be the countable random structure over the vocabulary* σ*, and let* ϕ *be an arbitrary strict E-*Σ^1_1 *sentence over* σ *of the form* $\exists \overline{S} \exists \overline{x} \forall \overline{y}\ \theta(\overline{S}, \overline{x}, \overline{y})$*. If* $\mathbf{A} \models \phi$*, then there is a first-order sentence* ψ *over* σ*, such that the 0–1 law holds for* ψ *and*

$$\models_c \psi \rightarrow \phi.$$

In particular, every strict E-Σ^1_1 *sentence that is true on* $\mathbf{A}$ *has probability 1 on every T-sequence.*

Proof: (Based on the proof in [KV1].) Let $\overline{a} = (a_1, \ldots, a_n)$ be a sequence of elements of $\mathbf{A}$ that witness the first-order existential quantifiers $\exists \overline{x}$ in $\mathbf{A}$. Let A_0 be the finite substructure of $\mathbf{A}$ with universe $\{a_1, \ldots, a_n\}$, and let ψ be the conjunction of all the $(n-1)$-extension axioms. Note that every model of ψ contains a substructure A'_0 isomorphic to A_0. Now, assume that B is a countable model of ψ. We can find a substructure B^* of $\mathbf{A}$ that contains A_0 and is isomorphic to B, by mapping A_0 to the appropriate substructure A'_0 of B, and using the extension axioms as follows. Assume that the remaining elements of B are $\{b_0, b_1, \ldots\}$. For each i, map an element in $\mathbf{A}$ to b_i using the $(n+i)$-extension axioms. Since universal statements are preserved under substructures, we conclude that $B \models \exists \overline{S} \exists \overline{x} \forall \overline{y}\ \theta(\overline{S}, \overline{x}, \overline{y})$. Here, $\overline{x}$ is interpreted by $\overline{a}$, and $\overline{S}$ is interpreted by the restriction to B of the relations on $\mathbf{A}$ that witness the existential second-order quantifiers. $\triangle$

Lemma 2.7 implies that the 0–1 law holds for first-order logic, since it is contained in E-Π^1_1 and is closed under negation. In addition, from Lemmas 2.7 and 2.8 it follows that the 0–1 law holds for strict E-Π^1_1 and strict E-Σ^1_1 too:

Proposition 2.9 *The 0–1 law on recursive structures holds for first-order logic, strict E-*Π^1_1 *and strict E-*Σ^1_1 *over a vocabulary* σ*. Moreover, if* $\mathbf{A}$ *is the countable random structure,* P *is a property expressed in first-order logic, strict E-*Σ^1_1 *or strict E-*Π^1_1*, and* $\mathcal{F}$ *is a T-sequence, then* $\mathbf{A} \models P$ *iff* $\mu_{\mathcal{F}}(P) = 1$.

Proposition 2.10 *The 0–1 law on recursive structures holds for iterative logic over a vocabulary* σ*. Moreover, if* $\mathbf{A}$ *is the countable random structure,* P *is a property expressed in iterative logic, and* $\mathcal{F}$ *is a T-sequence, then* $\mathbf{A} \models P$ *iff* $\mu_{\mathcal{F}}(P) = 1$.

Proof: (Based on the proof in [KV1], which generalizes the argument in [BGK] for fixpoint logic. See also [AHV]). We show that every sentence in iterative logic that is defined on all finite structures is in fact equivalent almost surely to a first-order sentence on recursive structures. The proposition then follows from the 0–1 law of first-order logic.

For simplicity, we start with a sentence $\tau = (X)$ **while** Y **do** S, where S is **while**-free. Let $X_1, \ldots, X_k$ be the variables appearing in the program S. There are first-order formulas θ^j_i over σ, for $1 \leq i \leq k$ and $j \geq 0$, such that θ^j_i describes the assignment to X_i after the j-th iteration of S. Consider the sequence $\{S^i(\mathbf{A})\}_{i>0}$, where $S^i(\mathbf{A})$ means applying S on $\mathbf{A}$ for i iterations. After each iteration, the content of each variable in S must be a union of full equivalence classes of $\cong_{\mathbf{A}}$. Since for each n, there are only finitely many equivalence classes of $\cong_{\mathbf{A}}$ for tuples of rank n [Ry, Va, HH2], the sequence $\{S^i(\mathbf{A})\}_{i>0}$ is periodic; i.e., there exists n_0 and p such that for each $n \geq n_0$, $S^n(\mathbf{A}) \equiv S^{n+p}(\mathbf{A})$. (The values of the variables after n_0 iterations are the same as their values after $n_0 + p$ iterations.)

Using this fact, we show that (i) the sequence $\{S^i(\mathbf{A})\}_{i>0}$ converges, and (ii) the sentence $\tau = (X)$ **while** Y **do** S is equivalent almost surely to some first-order sentence.

For proving (i), let $\psi_{ij} = ((\theta^i_1 \equiv \theta^j_1) \wedge \ldots \wedge (\theta^i_k \equiv \theta^j_k))$ be the first-order sentence asserting that the values of $X_1, \ldots, X_k$ after i iterations of S are the same as their values after j iterations. Suppose that $\{S^i(\mathbf{A})\}_{i>0}$ does not converge. The period of the sequence must thus be greater than 1, so there exist

$m, j, l,\ m < j < l$, such that $\mathbf{A}$ satisfies the first-order sentence $\psi = (\psi_{ml} \wedge \neg\psi_{mj})$. By the transfer property for finite structures, ψ is almost surely true on finite structures. It follows that the sequence $\{S^i(I)\}_{i>0}$ diverges almost surely. In particular, there exists a finite I for which $\{S^i(I)\}_{i>0}$ diverges. But we assume that τ converges for all finite inputs. Hence, $\{S^i(\mathbf{A})\}_{i>0}$ converges.

Now, by (i), the sequence $\{S^i(\mathbf{A})\}_{i>0}$ becomes constant after finitely many iterations, say n_0. Let ϕ^{n_0} be a first-order formula over σ that is equivalent to $(X)S^{n_0}$. Suppose $\mathbf{A}$ satisfies τ. Thus, $\mathbf{A}$ satisfies ϕ^{n_0}, since $\{S^i(\mathbf{A})\}_{i>0}$ becomes constant at the n_0-th iteration. Furthermore, $\mathbf{A}$ satisfies $\psi_{n_0(n_0+1)}$. Thus $\mathbf{A}$ satisfies $\phi^{n_0} \wedge \psi_{n_0(n_0+1)}$. By the transfer property for first-order logic, $\phi^{n_0} \wedge \psi_{n_0(n_0+1)}$ is almost surely true for finite, and also for recursive, structures. For each structure I for which $\psi_{n_0(n_0+1)}$ holds, $\{S^i(I)\}_{i>0}$ converges after n_0 iterations, so τ is equivalent to ϕ^{n_0}. The case where $\mathbf{A}$ does not satisfy τ is similar. It follows that τ is almost surely equivalent to ϕ^{n_0}, proving (ii).

Up to this point, we have dealt with sentences that contain a single occurrence of **while**. We now show that every sentence in iterative logic is equivalent almost surely to a first-order sentence, by induction on the structure of the sentence. If $\tau = (X)\psi_1$; **while** Y_1 **do** S_1; ψ_2; **while** Y_2 **do** S_2; ..., where $\psi_1, \psi_2, \ldots$ are in first-order logic, we can find a first-order sentence υ_1 that is equivalent almost surely to $(X)\psi_1$; **while** Y_1 **do** S_1, and we can then find a first-order sentence υ_2 that is equivalent almost surely to $(X)\upsilon_1; \psi_2$; **while** Y_2 **do** S_2, etc. It follows that τ is equivalent to $(X)\upsilon_1; \upsilon_2; \ldots$ which contains no **while** loops.

Let τ be a sentence in iterative logic containing a nesting of $n+1$ **while** loops. We can assume that $\tau = (X)\ \phi_1$; **while** Y **do** S; ϕ_2, where S is a nesting of n loops, and ϕ_1 is in first-order logic. (Otherwise, if ϕ_1 contains **while** looping, we can use the inductive hypothesis to get an almost surely equivalent first-order sentence.) Consider $\phi_1; \{S^i\}_{i>0}$. Each $\phi_1; S^i$ is a program containing up to n nestings of **while** loops. Hence, by the inductive hypothesis, it is almost surely equivalent to a first-order sentence, and we can apply the above proof technique to prove that the entire sentence τ is almost surely equivalent to a first-order sentence. $\triangle$

Proposition 2.11 [BGK] *Let ϕ be a formula in FO+IFP (which is actually FP_1) over vocabulary σ. There exists a first-order σ-formula ϕ' and a finite subset G of the extension axioms, such that ϕ and ϕ' are equivalent in all models of G.*

In fact, ϕ and ϕ' are almost surely equivalent in recursive structures, since almost all the structures in every T-sequence satisfy G.

We now extend this theorem from FP_1 to the entire fixpoint hierarchy FP. Let FP_k be the set of formulas in FP consisting of up to k compositions of fixpoint operators.

Proposition 2.12 *Let ϕ be a formula in FP over σ. There exists a first-order σ-formula ϕ' and a finite subset G of the extension axioms, such that ϕ and ϕ' are equivalent in all models of G.*

Proof: The proof is by induction on the structure of the formula. For FP_1 we have Prop. 2.11, and this clearly holds also for $\neg\mathrm{FP}_1$. Assume that the result holds for FP_k. Let ϕ be a formula in FP_{k+1}, of the form $\phi(B, \psi(B))$, where B is the input structure and ψ is in FP_k. By the inductive hypothesis, ψ is equivalent to a first-order formula ψ' in all models of some finite set of extension axioms G. Now replace each appearance of $\overline{x} \in \psi(B)$ in the formula ϕ by $\overline{x} \in \psi'(B)$. We obtain a formula ϕ' in FP_1. By Prop. 2.11, ϕ' is equivalent to a first-order formula ϕ'' in all models of a finite set G' of extension axioms. It follows that ϕ is equivalent to ϕ'' in all models of $G \cup G'$. $\triangle$

The 0–1 law for FP now follows from the 0–1 law for first-order logic:

Proposition 2.13 *The 0–1 law on recursive structures holds for FP over a vocabulary σ. Moreover, if $\mathbf{A}$ is the countable random structure, P is a property expressed in FP, and $\mathcal{F}$ is a T-sequence, then $\mathbf{A} \models P$ iff $\mu_{\mathcal{F}}(P) = 1$.*

It follows from the definitions, that all the logics for which the 0–1 law for recursive structures holds also satisfy the transfer property for recursive structures. Considering finite structures, there might be logics that satisfy the 0–1 law but do not satisfy the transfer property, but the logics we consider here all satisfy the 0–1 laws for both finite and recursive structures, as well as the transfer property for both.

Proposition 2.14 *Let ϕ be a sentence in first-order logic, strict E-Σ^1_1, strict E-Π^1_1, iterative logic, or FP. The following are equivalent:*

- $\mathbf{A}$ *satisfies* ϕ.
- $\mu_C(\phi) = 1$ *for the class C of finite structures.*
- $\mu_{\mathcal{F}}(\phi) = 1$ *for every T-sequence $\mathcal{F}$.*

Due to this proposition, we can determine the complexity of the decision problem for values of the probabilities of properties expressible in these logics from the results appearing in [KV1] for the finite case.

2.5 Non-expressibility by 0–1 laws

By Props. 2.9, 2.10 and 2.13, properties for which the 0–1 law does not hold are not expressible in first-order logic, strict E-Σ^1_1, strict E-Π^1_1, iterative logic or FP. Moreover, we can conclude from Lemma 2.7 non-expressibility of properties in E-Π^1_1 too:

Corollary 2.15 *Every property on recursive structures that is true on $\mathbf{A}$ but does not have probability 1 on some T-sequence, is not expressible in first-order logic, iterative logic, FP, strict E-Σ^1_1 or E-Π^1_1.*

Let bg-CLIQUE be the set of recursive graphs that have arbitrarily large cliques.

Example 2.2 ∞-CLIQUE, ∞-IND, bg-CLIQUE, *EA and ISO are not expressible in first-order logic, iterative logic, FP, strict E-Σ^1_1 or E-Π^1_1.*

1. ∞-CLIQUE and ∞-IND:

   $\mathbf{A} \models \infty$-CLIQUE and $\mathbf{A} \models \infty$-IND, since the random countable graph $\mathbf{A}$ is universal [F4], which means that every countable graph is embeddable in it.[3] In particular, infinite cliques and infinite independent sets are embeddable in $\mathbf{A}$. However, ∞-CLIQUE and ∞-IND do not have probability 1 on the T-sequences $\mathcal{F}$ and $\mathcal{F}'$, respectively, that were defined in Example 2.1.

2. bg-CLIQUE:

   $\mathbf{A} \models$ bg-CLIQUE, since $\mathbf{A}$ contains an infinite clique. But bg-CLIQUE has probability 0 on the T-sequence $\mathcal{F} = \{F_n\}_{n=1}^{\infty}$ that was defined in Example 2.1, since each F_n contains a maximal clique of size $n + 1$.

[3] A graph H is embeddable in graph G if H is isomorphic to a subgraph of G.

3. EA:

   $\mathbf{A} \models$ EA, but EA does not have probability 1 on the T-sequence $\mathcal{F} = \{F_n\}_{n=1}^{\infty}$, that was defined in Example 2.1. Otherwise, if $\mu_{\mathcal{F}}(\text{EA}) = 1$, there would be many isomorphic graphs in $\mathcal{F}$, since T, the theory of all extension axioms, is ω-categorical. But for all $i < j$, A_i is not isomorphic to A_j, since A_j contains a clique of size $j+1$, but A_i does not. In fact, for all i, A_i satisfies all the i-extension axioms but it does not satisfy some $(i+1)$-extension axiom.

4. ISO:

   Let $\sigma = \{P_1, P_2\}$ and let $\sigma' = \{E\}$, for binary P_1, P_2 and E. Let $\mathbf{A} = \{A', A''\}$ be a countable random structure over σ. In particular, A' and A'' are countable random graphs over σ'. Since all countable random graphs are isomorphic, $A' \cong A''$; i.e., $\mathbf{A} \models$ ISO.

   We now exhibit a T-sequence $\mathcal{F}$ over σ, for which $\mu_{\mathcal{F}}(\text{ISO}) = 0$. Let $\mathcal{F} = \{\langle A_n, A'_n\rangle\}_{n=1}^{\infty}$ be a sequence of recursive structures over σ, constructed as follows. Level 1 of $\langle A_n, A'_n\rangle$ contains n points. Assume that levels 1 to n of $\langle A_n, A'_n\rangle$ exist. For each set of n points $\overline{x} = (x_1, \ldots, x_n)$ from lower levels we add a new point z for each possible $(n+1)$-σ-type $s(\overline{x}, z)$ that extends the n-σ-type $t(\overline{x})$. We define the containment of pairs of the form (x_i, z) and (z, x_i), for $1 \leq i \leq n$, in A_n and in A'_n according to $s(\overline{x}, z)$. In addition, for each set of $n+1$ points $\overline{y} = (y_1, \ldots, y_n, y_{n+1})$ from lower levels, we add an additional new point v for each possible $(n+1)$-σ'-type $s(\overline{y}, v)$ extending the n-σ'-type $t'(\overline{y})$, and we define the containments of pairs over $\{\overline{y}, v\}$ in A' accordingly. Now, $\mathcal{F}$ must be a T-sequence, since each $\langle A_n, A'_n\rangle$ satisfies all the n-extension axioms over σ. In addition, when considering A_n and A'_n separately, A_n does not satisfy all the $(n+1)$-extension axioms over σ', whereas A'_n does. Hence, for each n, $A_n \ncong A'_n$.

$\triangle$

References

[AHV] S. Abiteboul, R. Hull and V. Vianu, *Foundations of Databases*, Addison-Wesley, Reading, MA, 1995.

[AV1] S. Abiteboul and V. Vianu, "Generic Computation and its Complexity", *Proc. 23rd ACM Symp. on Theory of Computing*, ACM Press, New York, 1991, pp. 209–219.

[AV2] S. Abiteboul and V. Vianu, "Computing with First-order Logic", *J. Comput. Syst. Sci.* **50** (1995), 309–335.

[AMS] R. Aharoni, M. Magidor and R. A. Shore, "On the Strength of König's Duality Theorem", *J. of Combinatorial Theory (Series B)* **54** (1992), 257–290.

[AF] M. Ajtai and R. Fagin. "Reachability is harder for directed than for undirected finite graphs", *J. of Symbolic Logic* **55** (1990), 113–150.

[B] F. Bancilhon, "On the completeness of query languages for relational data bases", in *Proc. 7th Symp. on Math. Found. Comput. Sci.*, Lecture Notes in Computer Science, Springer-Verlag, Berlin, 1978.

[BGK] A. Blass, Y. Gurevich and D. Kozen, "A Zero-One Law for Logic with a Fixed Point Operator", *Inf. and Cont.* **67** (1985), 70–90.

[CH1] A. K. Chandra and D. Harel, "Computable Queries for Relational Data Bases", *J. Comp. Syst. Sci.* **21** (1980), 156–178.

[CH2] A. K. Chandra and D. Harel, "Structure and Complexity of Relational Queries", *J. Comp. Syst. Sci.* **25** (1982),99–128.

[C] E.F. Codd, "Relational Completeness of Data Base Sublanguages", In *Data Base Systems* (Rustin, ed.), Prentice-Hall, Englewood Cliffs, N.J., 1972.

[Co] K. J. Compton, "0–1 laws in logic and combinatorics", in *NATO Adv. Study Inst. on Algorithms and Order* (I. Rival, ed.), D. Reidel, 1988, pp. 353–383.

[E] A. Ehrenfeucht, "An application of games to the completeness problem for formalized theories", *Fund. Math.* **49** (1961), 129–141.

[F1] R. Fagin, "Generalized First-Order Spectra and Polynomial-Time Recognizable Sets", In *Complexity of Computations* (R. Karp, ed.), SIAM-AMS Proceedings, Vol. 7, 1974, pp. 43–73.

[F2] R. Fagin, "Monadic Generalized Spectra", *Zeitschrift fur Math. Logic* **21** (1975), 89–96.

[F3] R. Fagin, "Probabilities on Finite Models", *J. of Symbolic Logic* **41** (1976), 50–58.

[F4] R. Fagin, "Finite-Model Theory — a Personal Perspective", *Theoret. Comput. Sci.* **116** (1993), 3–31.

[FSV] R. Fagin, L. Stockmeyer and M. Y. Vardi, "On Monadic NP vs. Monadic co-NP", *Inf. and Comp.* **120** (1995), 78–92.

[Fr] R. Fräisse, *Cours de Logique Mathematique*, Gauthier-Villars and E. Nauwelaerts, 1967.

[Ga] H. Gaifman, "Concerning Measures in First-Order Calculi", *Israel J. Math.* **2** (1964), 1–18.

[G1] Y. Gurevich, "Toward Logic Tailored for Computational Complexity", in *Computation and Proof Theory* (M. Richter et al., eds.), Springer-Verlag Lecture Notes in Mathematics, vol. 1104, New York, 1984, pp. 175–216.

[G2] Y. Gurevich, "On Finite Model Theory", in *Perspectives in Computer Science* (S. R. Buss and P. J. Scott, eds.), Birkhauser, 1990.

[GKLT] Y. V. Glebskii, D. I. Kogan, M. I. Liogonki and V. A. Talanov, "Range and Degree of Realizability of Formulas in the Restricted Predicate Calculus", *Cybernetics* **5** (1969), 142–154.

[GS] S. Grumbach and Jianwen Su, "Finitely Representable Databases", *Proc. 13th ACM Symp. on Principles of Database Systems*, ACM Press, New York, 1994, pp. 289–300.

[H] D. Harel, "Hamiltonian Paths in Infinite Graphs", *Israel J. Math.* **76** (1991), 317–336. (Preliminary version in *Proc. 23rd ACM Symp. on Theory of Computing,* ACM Press, New York, 1991, pp. 220–229.)

[Ha] W. Hanf, "Model-Theoretic Methods in the Study of Elementary Logic", in *the Theory of Models* (J. Addison, L. Henkin and A. Tarski, eds.), North-Holland, Amsterdam, 1965, pp. 132–145.

[HH1] T. Hirst and D. Harel, "Taking it to the Limit: On Infinite Variants of NP-Complete Problems", *J. Comput. Syst. Sci.*, to appear. (Also, *Proc. 8th IEEE Conf. on Structure in Complexity Theory*, IEEE Press, New York, 1993, pp. 292–304.)

[HH2] 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*, ACM Press, New York, 1993, pp. 244–252.)

[I] N. Immerman, “Relational Queries Computable in Polynomial Time”, *Inf. and Cont.* **68** (1986), 86–104.

[KV1] P. G. Kolaitis and M. Y. Vardi, “The Decision problem for the probabilities of higher-order properties”, *Proc. 19th ACM Symp. on Theory of Computing*, 1987, pp. 425–435.

[KV2] P. G. Kolaitis and M. Y. Vardi, “0–1 laws and decision problems for fragments of second-order logic” *Inf. and Comput.* **87** (1990), 302–339.

[KV3] P. G. Kolaitis and M. Y. Vardi, “Infinitary logics and 0–1 laws”, *Inf. and Comput.* **98** (1992), 258–294.

[Mo] A. S. Morozov, “Functional Trees and Automorphisms of Models”, *Algebra and Logic* **32** (1993), 28–38.

[Mos] Y. N. Moschovakis, *Elementary Induction on Abstract Structures*, North Holland, 1974.

[P] J. Paredaens, “On the Expressive Power of the Relational Algebra”, *Inf. Proc. Let.* **7** (1978), 107–111.

[Ra] R. Rado, “Universal Graphs and Universal Functions”, *Acta Arith.* **9** (1964), 331–340.

[R] H. Rogers, *Theory of Recursive Functions and Effective Computability*, McGraw-Hill, New York, 1967.

[Ry] C. Ryll Nardzeweski, “On the Categoricity in Power $\aleph_0$”, *Bull. Acad. Polon. Sci. Ser. Sci. Math. Astron. Phys.* **7** (1959), 239–263.

[S] A. Stolboushkin, “Towards Recursive Model Theory”, manuscript, May, 1996, based on a lecture given at the Logic Colloquium, 1995.

[TK] V.A. Talanov and V.V. Knyazer, “The asymptotic truth value of infinite formulas”, in *All-union Seminar on Discrete Mathematics and its Applications*, 1984, pp. 56–61.

[V1] M. Y. Vardi, “The Complexity of Relational Query Languages”, *Proc. 14th ACM Symp. on Theory of Computing*, 1982, pp. 137–146.

[V2] M. Y. Vardi, “On Decomposition of Relational Databases”, *Proc. 23rd IEEE Conf. on Found. Comput. Sci.*, (1982), 176–185.

[Va] R. Vaught, “Denumerable Models of Complete theories”, in *Infinitistic Methods*, Pergamon Press, 1961, pp. 303–321.

A Appendix: Ehrenfeucht-Fraisse Games

(Note: This appendix is mostly adapted from [FSV].)

An *r-round first-order Ehrenfeucht-Fraisse game* (or *r-game* for short), where r is a positive integer, is the following game. There are two players, called the *spoiler* and the *duplicator*, and two colored graphs, G_0 and G_1. In the first round, the spoiler selects a point in one of the two graphs, and the duplicator selects a point in the other one. Let a_1 be the point selected in G_0, and let b_1 be the point selected in G_1. The second round is similar, with the spoiler again selecting a point in one of the two graphs, and the duplicator selecting a point in the other one. Let a_2 be the point selected in G_0, and b_2 be the point selected in G_1. This continues for r rounds. The duplicator wins if the colored subgraph of G_0 induced by $\{a_1, \ldots, a_r\}$ is isomorphic to the colored subgraph of G_1 induced by $\{b_1, \ldots, b_r\}$, under the function that maps a_i to b_i for $1 \leq i \leq r$.

Ajtai and Fagin [AF] consider r-games over a class of graphs $\mathcal{S}$. The rules of this game are as follows: The duplicator begins by selecting a member of $\mathcal{S}$ to be G_0, and a member of $\overline{\mathcal{S}}$ to be G_1. The players then play an r-game on these two graphs, and the duplicator wins the general game if he wins the r-game.

Proposition A.1 [E, Fr, FSV] *$\mathcal{S}$ is first-order definable iff there is r such that the spoiler has a winning strategy in the r-game over $\mathcal{S}$.*

A more complicated game, which is a c-colored, r-round, monadic NP game, (or a *(c, r)-game* for short) was introduced in [F3] to prove that connectivity of finite graphs is not in monadic NP (which is really just monadic E-Σ^1_1). A (c, r)-game over a class $\mathcal{S}$ is defined as follows:

1. The duplicator selects a member of $\mathcal{S}$ to be G_0.
2. The duplicator selects a member of $\overline{\mathcal{S}}$ to be G_1.
3. The spoiler colors G_0 with the c colors.
4. The duplicator colors G_1 with the c colors.
5. The spoiler and the duplicator play an r-game.

The winner is decided as before.

Proposition A.2 [F2, FSV] *$\mathcal{S}$ is in monadic NP iff there are c and r such that the spoiler has a winning strategy in the (c, r)-game over $\mathcal{S}$.*

A variant of this game, in which the steps 2 and 3 above are reversed (so that the duplicator gets to see the spoiler's coloring of G_0 before selecting G_1), was defined by Ajtai and Fagin in [AF]. For this game we have:

Proposition A.3 [AF, FSV] *$\mathcal{S}$ is in monadic NP iff there are c and r such that the spoiler has a winning strategy in the Ajtai-Fagin (c, r)-game over $\mathcal{S}$.*

The following definitions are due to Hanf [Ha], and appear also in [FSV]. Let A be a structure and let a and b be two points in A. We say that a and b are *adjacent* (in A) if either $a = b$ or there is some relation R_i in A and some tuple t, such that $t \in R_i$ and such that a and b are entries in t. The *neighborhood of* *a of radius d*, *nbd*(d, a), consists of all points whose distance from a is *strictly* less than d. It is defined recursively as follows:

$$\begin{aligned} nbd(1, a) &= \{a\} \\ nbd(d+1, a) &= \{x \mid x \text{ is adjacent to some } b \in nbd(d, a)\} \end{aligned}$$

The *d-type* of a point a in a structure A is the isomorphism type of the neighborhood of radius d about a. Thus, two points a and b in A have the same d-type precisely if A restricted to $nbd(d, a)$ is isomorphic to A restricted to $nbd(d, b)$, under an isomorphism mapping a to b.

The structures A and B are (d, m)*-equivalent*, if for every d-type τ, either A and B have the same number of points with d-type τ, or else both have at least m points with d-type τ. They are *d-equivalent* if for every d-type τ, they have exactly the same number of points with d-type τ.

We write $A \sim_r B$ if the duplicator has a winning strategy in an r-game. In this case, A and B are indistinguishable by an r-game.

The following result is from [FSV]. The proof is based on a technique of Hanf [Ha]:

Proposition A.4 *Let r and f be positive integers. There are positive integers d and m, where d depends only on r, such that for any (d, m)-equivalent structures A and B, in which every point has degree at most f, we have $A \sim_r B$.*

Since d in Prop. A.4 depends only on r, and since two d-equivalent structures are (d, m)-equivalent for any m, the following is immediate:

Corollary A.5 *Let r be positive integer. There is a positive integer d, such that for any d-equivalent structures A and B, we have $A \sim_r B$.*