Logical Methods in Computer Science
Vol. 7 (3:14) 2011, pp. 1–24
www.lmcs-online.org

Submitted Nov. 17, 2010
Published Sep. 21, 2011

# RANDOMISATION AND DERANDOMISATION IN DESCRIPTIVE COMPLEXITY THEORY

KORD EICKMEYER AND MARTIN GROHE

Humboldt-Universität zu Berlin, Institut für Informatik, Logik in der Informatik
Unter den Linden 6, 10099 Berlin, Germany
*e-mail address*: eickmeye@informatik.hu-berlin.de, grohe@informatik.hu-berlin.de

Abstract. We study probabilistic complexity classes and questions of derandomisation from a logical point of view. For each logic $\mathsf{L}$ we introduce a new logic $\mathsf{BPL}$, *bounded error probabilistic* $\mathsf{L}$, which is defined from $\mathsf{L}$ in a similar way as the complexity class $\mathsf{BPP}$, bounded error probabilistic polynomial time, is defined from $\mathsf{P}$.

Our main focus lies on questions of derandomisation, and we prove that there is a query which is definable in $\mathsf{BPFO}$, the probabilistic version of first-order logic, but not in $\mathsf{C}^{\omega}_{\infty\omega}$, finite variable infinitary logic with counting. This implies that many of the standard logics of finite model theory, like transitive closure logic and fixed-point logic, both with and without counting, cannot be derandomised. Similarly, we present a query on ordered structures which is definable in $\mathsf{BPFO}$ but not in monadic second-order logic, and a query on additive structures which is definable in $\mathsf{BPFO}$ but not in $\mathsf{FO}$. The latter of these queries shows that certain uniform variants of $\mathsf{AC}^0$ (bounded-depth polynomial sized circuits) cannot be derandomised. These results are in contrast to the general belief that most standard complexity classes can be derandomised.

Finally, we note that $\mathsf{BPIFP{+}C}$, the probabilistic version of fixed-point logic with counting, captures the complexity class $\mathsf{BPP}$, even on unordered structures.

## 1. Introduction

The relation between different modes of computation — deterministic, nondeterministic, randomised — is a central topic of computational complexity theory. The $\mathsf{P}$ vs. $\mathsf{NP}$ problem falls under this topic, and so does a second very important problem, the relation between randomised and deterministic polynomial time. In technical terms, this is the question of whether $\mathsf{P} = \mathsf{BPP}$, where $\mathsf{BPP}$ is the class of all problems that can be solved by a randomised polynomial time algorithm with two-sided errors and bounded error probability. This question differs from the question of whether $\mathsf{P} = \mathsf{NP}$ in that most complexity theorists seem to believe that the classes $\mathsf{P}$ and $\mathsf{BPP}$ are indeed equal. This belief is supported by deep results due to Nisan and Wigderson [31] and Impagliazzo and Wigderson [20], which link the derandomisation question to the existence of one-way functions and to circuit lower

*1998 ACM Subject Classification:* F.4.1 [Mathematical Logic]: Finite Model Theory, F.1.2 [Modes of Computation]: Probabilistic Computation.

*Key words and phrases:* Descriptive Complexity, Probabilistic Complexity Classes, Derandomisation.


DOI:10.2168/LMCS-7 (3:14) 2011

bounds; cf. also [21]. Similar derandomisation questions are studied for other complexity classes such as logarithmic space, and it is believed that derandomisation is possible for these classes as well.

Descriptive complexity theory gives logical descriptions of complexity classes and thus enables us to translate complexity theoretic questions into the realm of logic. While logical descriptions are known for most natural deterministic and nondeterministic time and space complexity classes, probabilistic classes such as $\mathsf{BPP}$ have received very little attention in descriptive complexity theory yet. In this paper, we study probabilistic complexity classes and questions of derandomisation from a logical point of view. For each logic $\mathsf{L}$ we introduce a new logic $\mathsf{BPL}$, *bounded error probabilistic* $\mathsf{L}$, which is defined from $\mathsf{L}$ in a similar way as $\mathsf{BPP}$ is defined from $\mathsf{P}$. The randomness is introduced to the logic by letting formulas of vocabulary $\tau$ speak about *random expansions* of $\tau$-structures to a richer vocabulary $\tau \cup \rho$. We also introduce variants $\mathsf{RL}$, $\mathsf{co\text{-}RL}$ with one-sided bounded error and $\mathsf{PL}$ with unbounded error, corresponding to other well known complexity classes.

Our main technical results are concerned with questions of derandomisation. By this we mean upper bounds on the expressive power of randomised logics in terms of classical logics. Trivially, $\mathsf{BPL}$ is at least as expressive as $\mathsf{L}$, and if the two logics are equally expressive, then we say that $\mathsf{BPL}$ *derandomisable.* More generally, if $\mathsf{L}'$ is a (deterministic) logic that is at least as expressive as $\mathsf{BPL}$, then we say that $\mathsf{BPL}$ derandomisable *within* $\mathsf{L}'$. We prove that $\mathsf{BPFO}$, bounded error probabilistic first-order logic, is not derandomisable within $\mathsf{C}^{\omega}_{\infty\omega}$, finite variable infinitary logic with counting. This implies that many of the standard logics of finite model theory, like transitive closure logic and fixed-point logic, both with and without counting, cannot be derandomised. Note that these results are in contrast to the general belief that most standard complexity classes can be derandomised.

We then investigate whether $\mathsf{BPFO}$ can be derandomised on classes of structures with built-in relations, such as ordered structures and arithmetic structures. We prove that $\mathsf{BPFO}$ cannot be derandomised within $\mathsf{MSO}$, monadic second-order logic, on structures with built-in order. Furthermore, $\mathsf{BPFO}$ cannot be derandomised on structures with built-in order and addition. Interestingly and nontrivially, $\mathsf{BPFO}$ can be derandomised within $\mathsf{MSO}$ on structures with built-in order and addition. Behle and Lange [5] showed that the expressive power of $\mathsf{FO}$ on classes of ordered structures with certain predefined relation symbols corresponds to uniform subclasses of $\mathsf{AC}^0$, the class of problems decidable by circuit families of bounded depth, unbounded fan-in and polynomial size. In fact, for any set $\mathcal{R}$ of built-in relations they show that $\mathsf{FO}[\mathcal{R}]$ captures $\mathsf{FO}[\mathcal{R}]$-uniform $\mathsf{AC}^0$. Arguably the most intensively studied uniformity condition on $\mathsf{AC}^0$ is *dlogtime-uniform* $\mathsf{AC}^0$, which corresponds to $\mathsf{FO}[+, \times]$, first-order logic with built-in arithmetic (Barrington et al. [3]). The question of whether dlogtime-uniform $\mathsf{BPAC}^0$ can be derandomised is still open, but there is a conditional derandomisation by Viola [39]. There are less uniform variants of $\mathsf{BPAC}^0$ that can be proved to be derandomisable by standard arguments; cf. [1]. We prove that the more uniform $\mathsf{FO}[+]$-uniform $\mathsf{AC}^0$ is not derandomisable. This raises the question of how weak uniformity must be for derandomisation to be possible.

In the last section of this paper, we turn to more standard questions of descriptive complexity theory. We prove that $\mathsf{BPIFP{+}C}$, the probabilistic version of fixed-point logic with counting, captures the complexity class $\mathsf{BPP}$, even on unordered structures. For ordered structures, this result is a direct consequence of the Immerman-Vardi Theorem [18, 38], and for arbitrary structures it follows from the observation that we can define a random order with high probability in $\mathsf{BPIFP{+}C}$. Still, the result is surprising at first sight because of its similarity with the open question of whether there is a logic capturing P, and because it is believed that P = BPP. The caveat is that the logic BPIFP+C does not have an effective syntax and thus is not a "logic" according to Gurevich's [16] definition underlying the question for a logic that captures P. Nevertheless, we believe that BPIFP+C gives a completely adequate description of the complexity class BPP, because the definition of BPP is inherently ineffective as well (as opposed to the definition of P in terms of the decidable set of polynomially clocked Turing machines). We obtain similar descriptions of other probabilistic complexity classes. For example, randomised logspace is captured by the randomised version of deterministic transitive closure logic with counting.

### Related work

As mentioned earlier, probabilistic complexity classes such as BPP have received very little attention in descriptive complexity theory. There is an unpublished paper due to Kaye [22] that gives a logical characterisation of BPP on ordered structures. Müller [30] and Montoya (unpublished) study a logical BP-operator in the context of parameterised complexity theory. What comes closest to our work "in spirit" and also in some technical aspects is Hella, Kolaitis, and Luosto's work on *almost everywhere equivalence* [17], which may be viewed as a logical account of average case complexity in a similar sense that our work gives a logical account of randomised complexity. There is a another logical approach to computational complexity, known as implicit computational complexity, which is quite different from descriptive complexity theory. Mitchell, Mitchell, and Scedrov [28] give a logical characterisation of BPP by a higher-order typed programming language in this context.

Let us emphasise that the main purpose of this paper is not the definition of new probabilistic logics, but an investigation of these logics in a complexity theoretic context.

## 2. Preliminaries

### 2.1. Structures and Queries

A *vocabulary* is a finite set $\tau$ of relation symbols of fixed arities. A $\tau$-*structure* $A$ consists of a finite set $V(A)$, the *universe* of the structure, and, for all $R \in \tau$, a relation $R(A)$ on $A$ whose arity matches that of $R$. Thus we only consider *finite* and *relational* structures. Let $\sigma, \tau$ be vocabularies with $\sigma \subseteq \tau$. Then the $\sigma$-*restriction* of a $\tau$-structure $B$ is the $\sigma$-structure $B|_\sigma$ with universe $V(B|_\sigma) := V(B)$ and relations $R(B|_\sigma) := R(B)$ for all $R \in \sigma$. A $\tau$-*expansion* of a $\sigma$-structure $A$ is a $\tau$-structure $B$ such that $B|_\sigma = A$. For every class $\mathcal{C}$ of structures, $\mathcal{C}[\tau]$ denotes the class of all $\tau$-structures in $\mathcal{C}$. A *renaming* of a vocabulary $\tau$ is a bijective mapping $r$ from $\tau$ to a vocabulary $\tau'$ such that for all $R \in \tau$ the relation symbol $r(R) \in \tau'$ has the same arity as $R$. If $r : \tau \to \tau'$ is a renaming and $A$ is a $\tau$-structure then $A^r$ is the $\tau'$-structure with $V(A^r) := V(A)$ and $r(R)(A^r) := R(A)$ for all $R \in \tau$.

We let $\leqslant$, $+$ and $\times$ be distinguished relation symbols of arity two, three and three, respectively. Whenever any of these relations symbols appear in a vocabulary $\tau$, we demand that they be interpreted by a linear order and ternary addition and multiplication relations, respectively, in all $\tau$-structures. To be precise, let $[a, b]$ be the set $\{a, a+1, \ldots, b\}$ for

$a \leq b \in \mathbb{N}$, and denote by $\mathcal{N}_n$ the $\{\leqslant, +, \times\}$-structure with

$$V(\mathcal{N}_n) = [0, n-1], \qquad \leqslant(\mathcal{N}_n) = \{(a,b) \mid a \leqslant b\} \text{ and}$$
$$+(\mathcal{N}_n) = \{(a,b,c) \mid a+b=c\}, \qquad \times(\mathcal{N}_n) = \{(a,b,c) \mid a \cdot b = c\}.$$

We demand $A|_{\{\leqslant,+,\times\}\cap\tau} \cong (\mathcal{N}_{|A|})|_{\{\leqslant,+,\times\}\cap\tau}$ for all $\tau$-structures $A$. We call structures whose vocabulary contains any of these relation symbols *ordered*, *additive* and *multiplicative*, respectively. We say that a formula $\varphi(x)$ with exactly one free variable $x$ *defines an element* if in every structure it is satisfied by exactly one element. Since we may identify the elements of an ordered structure uniquely with natural numbers it makes sense to say, e.g., that "$\varphi(x)$ defines a prime number" or "$\varphi(x)$ defines a number $\leq \log^{O(1)} |A|$", and we will sometimes do so.

On ordered structures, every fixed natural number $i$ can be defined in first-order logic by a formula $\varphi_{i\text{-th}}$ using only three variables as follows:

$$\begin{aligned}\varphi_{0\text{-th}}(x) &:= \forall y\, x \leq y \\ \varphi_{(n+1)\text{-th}}(x) &:= \exists y \forall z \left(\varphi_{n\text{-th}}(y) \wedge \neg(x \doteq y) \wedge y \leq x \wedge \right. \\ &\qquad \left. ((y \leq z \wedge z \leq x) \to (y \doteq z \vee y \doteq z))\right).\end{aligned}$$

Because the ordering may be defined using the addition relation, the same holds true on additive structures, again using only three variables.

A *$k$-ary $\tau$-global relation* is a mapping $\mathcal{R}$ that associates a $k$-ary relation $\mathcal{R}(A)$ with each $\tau$-structure $A$. A 0-ary $\tau$-global relation is usually called a *Boolean* $\tau$-global relation. We identify the two 0-ary relations $\emptyset$ and $\{()\}$, where $()$ denotes the empty tuple, with the truth values $\mathsf{false}$ and $\mathsf{true}$, respectively, and we identify the Boolean $\tau$-global relation $\mathcal{R}$ with the class of all $\tau$-structures $A$ with $\mathcal{R}(A) = \mathsf{true}$. A *$k$-ary $\tau$-query* is a $k$-ary $\tau$-global relation $\mathcal{Q}$ preserved under isomorphism, that is, if $f$ is an isomorphism from a $\tau$-structure $A$ to a $\tau$-structure $B$ then for all $\vec{a} \in V(A)^k$ it holds that $\vec{a} \in \mathcal{Q}(A) \iff f(\vec{a}) \in \mathcal{Q}(B)$.

## 2.2. **Logics**

A *logic* $\mathsf{L}$ has a *syntax* that assigns a set $\mathsf{L}[\tau]$ of $\mathsf{L}$*-formulas of vocabulary* $\tau$ with each vocabulary $\tau$ and a *semantics* that associates a $\tau$-global relation $\mathcal{Q}_\varphi^{\mathsf{L}[\tau]}$ with every formula $\varphi \in \mathsf{L}[\tau]$ such that for all vocabularies $\sigma, \tau, \tau'$ the following three conditions are satisfied:

(1) For all $\varphi \in \mathsf{L}[\tau]$ the global relation $\mathcal{Q}_\varphi^{\mathsf{L}[\tau]}$ is a $\tau$-query.
(2) If $\sigma \subseteq \tau$ then $\mathsf{L}[\sigma] \subseteq \mathsf{L}[\tau]$, and for all formulas $\varphi \in \mathsf{L}[\sigma]$ and all $\tau$-structures $A$ it holds that $\mathcal{Q}_\varphi^{\mathsf{L}[\sigma]}(A|_\sigma) = \mathcal{Q}_\varphi^{\mathsf{L}[\tau]}(A)$.
(3) If $r : \tau \to \tau'$ is a renaming, then for every formula $\varphi \in \mathsf{L}[\tau]$ there is a formula $\varphi^r \in \mathsf{L}[\tau']$ such that for all $\tau$-structures $A$ it holds that $\mathcal{Q}_\varphi^{\mathsf{L}[\tau]}(A) = \mathcal{Q}_{\varphi^r}^{\mathsf{L}[\tau']}(A^r)$.

Condition (ii) justifies dropping the vocabulary $\tau$ in the notation for the queries and just write $\mathcal{Q}_\varphi^{\mathsf{L}}$. For a $\tau$-structure $A$ and a tuple $\vec{a}$ whose length matches the arity of $\mathcal{Q}_\varphi^{\mathsf{L}}$, we usually write $A \models_{\mathsf{L}} \varphi[\vec{a}]$ instead of $\vec{a} \in \mathcal{Q}_\varphi^{\mathsf{L}}(A)$. If $\mathcal{Q}_\varphi^{\mathsf{L}}$ is a $k$-ary query, then we call $\varphi$ a *$k$-ary* formula, and if $\mathcal{Q}_\varphi^{\mathsf{L}}$ is Boolean, then we call $\varphi$ a *sentence*. Instead of $A \models_{\mathsf{L}} \varphi[()]$ we just write $A \models_{\mathsf{L}} \varphi$ and say that $A$ *satisfies* $\varphi$. We omit the index $\mathsf{L}$ if $\mathsf{L}$ is clear from the context.

A query $\mathcal{Q}$ is *definable* in a logic $\mathsf{L}$ if there is an $\mathsf{L}$-formula $\varphi$ such that $\mathcal{Q} = \mathcal{Q}_\varphi^{\mathsf{L}}$. Two formulas $\varphi_1, \varphi_2 \in \mathsf{L}[\tau]$ are *equivalent* (we write $\varphi_1 \equiv \varphi_2$) if they define the same query. We say that a logic $\mathsf{L}_1$ is *weaker* than a logic $\mathsf{L}_2$ (we write $\mathsf{L}_1 \leqq \mathsf{L}_2$) if every query definable in $\mathsf{L}_1$ is also definable in $\mathsf{L}_2$. Similarly, we define it for $\mathsf{L}_1$ and $\mathsf{L}_2$ to be *equivalent* (we write $\mathsf{L}_1 \equiv \mathsf{L}_2$) and for $\mathsf{L}_1$ to be *strictly weaker* than $\mathsf{L}_2$ (we write $\mathsf{L}_1 \lneqq \mathsf{L}_2$). The logics $\mathsf{L}_1$ and $\mathsf{L}_2$ are *incomparable* if neither $\mathsf{L}_1 \leqq \mathsf{L}_2$ nor $\mathsf{L}_2 \leqq \mathsf{L}_1$.

**Remark 2.1.** Our notion of logic is very minimalistic, usually logics are required to meet additional conditions (see [8] for a thorough discussion). In particular, we do not require the syntax of a logic to be effective. Indeed, the main logics studied in this paper have an undecidable syntax. Our definition is in the tradition of abstract model theory (cf. [4]); proof theorists tend to have a different view on what constitutes a logic.

We assume that the reader has heard of the standard logics studied in finite model theory, specifically *first-order logic* $\mathsf{FO}$, *second-order logic* $\mathsf{SO}$ and its fragments $\Sigma^1_k$, *monadic second-order logic* $\mathsf{MSO}$, *transitive closure logic* $\mathsf{TC}$ and its *deterministic* variant $\mathsf{DTC}$, *least*, *inflationary*, and *partial fixed-point logic* $\mathsf{LFP}$, $\mathsf{IFP}$, and $\mathsf{PFP}$, and *finite variable infinitary logic* $\mathsf{L}^{\omega}_{\infty\omega}$. For all these logics except $\mathsf{LFP}$ there are also *counting versions*, which we denote by $\mathsf{FO{+}C}$, $\mathsf{TC{+}C}$, …, $\mathsf{PFP{+}C}$ and $\mathsf{C}^{\omega}_{\infty\omega}$, respectively. Only familiarity with first-order logic is required to follow most of the technical arguments in this paper. The other logics are more or less treated as “black boxes”. We will say a bit more about some of them when they occur later. The following diagram shows how the logics compare in expressive power:

$$\begin{array}{ccccccccccccc}
\mathsf{FO} & \lneqq & \mathsf{DTC} & \lneqq & \mathsf{TC} & \lneqq & \mathsf{LFP} \equiv \mathsf{IFP} & \lneqq & \mathsf{PFP} & \lneqq & \mathsf{L}^{\omega}_{\infty\omega} \\
\lneqq & & \lneqq & & \lneqq & & \lneqq & & \lneqq & & \lneqq \\
\mathsf{FO{+}C} & \lneqq & \mathsf{DTC{+}C} & \lneqq & \mathsf{TC{+}C} & \lneqq & \mathsf{IFP{+}C} & \lneqq & \mathsf{PFP{+}C} & \lneqq & \mathsf{C}^{\omega}_{\infty\omega}.
\end{array} \tag{2.1}$$

Furthermore, $\mathsf{MSO}$ is strictly stronger than $\mathsf{FO}$ and incomparable with all other logics displayed in (2.1).

## 2.3. Complexity theory

We assume that the reader is familiar with the basics of computational complexity theory and in particular the standard complexity classes such as $\mathsf{P}$ and $\mathsf{NP}$. Let us briefly review the class $\mathsf{BPP}$, *bounded error probabilistic polynomial time*, and other probabilistic complexity classes: A language $L \subseteq \Sigma^*$ is in $\mathsf{BPP}$ if there is a polynomial time algorithm $M$, expecting as input a string $x \in \Sigma^*$ and a string $r \in \{0,1\}^*$ of “random bits”, and a polynomial $p$ such that for every $x \in \Sigma^*$ the following two conditions are satisfied:

(i) If $x \in L$, then $\Pr_{r\in\{0,1\}^{p(|x|)}}\big(M \text{ accepts } (x,r)\big) \geq \frac{2}{3}$.
(ii) If $x \notin L$, then $\Pr_{r\in\{0,1\}^{p(|x|)}}\big(M \text{ accepts } (x,r)\big) \leq \frac{1}{3}$.

In both conditions, the probabilities range over strings $r \in \{0,1\}^{p(|x|)}$ chosen uniformly at random. The choice of the error bounds $1/3$ and $2/3$ in (i) and (ii) is somewhat arbitrary, they can be replaced by any constants $\alpha, \beta$ with $0 < \alpha < \beta < 1$ without changing the complexity class. (To reduce the error probability of an algorithm we simply repeat it several times with independently chosen random bits $r$.)

Hence $\mathsf{BPP}$ is the class of all problems that can be solved by a randomised polynomial time algorithm with bounded error probabilities. $\mathsf{RP}$ is the class of all problems that can be solved by a randomised polynomial time algorithm with bounded one-sided error on the positive side (the bound $1/3$ in (ii) is replaced by 0), and $\mathsf{co\text{-}RP}$ is the class of all problems that can be solved by a randomised polynomial time algorithm with bounded one-sided error on the negative side (the bound $2/3$ in (i) is replaced by 1). Finally, $\mathsf{PP}$

is the class we obtain if we replace the lower bound $\geq 2/3$ in (i) by $> 1/2$ and the upper bound $\leq 1/3$ in (ii) by $\leq 1/2$. Note that $\mathsf{PP}$ is not a realistic model of "efficient randomised computation", because there is no easy way of deciding whether an algorithm accepts or rejects its input. Indeed, by Toda's Theorem [37], the class $\mathsf{P}^{\mathsf{PP}}$ contains the full polynomial hierarchy. By the Sipser-Gács Theorem (see [24]), $\mathsf{BPP}$ is contained in the second level of the polynomial hierarchy. More precisely, $\mathsf{BPP} \subseteq \Sigma_2^p \cap \Pi_2^p$. It is an open question whether $\mathsf{BPP} \subseteq \mathsf{NP}$. However, as pointed out in the introduction, there are good reasons to believe that $\mathsf{BPP} = \mathsf{P}$.

### 2.4. Descriptive complexity

It is common in descriptive complexity theory to view complexity classes as classes of Boolean queries, rather than classes of formal languages. This allows it to compare logics with complexity classes. The translation between queries and languages is carried out as follows: Let $\tau$ be a vocabulary, and assume that $\leqslant \notin \tau$. With each ordered $(\tau \cup \{\leqslant\})$-structure $B$ we can associate a binary string $s(B) \in \{0,1\}^*$ in a canonical way. Then with each class $\mathcal{C} \subseteq \mathcal{O}[\tau \cup \{\leqslant\}]$ of ordered $\tau$ structures we associate the language $L(\mathcal{C}) := \{s(B) \mid B \in \mathcal{C}\} \subseteq \{0,1\}^*$. For a Boolean $\tau$-query $\mathcal{Q}$, let $\mathcal{Q}_{\leqslant} := \big\{B \in \mathcal{O}[\tau \cup \leqslant] \bigm| B|_\tau \in \mathcal{Q}\big\}$ be the class of all ordered $(\tau \cup \{\leqslant\})$-expansions of structures in $\mathcal{Q}$. We say that $\mathcal{Q}$ is *decidable* in a complexity class $\mathsf{K}$ if the language $L(\mathcal{Q}_{\leqslant})$ is contained in $\mathsf{K}$. We say that a logic $\mathsf{L}$ *captures* $\mathsf{K}$ if for all Boolean queries $\mathcal{Q}$ it holds that $\mathcal{Q}$ is definable in $\mathsf{L}$ if and only if $\mathcal{Q}$ is decidable in $\mathsf{K}$. We say that $\mathsf{L}$ is *contained* in $\mathsf{K}$ if all Boolean queries definable in $\mathsf{L}$ are decidable in $\mathsf{K}$.

**Remark 2.2.** Just like our notion of "logic", our notion of a logic "capturing" a complexity class is very minimalistic, but completely sufficient for our purposes. For a deeper discussion of logics capturing complexity classes we refer the reader to one of the textbooks [9, 15, 19, 25].

## 3. Randomised logics

Throughout this section, let $\tau$ and $\rho$ be disjoint vocabularies. Relations over $\rho$ will be "random", and we will reserve the letter $R$ for relation symbols from $\rho$. We are interested in *random* $(\tau \cup \rho)$*-expansions* of $\tau$-structures. For a $\tau$-structure $A$, by $\mathcal{X}(A, \rho)$ we denote the class of all $(\tau \cup \rho)$-expansions of $A$. We view $\mathcal{X}(A, \rho)$ as a probability space with the uniform distribution. Note that we can "construct" a random $X \in \mathcal{X}(A, \rho)$ by deciding independently for all $k$-ary $R \in \rho$ and all tuples $\vec{a} \in V(A)^k$ with probability $1/2$ whether $\vec{a} \in R(X)$. Hence if $\rho = \{R_1, \ldots, R_k\}$, where $R_i$ is $r_i$-ary, then a random $X \in \mathcal{X}(A, \rho)$ can be described by random bitstring of length $\sum_{i=1}^{k} n^{r_i}$, where $n := |V(A)|$. We are mainly interested in the probabilities

$$\Pr_{X \in \mathcal{X}(A,\rho)}(X \models \varphi)$$

that a random $(\tau \cup \rho)$-expansion of a $\tau$-structure $A$ satisfies a sentence $\varphi$ of vocabulary $\tau \cup \rho$ of some logic.

**Definition 3.1.** Let $\mathsf{L}$ be a logic and $0 \leq \alpha \leq \beta \leq 1$.

(1) A formula $\varphi \in \mathsf{L}[\tau \cup \rho]$ that defines a $k$-ary query has an $(\alpha, \beta]$-*gap* if for all $\tau$-structures $A$ and all $\vec{a} \in V(A)^k$ it holds that

$$\Pr_{X \in \mathcal{X}(A,\rho)}(X \models \varphi[\vec{a}]) \leq \alpha \qquad \text{or} \qquad \Pr_{X \in \mathcal{X}(A,\rho)}(X \models \varphi[\vec{a}]) > \beta.$$

(2) The logic $\mathsf{P}_{(\alpha,\beta]}\mathsf{L}$ is defined as follows: For each vocabulary $\tau$,

$$\mathsf{P}_{(\alpha,\beta]}\mathsf{L}[\tau] := \bigcup_{\rho} \big\{\varphi \in \mathsf{L}[\tau \cup \rho] \bigm| \varphi \text{ has an } (\alpha,\beta]\text{-gap}\big\},$$

where the union ranges over all vocabularies $\rho$ disjoint from $\tau$. To define the semantics, let $\varphi \in \mathsf{P}_{(\alpha,\beta]}\mathsf{L}[\tau]$. Let $k, \rho$ such that $\varphi \in \mathsf{L}[\tau \cup \rho]$ and $\varphi$ is $k$-ary. Then for all $\tau$-structures $A$,

$$\mathcal{Q}_{\varphi}^{\mathsf{P}_{(\alpha,\beta]}\mathsf{L}}(A) := \big\{\vec{a} \in V(A)^k \bigm| \Pr_{X \in \mathcal{X}(A,\rho)}(X \models_{\mathsf{L}} \varphi[\vec{a}]) > \beta\big\}.$$

It is easy to see that for every logic $\mathsf{L}$ and all $\alpha, \beta$ with $0 \leq \alpha \leq \beta \leq 1$ the logic $\mathsf{P}_{(\alpha,\beta]}\mathsf{L}$ satisfies conditions (i)–(iii) from Subsection 2.2 and hence is indeed a well-defined logic. We let

$$\mathsf{PL} := \mathsf{P}_{(1/2,1/2]}\mathsf{L} \quad \text{and} \quad \mathsf{RL} := \mathsf{P}_{(0,2/3]}\mathsf{L} \quad \text{and} \quad \mathsf{BPL} := \mathsf{P}_{(1/3,2/3]}\mathsf{L}.$$

We can also define a logic $\mathsf{P}_{[\alpha,\beta)}\mathsf{L}$ and let $\mathsf{co\text{-}RL} := \mathsf{P}_{[1/3,1)}\mathsf{L}$. The following lemma, which is an adaptation of classical probability amplification techniques to randomised logics, shows that for reasonable $\mathsf{L}$ the strength of the logic $\mathsf{P}_{(\alpha,\beta]}\mathsf{L}$ does not depend on the exact choice of the parameters $\alpha, \beta$. This justifies the arbitrary choice of the constants $1/3, 2/3$ in the definitions of $\mathsf{RL}$ and $\mathsf{BPL}$.

**Lemma 3.2.** *Let $\mathsf{L}$ be a logic that is closed under conjunctions and disjunctions. Then for all $\alpha, \beta$ with $0 < \alpha < \beta < 1$ it holds that $\mathsf{P}_{(0,\beta]}\mathsf{L} \equiv \mathsf{RL}$ and $\mathsf{P}_{(\alpha,\beta]}\mathsf{L} \equiv \mathsf{BPL}$.*

*Proof.* Let $\tau$ an $\rho = \{R_1, \ldots, R_k\}$ be disjoint relational vocabularies and let $\varphi \in \mathsf{L}[\tau \cup \rho]$. For any $n \geq 1$ we define a new vocabulary

$$\rho^{(n)} := \{R_j^{(i)} \mid 1 \leq i \leq n, 1 \leq j \leq k\},$$

where the arity of $R_j^{(i)}$ is that of $R_j \in \rho$. Using the renaming property with the renaming

$$r^{(i)} : (\tau \cup \rho) \to (\tau \cup \rho^{(n)})$$

that leaves $\tau$ fixed and maps $R_j \in \rho$ to $R_j^{(i)}$ we get sentences $\varphi^{(i)}$, which are the sentence $\varphi$ with every occurrence of $R_j$ replaced by $R_j^{(i)}$. Since $\mathsf{L}$ is closed under conjunctions and disjunctions, for every $0 < l \leq n$ there is an $\mathsf{L}[\tau \cup \rho^{(n)}]$-sentence

$$\varphi^{(n,l)} := \bigvee_{\substack{I \subseteq [n] \\ |I| = l}} \bigwedge_{i \in I} \varphi^{(i)}$$

which is satisfied iff at least $l$ of the $\varphi^{(i)}$ are satisfied. Notice that the $\varphi^{(i)}$ use distinct random relations, so they are satisfied independently of each other.

Clearly, if $\Pr(X \models \varphi) = 0$ then also $\Pr(X \models \varphi^{(n,l)}) = 0$, because we assumed $l \geq 1$. On the other hand, if $\Pr(X \models \varphi) > \beta$ for some $\beta \in (0,1)$, then

$$\begin{aligned} \Pr(X \models \varphi^{(n,1)}) &= 1 - (1 - \Pr(X \models \varphi))^n && (3.1)\\ &> 1 - (1-\beta)^n, && (3.2) \end{aligned}$$

and this bound can be made arbitrarily close to 1 by choosing $n$ sufficiently large. This proves the claim about $\mathsf{RL}$.

For $\mathsf{BPL}$, notice that if $\varphi$ has an $(\alpha, \beta]$-gap for some any $0 < \alpha < \beta < 1$, then for any $0 < \alpha' < \beta' < 1$ there is an $n \in \mathbb{N}$ such that

$$\varphi^{(n, \lceil \frac{\beta - \alpha}{2} \rceil)}$$

has an $(\alpha', \beta']$-gap. In fact, the Chernoff bound (see, e.g., [29]) gives very sharp estimates on $n$ in terms of $\alpha$, $\beta$, $\alpha'$ and $\beta'$, though we only need the mere existence of such an $n$ here. $\square$

### 3.1. **First observations**

We start by observing that the syntax of $\mathsf{BPFO}$ and thus of most other logics $\mathsf{BPL}$ is undecidable. This follows easily from Trakhtenbrot's Theorem (see [9] for similar undecidability proofs):

**Observation 3.3.** For all $\alpha, \beta$ with $0 \leq \alpha < \beta < 1$ and all vocabularies $\tau$ containing at least one at least binary relation symbol, the set $\mathsf{BP}_{(\alpha,\beta]}\mathsf{FO}[\tau]$ is undecidable.

*Proof Sketch.* Assume for some $0 \leq \alpha < \beta < 1$ and some $\tau$ containing a binary relation symbol $E$ the set $\mathsf{BP}_{(\alpha,\beta]}\mathsf{FO}[\tau]$ is decidable.

By Trakhtenbrot's Theorem (cf. [9, Thm. 7.2.1]), the satisfiability of a first-order formula $\psi \in \mathsf{FO}[\tau]$ on finite graphs is undecidable. Let $\mathcal{G}$ be the class of all graphs with exactly one isolated vertex, and let $\varphi_{\mathcal{G}}$ be a sentence defining $\mathcal{G}$ on finite structures. By standard arguments, whether a formula is satisfiable in $\mathcal{G}$ or on is undecidable.

Let $p = a \cdot 2^{-k} \in (\alpha, \beta)$ with $a \in \mathbb{N}$ be a dyadic rational in the interval $(\alpha, \beta)$, and let $R_1, \ldots, R_k$ be unary random relations. For every $S \subset [k]$, the sentence

$$\psi_S := \exists x \left( (\forall y \, \neg Exy) \wedge \bigwedge_{i \in S} R_i x \wedge \bigwedge_{i \notin S} \neg R_i x \right)$$

has satisfaction probability $2^{-k}$ in all structures in $\mathcal{G}$. Thus for a family $\mathcal{S} = \{S_1, \ldots, S_a\}$ of $a$ distinct subsets of $[k]$, the sentence

$$\psi_{\mathcal{S}} := \bigvee_{S \in \mathcal{S}} \psi_S$$

is satisfied with probability $p$ on such structures. But now the sentence

$$\varphi_{\mathcal{G}} \to (\chi \wedge \psi_{\mathcal{S}})$$

is in $\mathsf{BP}_{(\alpha,\beta]}\mathsf{FO}[\tau]$ if and only if $\chi$ is not satisfiable on $\mathcal{G}$. $\square$

For each $n$, let $S_n$ be the $\emptyset$-structure with universe $V(S_n) := \{1, \ldots, n\}$. Recall the 0-1-law for first order logic [12, 14]. In our terminology, it says that for each vocabulary $\rho$ and each sentence $\varphi \in \mathsf{FO}[\rho]$ it holds that

$$\lim_{n \to \infty} \Pr_{X \in \mathcal{X}(S_n, \rho)} (X \models \varphi) \in \{0, 1\}$$

(in particular, this limit exists). There is also an appropriate asymptotic law for formulas with free variables. This implies that on structures with empty vocabulary, $\mathsf{PFO}$ (and in particular $\mathsf{BPFO}$) has the same expressive power as $\mathsf{FO}$. As there is also a 0-1-law for the logic $\mathsf{L}^{\omega}_{\infty\omega}$ [23], we actually get the following stronger statement:

**Observation 3.4.** Every formula $\varphi \in \mathsf{PL}^{\omega}_{\infty\omega}[\emptyset]$ is equivalent to a formula $\varphi' \in \mathsf{FO}[\emptyset]$.

As $\mathsf{FO{+}C}$ is strictly stronger than $\mathsf{FO}$ even on structures of empty vocabulary, this observation implies that there are queries definable in $\mathsf{FO{+}C}$, but not in $\mathsf{(B)PL}^{\omega}_{\infty\omega}$.

Furthermore, the Sipser-Gács Theorem [24] that $\mathsf{BPP} \subseteq \Sigma^p_2 \cap \Pi^p_2$, the fact that the fragment $\Sigma^1_2$ of second-order logic captures $\Sigma^p_2$ [11, 36], and the observation that $\mathsf{BPFO} \leqq \mathsf{BPP}$ imply the following:

**Observation 3.5.**
$$\mathsf{BPFO} \leqq \Sigma^1_2.$$

We will use Lautemann's proof of the Sipser-Gács Theorem in section 5 in the context of monadic second-order logic.

We close this section by observing that randomised logics *without* probability gaps are considerably more powerful than their non-randomised counterparts:

**Observation 3.6.** Let $\mathcal{K}$ be a class of finite structures such that there is a first-order formula $\varphi_c(x)$ defining a single element in each structure of $\mathcal{K}$. Then every $\Sigma^1_1$-query on $\mathcal{K}$ can be defined in $\mathsf{PFO}$.

*Proof.* Let $\varphi$ be a $\Sigma^1_1$-query on $\mathcal{K}$, i.e., $\varphi$ is of the form $\exists X_1 \cdots \exists X_k \psi$, where the $X_i$ are relation variables and $\psi$ is first-order. We replace each of the $X_i$ by a random relation $R_i$ of the same arity to get a new sentence $\varphi'$ and introduce an extra unary random relation $R_0$. Then $\varphi$ is equivalent to the $\mathsf{PFO}$-sentence

$$\exists x(R_0 x \wedge \varphi_c(x)) \vee \varphi',$$

because the first part is satisfied with probability exactly 1/2. □

Toda's Theorem [37] that the polynomial hierarchy is contained in $\mathsf{P}^{\mathsf{PP}}$ suggests that, in fact, every second-order query is definable in $\mathsf{PFO}$. However, Toda's proof does not carry over easily to the $\mathsf{PFO}$-case. Observation 3.4 suggests that some technical condition such as definability of an element of the structure is necessary to separate $\mathsf{PFO}$ from $\mathsf{FO}$ at all. One example of such a class $\mathcal{K}$ is the class of all ordered structures, with $\varphi_c(x)$ defining the minimum element.

## 4. Separation results for BPFO

In this section we study the expressive power of the randomised logics $\mathsf{RFO}$, $\mathsf{co\text{-}RFO}$, and $\mathsf{BPFO}$. Our main results are the following:

- $\mathsf{RFO}$ is not contained in $\mathsf{C}^{\omega}_{\infty\omega}$
- $\mathsf{BPFO}$ is not contained in $\mathsf{MSO}$ on ordered structures
- $\mathsf{RFO}$ is stronger than $\mathsf{FO}$ on additive structures

A forteriori, the first and the third result also hold with $\mathsf{BPFO}$ instead of $\mathsf{RFO}$, and the constructions used in their proofs are also definable in $\mathsf{co\text{-}RFO}$.

It turns out that we need three rather different queries to get these separation results. For the first two queries this is obvious, because *every* query on ordered structures is definable in $\mathsf{C}^{\omega}_{\infty\omega}$. The third query (on additive structures) is readily seen to be definable in $\mathsf{MSO}$. In fact, in Section 5 we show the following:

• Any BPFO-definable query on additive structures can be defined in MSO.

### 4.1. **RFO is not contained in $\mathsf{C}^{\omega}_{\infty\omega}$**

Formulas of the logic $\mathsf{C}^{\omega}_{\infty\omega}$ may contain arbitrary (not necessarily finite) conjunctions and disjunctions, but only finitely many variables, and counting quantifiers of the form $\exists^{\geq n} x\ \varphi$ ("there exists at least $n$ $x$ such that $\varphi$"). For example, the class of finite structures of even cardinality can be defined in this logic by the sentence

$$\bigvee_{k\geq 0} \left(\exists^{\geq 2k} x\ x \doteq x\right) \wedge \neg \left(\exists^{\geq 2k+1} x\ x \doteq x\right).$$

**Theorem 4.1.** *There is a class $\mathcal{TCFI}$ of structures that is definable in* RFO *and* co-RFO*, but not in* $\mathsf{C}^{\omega}_{\infty\omega}$.

Recall that by Observation 3.4 there also is a class of structures definable in FO+C $\leq \mathsf{C}^{\omega}_{\infty\omega}$, but not in BPFO.

Our proof of Theorem 4.1 is based on a well-known construction due to Cai, Fürer, and Immerman [6], who gave an example of a Boolean query in P that is not definable in $\mathsf{C}^{\omega}_{\infty\omega}$. We modify their construction in a way reminiscent to a proof by Dawar, Hella, and Kolaitis [7] for results on implicit definability in first-order logic, and obtain a query $\mathcal{TCFI}$ definable in (co-)RFO, but not in $\mathsf{C}^{\omega}_{\infty\omega}$. Just like in Cai, Fürer and Immerman's original proof, the reason why $\mathsf{C}^{\omega}_{\infty\omega}$ can not define our query $\mathcal{TCFI}$ is its inability to choose one out of a pair of two elements. Using a random binary relation this can – with high probability – be done in FO.

We first review the construction of [6] and then show how to modify it to suit our needs. Given a graph $G = (V, E)$, Cai et al. construct a new graph $G'$, replacing all vertices and edges of $G$ with certain gadgets. We shall call graphs $G'$ resulting in this fashion *CFI-graphs*, and will from now on restrict ourselves to connected 3-regular graphs $G$ and CFI-graphs resulting from these.

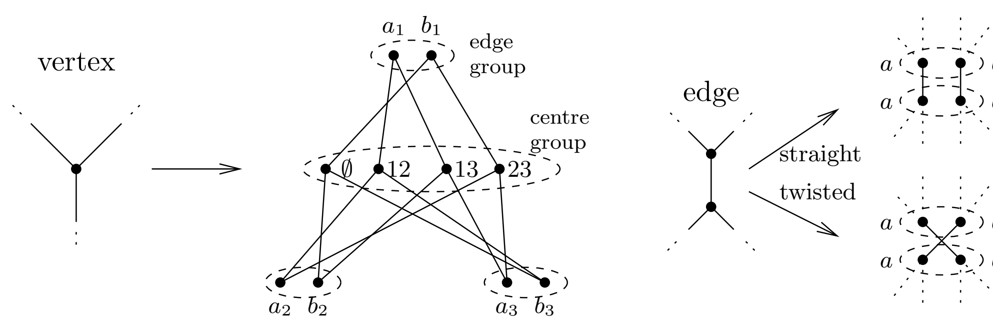


Figure 1: The gadgets for CFI-graphs. Dashed ellipses indicate groups of equivalent vertices. Vertex labels are not part of the actual structure.

The construction is as follows: For each vertex in $G$, we place a copy of the gadget shown on the left of Figure 1 in $G'$. It has a group of four nodes (henceforth called *centre nodes*) plus three pairs of nodes, which are to be thought of as ends of the three edges incident with that node. For the time being, we think of the pairs as ordered from 1 to

3 and distinguish between the two nodes in each pair, say one of them is the $a$-node, the other one being the $b$ node. Each of the four centre nodes is connected to one node from each pair, and each of them to an even number of $a$'s. To illustrate this, the centre nodes are labelled with the even subsets of $\{1, 2, 3\}$. We also introduce an equivalence relation (or colouring, if you like) of nodes as shown in Figure 1, so any isomorphism of the gadget necessarily permutes nodes within each edge group and the centre group.

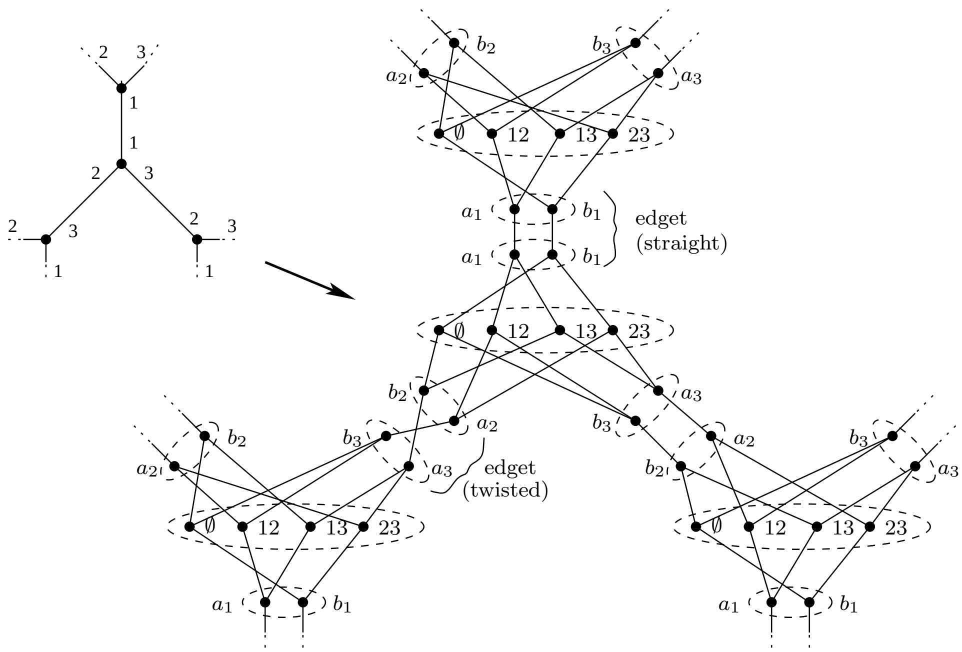


Figure 2: The CFI-graph construction for a part of a graph. Edge and nodes labels are not part of the actual graph.

For each edge in $G$, we connect the $a$- and $b$-nodes in the corresponding pairs as shown on the right of Figure 1. We say an edge is "twisted" if the $a$-node of one pair is connected to the $b$-node of the other and vice versa. This completes our construction of $G'$. For definiteness, when we speak of an *edge group* we mean an equivalence class of size two, and by a *centre group* we mean one of size four. An *edget* is a pair of edge groups which form an edge gadget as on the right of Figure 1. Figure 2 shows the result of applying this construction to a small subgraph (a vertex with its three neighbours).

Without the $a$- and $b$-labels, we cannot decide which of the edges have been twisted. In fact there are only two isomorphism classes of CFI-graphs derived from $G$, namely those with an even number of edges twisted and those with an odd number (we call the latter ones *twisted* CFI-graphs). This relies on the fact that isomorphisms of the gadget on the left of Figure 1 are exactly those permutations swapping an even number of $a$'s and $b$'s. Since we assume $G$ to be connected, we can twist edges along a path between two nodes adjacent to twisted edges, reducing the number of twisted edges by two; cf. [6, Lemma 6.2] for details.

By [6, Thm. 6.4], if the original graph $G$ has no separator of size at most $s$ then the two isomorphism classes of CFI graphs derived from it can not be distinguished by a sentence

$\varphi \in \mathsf{C}^{s}_{\infty\omega}$, i.e., by a $\mathsf{C}^{\omega}_{\infty\omega}$ sentence with at most $s$ distinct variables. In $\mathsf{P}$, on the other hand, twisted CFI-graphs can easily be recognised: Choose exactly one node from each edge group and label this one $a$ and the other one $b$. A centre node is connected to an even number of $a$'s if and only if all four nodes in its centre group are. In this case we call the centre group even, otherwise we call it odd. Then a CFI-graph is twisted if and only if

$$\text{(number of odd centre groups} + \text{number of twisted edgets) is odd.}$$

We aim for a (co-)$\mathsf{RFO}$-sentence which defines exactly the twisted connected 3-regular CFI-graphs. In view of the above $\mathsf{P}$-algorithm, we are done if we can

- express connectedness of the graph,
- count edgets and centre groups modulo two and
- choose one representative from each centre group, edge group and edget.

For counting modulo two and to get representatives for centre groups and edgets, we augment the structures with a Boolean algebra in the following way: Let $\tau$ be the vocabulary $\{E, \sim, <, \sqsubseteq, P, O\}$, with unary $P$ and $O$, and binary $E$, $\sim$, $<$ and $\sqsubseteq$. Let $\mathcal{CFI}$ be the class of structures $A$ such that

- $E$ defines a 3-regular, connected CFI-graph on $V(A) \setminus P(A)$,
- $(P(A), \sqsubseteq)$ is a Boolean algebra $\mathfrak{B}$, and $O$ is true exactly for its members of even cardinality
- $<$ defines a linear order on the set of atoms of $\mathfrak{B}$ (and no other element of $A$ is $<$-related to any other).
- $\sim$ defines an equivalence relation, where each equivalence class
  - contains one atom of $\mathfrak{B}$ and the nodes of one edget
  - or contains one atom of $\mathfrak{B}$ and the nodes of one centre group
  - or consists of a single non-atom of $\mathfrak{B}$.

  In particular, the number of atoms of the Boolean algebra $\mathfrak{B}$ is equal to the number of edgets plus the number of centre groups. Note also that we can distinguish the two edge groups in an edget because only nodes in the same edge group are connected to nodes in the same centre group.

**Theorem 4.2.** *The class $\mathcal{CFI}$ is definable in* $\mathsf{FO}$*. The subclass $\mathcal{TCFI}$ of twisted CFI-graphs is definable in* $\mathsf{BPFO}$ *but not in* $\mathsf{C}^{\omega}_{\infty\omega}$.

*Proof.* That $\mathcal{CFI}$ is definable is easy to establish, the only subtlety being that $\mathfrak{B}$ allows us to quantify over sets of centre groups, which makes connectedness expressible.

The proof that $\mathcal{TCFI}$ is not definable in $\mathsf{C}^{\omega}_{\infty\omega}$ is the same as in [6]; it is unaffected by the additional structure. Note that because the atoms are ordered, the Boolean algebra is rigid, i.e., it has no non-trivial automorphism, therefore the isomorphism group of a CFI-graph is not changed by adding the Boolean algebra.

It remains to show that twistedness can be defined in $\mathsf{BPFO}$. We pick one vertex from each edge group by viewing a random binary relation $R$ as assigning an $m$-bit number to each vertex, where $m$ is the number of atoms in the Boolean algebra. From each pair, we choose the vertex with the smaller number, expressed by

$$\xi(x) := \exists y\Big(x \sim y \wedge \exists z\big(\alpha(z) \wedge \neg Rxz \wedge Ryz \wedge \forall w(w < z \to (Rxw \leftrightarrow Ryw))\big)\Big),$$

where $\alpha(x)$ is an $\mathsf{FO}$-formula satisfied exactly by the atoms of the Boolean algebra. It is easy to see that if the random relation $R$ assigns a different set of atoms to the two vertices in each edge group, then $\xi$ succeeds in picking exactly one vertex from each edge group, and

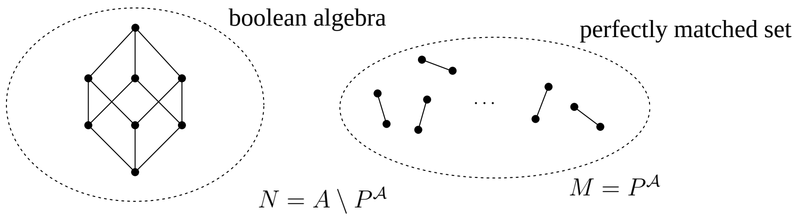


Figure 3: The structures in $\mathcal{B}$ contain a Boolean algebra and a perfectly matched set.

twistedness can then be checked by looking at the $O$-predicate of the element of $\mathfrak{B}$ which contains exactly the atoms equivalent to twisted centre groups or twisted edgets.

To prove that the resulting formula has a large probability gap, we need to establish a high probability of success only for structures in the class $\mathcal{CFI}$, because this class is FO-definable. But in such structures, the probability that the two nodes of an edge group are assigned the same number is $2^{-m}$, so by a union bound the probability that we successfully pick one node from each group is at least

$$1 - m2^{-m} \to 1$$

because there are less than $m$ edgets. Furthermore, we can check in FO whether there is an edge group whose members we can not distinguish, and choose to invariably reject or accept in these cases, resulting in an RFO or co-RFO sentence, respectively. □

### 4.2. **BPFO on ordered structures is not contained in MSO**

In the presence of a linear order, *any* query becomes definable in $\mathsf{L}^{\omega}_{\infty\omega}$, and the query $\mathcal{TCFI}$ becomes definable even in FO. However, randomisation adds expressive power to FO also on ordered structures:

**Theorem 4.3.** *There is a class $\mathcal{B}$ of ordered structures that is definable in* BPFO*, but not in* MSO*.*

Remember that monadic second-order logic MSO is the the fragment of second-order logic that allows quantification over individual elements and sets of elements.

Let $\sigma_{EP\leq} := \{\leq, E, P\}$, with binary relations $\leq$ and $E$, and a unary predicate $P$. We define two classes $\mathcal{B}'$, $\mathcal{B}$ of $\sigma_{EP\leq}$-structures (cf. Figure 4.2):
$\mathcal{B}'$ is the class of all $\sigma_{EP\leq}$-structures $A$ for which
(1) $E$ defines a perfect matching on the set $M := P(A)$
(2) the set $N := V(A) \setminus P(A)$ forms a Boolean algebra with the relation $E$ and
(3) no $x \in N$ and $y \in M$ are $E$-related
(4) $\leq$ defines a linear order on the whole structure, which puts the $M$ before the $N$ and orders $M$ in such a way that matched elements are always successive.

It is easy to see that the class $\mathcal{B}'$ is definable in FO. $\mathcal{B}$ is the subclass of $\mathcal{B}'$ whose elements satisfy the additional condition

$$2^{|M|} \geq |N|^2 \,. \tag{4.1}$$

We will prove that $\mathcal{B}$ is definable in BPFO, but not in MSO. To prove that $\mathcal{B}$ is definable in BPFO, we will use the following lemma:

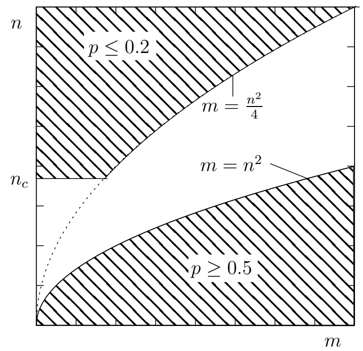


Figure 4: The Birthday Paradox with $\epsilon_1 = 0.2$, $\epsilon_2 = 0.5$ and $c = 4$. Here, $p$ denotes $\Pr(f \text{ is injective})$.

**Lemma 4.4** (Birthday Paradox)**.** *Let $m, n \geq 1$ and let $F : [n] \to [m]$ be a random function drawn uniformly from the set of all such functions.*

(1) *For any $\epsilon_1 > 0$ and $c > 2 \ln \frac{1}{\epsilon_1}$ there is an $n_c \geq 1$ such that if $n > n_c$ and $m \leq \frac{n^2}{c}$ we have*
$$\Pr(F \textit{ is injective}) \leq \epsilon_1$$

(2) *For any $\epsilon_2 > 0$, if $m \geq \frac{n^2}{2\epsilon_2}$, then*
$$\Pr(F \textit{ is injective}) \geq 1 - \epsilon_2$$

*Proof.* For the first part, we note that
$$\Pr(F \text{ injective}) = \prod_{i=0}^{n-1} \left(1 - \frac{i}{m}\right) \leq \prod_{i=0}^{n-1} \exp\left(-\frac{i}{m}\right) = \exp\left(-\frac{n(n-1)}{2m}\right).$$
For the second part, note that
$$\Pr(F \text{ not injective}) = \Pr\Big(F(i) = F(j) \text{ for all } i < j\Big) \leq \sum_{i<j} \frac{1}{m} = \binom{n}{2} \frac{1}{m} \leq \frac{n^2}{2m}. \qquad \square$$

*Proof of Theorem 4.3.* To see that $\mathcal{B}$ is not definable in MSO, we use two simple and well-known facts about MSO. The first is that for every $q \geq 0$ there are natural numbers $p, m$ such that for all $k \geq 0$, a plain linear order of length $m$ is indistinguishable from the linear order of length $m + k \cdot p$ by MSO-sentences of quantifier rank at most $q$. The same fact also holds for linear orders with a perfect matching on successive elements, because such a matching is definable in MSO anyway. The second fact we use is a version of the Feferman-Vaught Theorem (cf. [27, Thm. 1.5(ii)]):

**Theorem 4.5.** *Suppose two $\tau$-structures $U$ and $V$ satisfy the same* MSO*-sentences of quantifier rank up to $q$, and let $W$ be another $\tau$-structure. Denote by $U \sqcup W$ (resp. $V \sqcup W$) the disjoint union of $U$ (resp. $V$) and $W$. Then $U \sqcup W$ and $V \sqcup W$ satisfy the same* MSO*-sentences of quantifier rank up to $q$.*

The theorem also holds for the ordered disjoint union $\sqcup_<$ instead of the disjoint union, but in our case the elements of the individual structures in the disjoint union are definable anyway. If we put these two facts together, we see that for every $q \geq 0$ there are $p, m$

such that for all $k, n$ the structure $A \in \mathcal{B}$ with parts $M, N$ of sizes $m$, $n$, respectively, is indistinguishable from the structure $A'$ with parts of sizes $m + k \cdot p$ and $n$. We can easily choose $k$ and $n$ in such a way that $A \in \mathcal{B}$ and $A' \notin \mathcal{B}$.

It remains to prove that $\mathcal{B}$ is definable in $\mathsf{BPFO}$. Consider the sentence

$$\varphi_{\text{inj}} := \forall x \forall y \Big( x \doteq y \vee Px \vee Py \vee \exists z \big( Pz \wedge \neg (Rxz \leftrightarrow Ryz) \big) \Big),$$

which states that the random binary relation $R$, considered as a function

$$f : N \to \mathrm{Pow}(M), \quad x \mapsto \{ y \in M \mid Rxy \}$$

from $N$ to subsets of $M$, is injective. By the definition of $R$, the function $f$ is drawn uniformly from the set of all such functions. If we fix $|N|$, the probability for $f$ to be injective increases monotonically with $|M|$. Furthermore, for every structure in $\mathcal{B}'$, the size of $N$ and $M$ are a power of two and an even number, respectively. Thus either

$$2^{|M|} \leq \frac{1}{4} |N|^2 \quad \text{or} \quad 2^{|M|} \geq |N|^2 \,,$$

and this factor of 4 translates into a probability gap for $\varphi_{\text{inj}}$ in all sufficiently large structures in $\mathcal{B}'$, by Lemma 4.4 with $\epsilon_1 = 0.2$, $\epsilon_2 = 0.5$ and $c = 4$. The remaining finitely many structures in $\mathcal{B}'$ can be dealt with separately. $\square$

### 4.3. RFO is stronger than FO on additive structures

Recall that an additive structure is one whose vocabulary contains a ternary relation $+$, such that $A|_+$ is isomorphic to $([0, |A| - 1], \{(a, b, c) \mid a + b = c\})$.

**Theorem 4.6.** *There is a class $\mathcal{A}$ of additive structures that is definable in* $\mathsf{RFO}$ *and* co-$\mathsf{RFO}$*, but not in* $\mathsf{FO}$*.*

Our proof uses the following result:

**Theorem 4.7** (Lynch [26])**.** *For every $k \in \mathbb{N}$ there is an infinite set $A_k \subseteq \mathbb{N}$ and a $d_k \in \mathbb{N}$ such that for all finite $Q_0, Q_1 \subseteq A_k$ with $|Q_0| = |Q_1|$ or $|Q_0|, |Q_1| > d_k$ the structures $(\mathbb{N}, +, Q_0)$ and $(\mathbb{N}, +, Q_1)$ satisfy exactly the same* $\mathsf{FO}$*-sentences of quantifier rank at most $k$.*

Here $(\mathbb{N}, +, Q_i)$ denotes a $\{+, P\}$-structure with ternary $+$ and unary $P$, where $+$ is interpreted as above and $P$ is interpreted by $Q_i$. For a finite set $M \subseteq \mathbb{N}$ we denote by $\max M$ the maximum element of $M$. By relativising quantifiers to the maximum element satisfying $P$, we immediately get the following corollary:

**Corollary 4.8.** *Let $k$, $A_k$, $d_k$, $Q_0$ and $Q_1$ be as above. Then the (finite) structures $([0, \max Q_0], +, Q_0)$ and $([0, \max Q_1], +, Q_1)$ satisfy exactly the same* $\mathsf{FO}$*-sentences of quantifier rank at most $k$.*

We call a set $Q \subseteq \mathbb{N}$ *sparse* if $|Q \cap \{n, \ldots, 3n\}| \leq 1$ for all $n \geq 0$. Note that if $Q$ is sparse and finite, then $|Q| \leq \log_3(\max Q) + 1$. It is easy to see that there is an $\mathsf{FO}[\{+, P\}]$-sentence $\varphi_{\text{sparse}}$ such that

$$([0, \max Q], +, Q) \models \varphi_{\text{sparse}} \quad \Leftrightarrow \quad Q \text{ is sparse}$$

for all finite $Q \subseteq \mathbb{N}$.

*Proof of Theorem 4.6.* We define the following class of additive $\{+, P\}$-structures:

$$\mathcal{A} = \{([0, \max Q], +, Q) \mid Q \text{ is finite, sparse and } |Q| \text{ is even}\},$$

with $+$ defined as usual. It follows immediately from Corollary 4.8 that $\mathcal{A}$ is not definable in $\mathsf{FO}$.

It remains to prove that $\mathcal{A}$ is definable in (co-)RFO. We consider a binary random relation $R$ on $\mathcal{Q} = ([0, \max Q], +, Q)$ for some finite $Q \subseteq \mathbb{N}$.

Each element $a \in [0, \max Q]$ defines a subset of $Q$, namely the set of $b \in Q$ for which $(a, b) \in R(\mathcal{Q})$ holds. If $Q$ is a sparse set, it has

$$2^{|Q|} \leq 2^{\log_3(\max Q)+1} \leq \frac{\max Q}{2 \ln(\max Q)}$$

many subsets, and by standard estimates on the coupon collector's problem (see, e.g., [29]; or use a union-bound argument), if $\max Q$ is large enough, with high probability every subset of $Q$ is defined by some element of $[0, \max Q]$. We may check in FO whether this is actually the case. If so, we use the random relation $R$ and the linear order induced by $+$ to check whether $Q$ is even. Otherwise we reject (accept) to get an RFO- (co-RFO-)sentence. $\square$

## 5. BPFO is contained in MSO on additive structures

In this section, we prove our first and only nontrivial derandomisation result. It complements the result of Section 4.2 by saying that, on additive structures, every BPFO-sentence is equivalent to an MSO-sentence.

**Theorem 5.1.** *Let $\tau$ be a finite relational vocabulary containing a ternary relation $+$ and let $\varphi$ be a* BPFO$[\tau]$*-sentence. Then there exists an* MSO*-sentence $\psi$ such that on additive structures $A$*

$$A \models \varphi \quad \Leftrightarrow \quad A \models \psi.$$

We first use Nisan's pseudorandom generator for constant depth circuits [32] to reduce the number of random bits to $\log^{O(1)} n$; throughout this section, $n$ will denote the size of the input structure. We then derandomise the resulting formula following Lautemann's argument in [24]. The second-order quantifier depth of the resulting MSO formula does not depend on the input formula $\varphi$.

In MSO$[+]$, one can define a multiplication relation (see [35, Lemma 5.4]) and thus quantify over pairs of elements in $[0, \sqrt{n}]$. We only need the existence of such a pairing function, a slightly weaker form of which is made precise in the following lemma:

**Lemma 5.2** (Pairing Lemma)**.** *There are* MSO$[+]$*-formulas $\varphi_p(x)$ and $\varphi_{\langle\cdot,\cdot,\cdot\rangle}(x, y, z, w)$ such that on additive structures $A$*

- *$\varphi_p(x)$ defines a number $p$ satisfying*

  $$\frac{\sqrt{|A|}}{2} \leq p \leq \sqrt{|A|}.$$

  *Moreover, $p$ is a prime number.*
- *For every $b, c < p$ there is a unique $m$ such that $\varphi_{\langle\cdot,\cdot,\cdot\rangle}(0, b, c, m)$ is satisfied. Furthermore, for every $m$ there is a unique tuple $(a, b, c) \in [0, p-1]^3$ such that $\varphi_{\langle\cdot,\cdot,\cdot\rangle}(a, b, c, m)$ is satisfied. Henceforth we write $m = \langle a, b, c\rangle$ for this.*

*Proof.* In $\mathsf{MSO}[+]$, we may define a formulas $\varphi_{X=\langle x\rangle}(X,x)$ and $\varphi_{\text{divides}}(x,y)$ stating that $X$ is the set of multiples of $x$ and $x$ divides $y$, respectively. We may thus check whether $x$ is a prime number. Furthermore, we may define the set of powers of a prime number $x$: It is the largest set containing only numbers whose only prime divisor is $x$.

Then $p$ is the largest prime number whose set of powers contains at least one element other that 0 and itself. Any number $m \in [0, p^2-1]$ may be written as $m = bp + c$ with $b, c \in [0, p-1]$. Both $b$ and $c$ are definable in $\mathsf{MSO}[+]$; notice that $b$ is the largest divisor of $m - c$ smaller than $p$, or 0 if $m < p$. For $m \geq p^2$ we define $m = \langle a, b, c\rangle$ with $a \in \{1, 2, 3\}$ and $m - ap^2 = \langle 0, b, c\rangle$. $\square$

Whenever we write $p$ in this section, we mean the $p$ defined by the $\varphi_p$ above. The Pairing Lemma allows us to quantify over binary relations on $[0, p-1] \cong \mathbb{F}_p$. In particular, we may define addition and multiplication modulo $p$, i.e., there are $\mathsf{MSO}[+]$-formulas $\varphi_+(x,y,z)$ and $\varphi_\times(x,y,z)$ such that for $a,b,c \in \mathbb{F}_p$,

$$A \models \varphi_+(a,b,c) \quad \Leftrightarrow \quad a+b \equiv c \pmod p$$

and

$$A \models \varphi_\times(a,b,c) \quad \Leftrightarrow \quad a \cdot b \equiv c \pmod p.$$

For the proof of Theorem 5.1 we may assume that the $\mathsf{BPFO}$-sentence $\varphi$ contains only one random relation, say $R$ of arity $r$. In fact, using the formulas $\varphi_{i\text{-th}}$ defining the $i$-th element of an additive structure (cf. section 2.1) we may pack several random relations $R_1, \ldots, R_k$ of arities $r_1, \ldots, r_k$ into one random relation $R$ of arity $r = 1 + \max\{r_1, \ldots, r_k\}$ by replacing every occurrence of $R_i x_1 \ldots x_{r_i}$ by

$$\exists y\, (\varphi_{i\text{-th}}(y) \wedge R \underbrace{y \ldots y}_{(r-r_i)\text{ times}} x_1 \ldots x_{r_i}).$$

We first apply a result by Nisan [32] to reduce the number of random bits:

**Lemma 5.3.** *For every $r, d \in \mathbb{N}$ and $\epsilon > 0$ there are $n_0 \in \mathbb{N}$ and $\mathsf{MSO}[+]$-formulas $\varphi_l(x)$ and $\varphi_{\text{prg}}(S, x_1, \ldots, x_r)$, where $S$ is a set variable, such that in every additive structure $A$ of size $n > n_0$,*

- *$\varphi_l$ defines a number $l \leq \log^{O(1)} n$ and*
- *if $\varphi$ is an $\mathsf{FO}[\tau \cup \{R\}]$-sentence of quantifier rank $\leq d$, where $\tau$ is some finite relational vocabulary and $R$ is of arity $r$, then*

$$\left| \Pr_{X \in \mathcal{X}(A,\{R\})}(X \models \varphi) - \Pr_{S \subseteq [l]}(A \models \varphi'(S)) \right| < \epsilon,$$

*where $\varphi'$ is the $\mathsf{MSO}[+]$-formula obtained from $\varphi$ by replacing every occurrence of $R\vec{x}$ by $\varphi_{\text{prg}}(S, \vec{x})$.*

*Proof.* For any fixed structure $A$ of size $n$ we may construct a polynomial-sized circuit $C_{\varphi,A}$ of depth $\leq d$ which describes the behaviour of $\varphi$ on $(\tau \cup \{R\})$-extensions of $A$. The circuit has $n^r$ inputs indexed by the elements of $V(A)^r$, and an input vector $\vec{x}$ denotes the $(\tau \cup \{R\})$-extension $B_{\vec{x}}$ of $A$ given by

$$\vec{a} \in R(B_{\vec{x}}) \quad \text{iff} \quad x_{\vec{a}} = 1.$$

Then $C_{\varphi,A}(\vec{x})$ evaluates to 1 iff $B_{\vec{x}} \models \varphi$.

Nisan [32] gave a pseudorandom generator for such circuits which hinges on the following lemma:

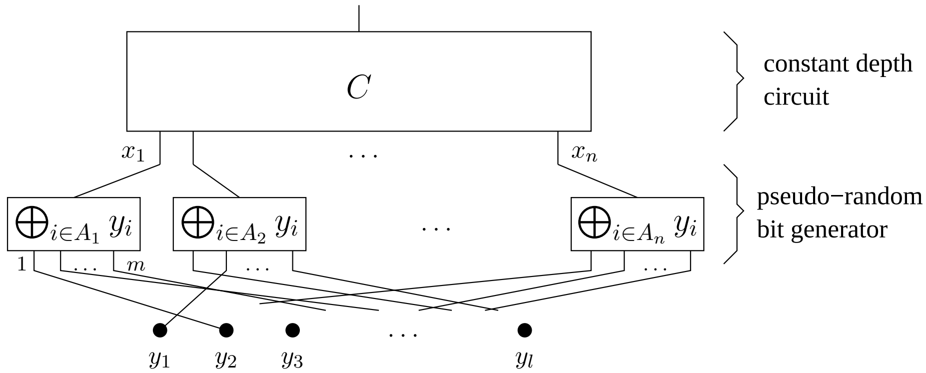


Figure 5: Nisan's pseudo-random bit generator. The sets $A_i \subseteq \{1, \dots, l\}$ form a partial-$(\log n, m)$-design, i.e., they satisfy $|A_i| = m$ and $|A_i \cap A_j| \leq \log n$ for all $1 \leq i \neq j \leq n$.

**Lemma 5.4** (restated from [32, Lemma 2.2])**.** *Let $\{C_n\}$ be a family of circuits of depth $d$ and polynomial size, let $m = m(n) = (\log n)^{d+3}$, $l = l(n)$ and suppose for each $n$ the sets $A_1^{(n)}, \dots, A_n^{(n)} \subseteq [l]$ satisfy*

- $\left|A_i^{(n)}\right| = m$ *for all* $1 \leq i \leq n$ *and*
- $\left|A_i^{(n)} \cap A_j^{(n)}\right| \leq \log n$ *for all* $1 \leq i \neq j \leq n$.

*Then*

$$|\Pr(C_n(\vec{x}) = 0) - \Pr(C_n(\oplus_{i\in A_1} y_i, \dots, \oplus_{i\in A_n} y_i) = 0)| \leq \frac{1}{n^c}$$

*for any $c \in \mathbb{N}$ and large enough $n$. Here, the first probability is taken uniformly over all strings $\vec{x} \in \{0,1\}^n$, whereas the second is taken uniformly over all strings $\vec{y} \in \{0,1\}^l$.*

The resulting pseudorandom generator is depicted in Figure 5. Families of sets $A_i^{(n)}$ satisfying the above conditions are called *partial-*$(\log n, m)$*-designs*. Nisan gives a construction with $l = m^2 = \log^{O(1)} n$, which drastically reduces the size of the probability space, i.e., the number of random bits needed. We now show how his construction can be defined in $\mathsf{MSO}[+]$.

On $[0, p-1]$, we may define a formula $\varphi_{\log}(x, y)$ which is satisfied iff $x = \lceil \log_2 y \rceil$. Using this and the fact that

$$2\lceil \log p\rceil - 1 \leq \lceil \log n\rceil \leq 2\lceil \log p\rceil + 2,$$

we let $\varphi_m(x)$ and $\varphi_l(x)$ be two formulas defining natural numbers $m$ and $l$ such that

- $m$ is a prime number between $(r^2\lceil \log n\rceil)^{d+3}$ and $2(r^2(\lceil \log n\rceil + 3)^{d+3}$
- $l = m^2$

Using the pairing function $\varphi_{\langle\cdot,\cdot,\cdot\rangle}$ we may assume that $R$ is a $3r$-ary relation which we only need to define for elements in $\mathbb{F}_p$. That is, we define $\varphi_{\text{prg}}(S, x_1, \dots, x_r)$ by

$$\exists z_1 \cdots \exists z_{3r}\, x_1 = \langle z_1, z_2, z_3\rangle \wedge \ldots \wedge x_r = \langle z_{3r-2}, z_{3r-1}, z_{3r}\rangle \wedge \varphi'_{\text{prg}}(S, z_1, \dots, z_{3r})$$

The formula $\varphi'_{\text{prg}}(S, \vec{z})$ takes the parity of a subset of $S$ indexed by $\vec{z}$:

$$\varphi'_{\text{prg}}(S, \vec{z}) := \text{“}\,|S \cap \psi(A; \vec{z})| \text{ is even”},$$

where $\psi(x,\vec{z})$ is an $\mathsf{MSO}[+]$-formula and $\psi(A;\vec{z}) := \{x \mid A \models \psi(x,\vec{z})\}$; evenness may be expressed in $\mathsf{MSO}$ on ordered structures. By Lemma 5.4, we are done if we can define a formula $\psi(x,\vec{z})$ such that

(i) $\psi(A;\vec{z}) \subseteq [l]$ for all $\vec{z} \in \mathbb{F}_p^{3r}$,

(ii) $|\psi(A;\vec{z})| = m$ for all $\vec{z} \in \mathbb{F}_p^{3r}$, and

(iii) $|\psi(A;\vec{z}_1) \cap \psi(A;\vec{z}_2)| \leq \log n$ for all $\vec{z}_1 \neq \vec{z}_2 \in \mathbb{F}_p^{3r}$,

which means the sets $\psi(A;\vec{z})$ form a partial-$(\log n, m)$-design. We use the same construction as Nisan: We interpret the tuple $\vec{z}$ as a polynomial $f_{\vec{z}} \in \mathbb{F}_m[\xi]$ of degree $\leq \log n$. The set $\psi(A;\vec{z})$ is then the graph of this polynomial, namely

$$\psi(A;\vec{z}) = \{(\xi, f_{\vec{z}}(\xi)) \mid \xi \in \mathbb{F}_m\} \subseteq \mathbb{F}_m^2,$$

and we identify $\mathbb{F}_m^2$ with $[l]$. We first encode the coefficients of $f_{\vec{z}}$ into a set variable $X$ as follows: Consider the binary representations

$$z_i = \sum_{j \geq 0} z_{i,j} 2^j \quad \text{with } z_{i,j} \in \{0,1\}$$

of the $z_i$. We can define an $\mathsf{MSO}[+]$-sentence $\varphi_{\text{pack}}(\vec{z}, X)$ which holds iff $X$, interpreted as a binary relation over $\mathbb{F}_p$, holds exactly for pairs $(a,b)$ with

$$0 \leq a \leq \lceil \log p \rceil \quad \text{and} \quad b = \sum_{1 \leq i \leq 3r} z_{i,a} 2^{i-1}.$$

Thus for each $0 \leq a \leq \lceil \log p \rceil$ there is exactly one $b = b(a)$ with $(a,b) \in X$, and all $b$s are between 0 and $2^{3r}$, and thus in $\mathbb{F}_m$ if $n$ is large enough. We may now define an $\mathsf{MSO}[+]$-sentence $\varphi_{\text{eval}}(X,u,v)$ which, for these $X$s, holds iff

$$v = f_{\vec{y}}(u) = \sum_{0 \leq a < \lceil \log p \rceil} b(a) u^a,$$

with addition and multiplication according to $\mathbb{F}_m$. Putting these ingredients together, we define

$$\psi(x,\vec{z}) = \exists X \exists u \exists v \text{ “}0 \leq u, v < m\text{”} \wedge \varphi_{\text{pack}}(\vec{z}, X) \wedge \varphi_{\text{eval}}(X,u,v) \wedge \text{“}x = u \cdot m + v\text{”},$$

which is easily verified to satisfy conditions (i) to (iii) above. $\square$

So far we have reduced the number of random bits from $n^r$ to $l = \log^{O(1)} n$, and these are conveniently packed into the first $l$ bits of a single set variable $S$. We may now follow Lautemann's proof [24] to derandomise this sentence.

*Proof of Theorem 5.1.* After applying Lemma 5.3 we are left with $\mathsf{MSO}[+]$-sentences $\varphi_l$ and $\varphi'$ such that $\varphi_l$ defines a number $l \leq \log^{O(1)} n$ and $\varphi'$ has a free set variable $S$. We may assume that for all additive structures $A$,

$$\text{either } \Pr_{S \subseteq [l]}(A \models \varphi'(S)) < \frac{1}{l} \qquad \text{or } \Pr_{S \subseteq [l]}(A \models \varphi'(S)) > 1 - \frac{1}{l}, \tag{5.1}$$

because otherwise we may use independent repetition and majority vote to obtain these bounds. To be precise, let $\chi(S,i,j)$ be defined by

$$\chi(S,i,j) := (0 \leq i < l) \wedge (0 \leq j < l) \wedge \exists z (z \doteq i \cdot l + j \wedge Sz).$$

That is, we divide the first $l^2$ bits of $S$ into $l$ blocks of $l$ bits each, and let $\chi(S,i,j)$ select the $i$-th bit of the $j$-th block. We replace each occurrence of $Sx$ in $\varphi'$ by $\chi(S,i,x)$ to obtain a formula $\tilde{\varphi}'(S,i)$. Because $l$ is of order $\log^{O(1)} n$, we may quantify over pairs of elements of $[0, l-1]$, which allows us to express the formula

$$\bar{\varphi}'(S) = \text{“}\tilde{\varphi}'(S,i) \text{ holds for at least half of the } i \in [0, l-1]\text{”}$$

in $\mathsf{MSO}[+]$, e.g., by stating that there exists a matching $M$ on $[0, l-1]$ such that

- if $\{i,j\} \in M$, then exactly one of $\tilde{\varphi}'(S,i)$ and $\tilde{\varphi}'(S,j)$ holds and
- all $i \in [0, l-1]$ for which $\tilde{\varphi}'(S,i)$ does not hold are matched by $M$.

Then $\bar{\varphi}'$ uses $l^2 = \log^{O(1)} n$ many bits of $S$, and by the Chernoff bound on the tails of the binomial distribution it satisfies (5.1), even with $l$ replaced by $l^2$ (details can be found in [2, sec. 7.4]).

We identify subsets of $[l]$ with vectors in $\mathbb{F}_2^l$. Let $M \subseteq \mathbb{F}_2^l$ be the set of vectors for which $A \models \varphi'(S)$ holds. Equation (5.1) translates into

$$|M| < \frac{\left|\mathbb{F}_2^l\right|}{l} \quad \text{or} \quad |M| > \left(1 - \frac{1}{l}\right)\left|\mathbb{F}_2^l\right|.$$

For a vector $\vec{y} \in \mathbb{F}_2^l$ we define

$$\vec{y} \oplus M := \{\vec{x} \oplus \vec{y} \mid \vec{x} \in M\}$$

to be the set $M$ translated by $\vec{y}$. We claim the following:

(a) If $|M| < \left|\mathbb{F}_2^l\right|/l$, then for every choice of vectors $\vec{y}_1, \ldots, \vec{y}_l$ we have

$$\bigcup_{1 \le i \le l} (\vec{y}_i \oplus M) \neq \mathbb{F}_2^l.$$

(b) If $|M| > (1 - 1/l)\left|\mathbb{F}_2^l\right|$, then there are vectors $\vec{y}_1, \ldots, \vec{y}_l$ such that

$$\bigcup_{1 \le i \le l} (\vec{y}_i \oplus M) = \mathbb{F}_2^l.$$

The first claim follows immediately from $|\vec{y} \oplus M| = |M|$. For (b), assume that we randomly choose the vectors $\vec{y}_i$ independently and uniformly from $\mathbb{F}_2^l$. For any vector $\vec{x} \in \mathbb{F}_2^l$ we have

$$\begin{aligned}\Pr\left(\vec{x} \notin \bigcup(\vec{y}_i \oplus M)\right) &= \prod_i \Pr(\vec{x} \notin \vec{y}_i \oplus M)\\ &\le \left(\frac{1}{l}\right)^l,\end{aligned}$$

by the independence of the $\vec{y}_i$. But then the expected number of vectors *not* in $\bigcup(\vec{y}_i \oplus M)$ is

$$\begin{aligned}\mathbb{E}\left[\left|\mathbb{F}_2^l \setminus \bigcup(\vec{y}_i \oplus M)\right|\right] &= \sum_{\vec{x} \in \mathbb{F}_2^l} \Pr\left(\vec{x} \notin \bigcup(\vec{y}_i \oplus M)\right)\\ &\le \frac{\left|\mathbb{F}_2^l\right|}{l^l} = \left(\frac{2}{l}\right)^l < 1,\end{aligned}$$

so there must be a choice of $\vec{y}_i$s such that this number is zero, i.e., $\bigcup(\vec{y}_i \oplus M) = \mathbb{F}_2^l$.

Again using the formula $\chi(S,i,j)$, we can pack the vectors $\vec{y}_1, \ldots, \vec{y}_l$ into a single existentially quantified set variable and check that $\bigcup(\vec{y}_i \oplus M) = \mathbb{F}_2^l$ as follows:

$$\varphi'' = \exists Y \forall X \exists i\, \varphi'(X \oplus \chi(Y, i, \cdot)),$$

where $\varphi'(X \oplus \chi(Y,i,\cdot))$ is the formula $\varphi'(S)$ with every occurrence of $Sx$ replaced by

$$(Xx \wedge \chi(Y,i,x)) \vee (\neg Xx \wedge \neg\chi(Y,i,x)).$$

Claims (a) and (b) imply that

$$A \models \varphi'' \quad \Leftrightarrow \quad \Pr(A \models \varphi'(S)) > 1 - \frac{1}{l},$$

which completes the proof. □

## 6. A logic capturing BPP

In this section, we prove that the logic BPIFP+C captures the complexity class BPP. Technically, the results of this section are closely related to results in [17].

Counting logics like FO+C and IFP+C are usually defined via two-sorted structures, which are equipped with an initial segment of the natural numbers of appropriate length. The expressive power of the resulting logic turns out to be rather robust under changes in the exact definition, see [33] for a detailed survey of this. However, we will only need the limited counting ability provided by the *Rescher quantifier*, which goes back to a unary majority quantifier defined in [34], see [33].

We let $\mathsf{FO}(\mathcal{J})$ be the logic obtained from first-order logic by adjoining a generalised quantifier $\mathcal{J}$, the *Rescher quantifier*. For any two formulas $\varphi_1(\vec{x})$ and $\varphi_2(\vec{x})$, where $\vec{x}$ is a $k$-tuple of variables, we form a new formula

$$\mathcal{J}\vec{x}.\varphi_1(\vec{x})\varphi_2(\vec{x}).$$

Its semantics is defined by

$$A \models \mathcal{J}\vec{x}.\varphi_1(\vec{x})\varphi_2(\vec{x}) \quad \text{iff} \quad \Big|\{\vec{a} \in V(A)^k \mid A \models \varphi_1[\vec{a}]\}\Big| \leq \Big|\{\vec{a} \in V(A)^k \mid A \models \varphi_2[\vec{a}]\}\Big|. \tag{6.1}$$

The logic $\mathsf{IFP}(\mathcal{J})$ is defined similarly.

**Lemma 6.1.** *Let $R$ be a 6-ary relation symbol. There is a formula $\varphi_{\leq}(x,y) \in \mathsf{FO}(\mathcal{J})[\{R\}]$ such that*

$$\lim_{n\to\infty} \Pr_{A \in X(S_n,\{R\})} \Big(\{(a,b) \mid A \models \varphi_{\leq}[a,b]\} \textit{ is a linear order of } V(A)\Big) = 1.$$

*(Recall that $S_n$ is the $\emptyset$-structure with universe $\{1,\ldots,n\}$. Thus $X(S_n,\{R\})$ just denotes the set of all $\{R\}$-structures with universe $\{1,\ldots,n\}$.)*

*Proof.* We let

$$\varphi_{\leq}(x,y) := \mathcal{J}x_1\ldots x_5.Rxx_1\ldots x_5\, Ryx_1\ldots x_5.$$

To see that $\varphi_{\leq}(x,y)$ defines an order with high probability, let $A$ be a structure with universe $V(A) = \{1,\ldots,n\}$. For each $a \in V(A)$, let

$$X_a := \Big|\{\vec{a} \in V(A)^5 \mid A \models Ra\vec{a}.\}\Big|$$

Then $A \models \varphi_{\leq}(a,b)$ iff $X_a \leq X_b$, and $\varphi_{\leq}$ linearly orders $A$ iff the $X_a$ are pairwise distinct. But for $a \neq b \in V(A)$, the random variables $X_a$ and $X_b$ are independent and each is binomially distributed with parameters $p = 1/2$ and $m = n^5$, and thus

$$\Pr(X_a = X_b) = \sum_{k=0}^{m} \left( \frac{1}{2^m} \binom{m}{k} \right)^2 = \frac{1}{2^{2m}} \sum \binom{m}{k}^2$$

$$= \frac{1}{2^{2m}} \sum \binom{m}{k} \binom{m}{m-k} = \frac{1}{2^{2m}} \binom{2m}{m} = \Theta\left( \frac{1}{\sqrt{m}} \right),$$

where the final approximation can be found, for example, in [13]. The second part now follows by a union bound over the $\binom{n}{2} = \Theta(m^{2/5})$ pairs $a \neq b$. $\square$

**Theorem 6.2.** *The logic* $\mathsf{BPIFP}(\mathcal{J})$ *captures* $\mathsf{BPP}$.

*Proof.* $\mathsf{BPIFP}(\mathcal{J})$ is contained in $\mathsf{BPP}$, because a randomised polynomial time algorithm can interpret the random relations by using its random bits.

For the other direction, let $\mathcal{Q}$ be a Boolean query in $\mathsf{BPP}$. This means that there is a randomised polynomial time algorithm $M$ that decides the query $\mathcal{Q}_{\leq}$ of ordered expansions of structures in $\mathcal{Q}$. We may view the (polynomially many) random bits used by $M$ as part of the input. Then it follows from the Immerman-Vardi Theorem that there is a $\mathsf{BPIFP}$-sentence $\psi_M$ defining $\mathcal{Q}_{\leq}$. Note that, by the definition of $\mathcal{Q}_{\leq}$, this sentence is order-invariant. We replace every occurrence of $\leq$ in $\psi_M$ by the formula $\varphi_{\leq}(x, y)$ of Lemma 6.1, which with high probability defines a linear order on the universe. $\square$

It is easy to see that $\mathsf{BPIFP{+}C}$ is also contained in $\mathsf{BPP}$ and that $\mathsf{IFP}(\mathcal{J}) \leqq \mathsf{IFP{+}C}$. Thus we get the following corollary.

**Corollary 6.3.** $\mathsf{BPIFP{+}C} = \mathsf{BPIFP}(\mathcal{J})$*, and both capture* $\mathsf{BPP}$.

**Remark 6.4.** Lemma 6.1 also implies that $\mathsf{BPL}^{\omega}_{\infty\omega}(\mathcal{J}) \equiv \mathsf{BPC}^{\omega}_{\infty\omega}$, because, in the presence of an ordering, a quantifier of the form $\exists^{\geq n} x\, \varphi$ may be spelled out as

$$\bigvee_{\substack{S \subset \mathbb{N} \\ |S| = n}} \bigwedge_{i \in S} \exists x\, (\varphi_{i\text{-th}}(x) \wedge \varphi(x)),$$

where $\varphi_{i\text{-th}}(x)$ defines $i$-th element in the linear order (cf. section 2.1).

In fact, because the formulas $\varphi_i$ use only three distinct variables independent of $i$, *any* query is definable in $\mathsf{L}^{\omega}_{\infty\omega}$ on ordered structures, as well as on $\mathsf{BPC}^{\omega}_{\infty\omega}$.

## 7. Summary and Open Problems

Our main motivation for introducing randomised logics was to apply tools from finite model theory to problems in computational complexity theory, and possibly vice versa. Because most capturing results from descriptive complexity remain valid when both the logic and the complexity class they involve are randomised in the same way, our definitions are indeed suitable for this purpose. In particular, the capturing results by Barrington et al. [3] for $\mathsf{FO}[+, \times]$ and Behle and Lange [5] for $\mathsf{FO}[\leq]$ and $\mathsf{FO}[+]$ fall into this category.

This asks for a more detailed investigation of the expressive power of randomised logics. For example, we have shown that $\mathsf{BPFO}[+]$ can not be derandomised, while conditional derandomisation results for dlogtime-uniform $\mathsf{BPAC}^0$ (cf. [39]) suggest that $\mathsf{BPFO}[+, \times]$ might be derandomisable. As this question seems to elude currect techniques, a first step might be to find *some* relation $R$ for which $\mathsf{BPFO}[R]$ is derandomisable. Note that derandomisability of non-uniform $\mathsf{BPAC}^0$ implies the existence of an infinite sequence $(R_i)_{i\geq 1}$ of relations for which $\mathsf{BPFO}[R_1, R_2, \ldots]$ is derandomisable.

One obstruction to proving results about randomised logics is that, for example, Ehrenfeucht-Fraïssé games become quite complicated on structures with both a random and a non-random part. In [10], the first author proves some non-definability results for BPFO, namely that, on vocabularies with only unary relations, BPFO can be derandomised, and that the ordering relation $\leq$ can not be defined in BPFO from its corresponding successor relation. A natural next step would be to prove whether BPFO can be derandomised on word models or not.

## Acknowledgements

We would like to thank Nicole Schweikardt and Dieter van Melkebeek for helpful comments on an earlier version of this paper.

## References

[1] Miklos Ajtai and Michael Ben-Or. A theorem on probabilistic constant depth computations. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, STOC, pages 471–474, New York, NY, USA, 1984. ACM.

[2] Sanjeev Arora and Boaz Barak. *Computational Complexity*. Cambridge University Press, 2009.

[3] David A. Mix Barrington, Neil Immerman, and Howard Straubing. On uniformity within $NC^1$. *J. Comput. Syst. Sci.*, 41(3):274–306, 1990.

[4] J. Barwise and S. Feferman, editors. *Model Theoretic Logics*. Perspectives in Mathematical Logic. Springer-Verlage, 1985.

[5] Christoph Behle and Klaus-Jörn Lange. FO[<]-uniformity. In *IEEE Conference on Computational Complexity*, pages 183–189, 2006.

[6] J.-Y. Cai, M. Fürer, and N. Immerman. An optimal lower bound on the number of variables for graph identifications. *Combinatorica*, 12(4):389–410, 1992.

[7] Anuj Dawar, Lauri Hella, and Phokion G. Kolaitis. Implicit definability and infinitary logic in finite model theory. In *ICALP*, volume 944 of *LNCS*, pages 624–635. Springer Verlag, 1995.

[8] H.-D. Ebbinghaus. Extended logics: The general framework. In J. Barwise and S. Feferman, editors, *Model–Theoretic Logics*, pages 25–76. Springer-Verlag, 1985.

[9] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory*. Perspectives in Mathematical Logic. Springer-Verlag, 2nd edition, 1999.

[10] Kord Eickmeyer. Non-definability results for random first-order logic. In *Computer Science Logic*, September 2011.

[11] R. Fagin. Generalized first-order spectra and polynomial-time recognizable sets. In Richard M. Karp, editor, *Complexity of Computation*, volume 7 of *SIAM-AMS Proceedings*, pages 43–73, 1974.

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

[13] W. Feller. *An Introduction to Probability Theory and Its Aplications*, volume I. John Wiley & Sons, 1957.

[14] Y.V. Glebskiĭ, D.I. Kogan, M.I. Liogon'kiĭ, and V.A. Talanov. Range and degree of realizability of formulas in the restricted predicate calculus. *Kibernetika*, 2:17–28, 1969. Englisch translation, Cybernetics 5:142–154,1969.

[15] E. Grädel, P.G. Kolaitis, L. Libkin, M. Marx, J. Spencer, M.Y. Vardi, Y. Venema, and S. Weinstein. *Finite Model Theory and Its Applications*. Texts in Theoretical Computer Science. Springer-Verlag, 2007.

[16] Y. Gurevich. Logic and the challenge of computer science. In E. Börger, editor, *Current trends in theoretical computer science*, pages 1–57. Computer Science Press, 1988.

[17] L. Hella, P.G. Kolaitis, and K. Luosto. Almost everywhere equivalence of logics in finite model theory. *The Bulletin of Symbolic Logic*, 2(4):422–443, December 1996.

[18] N. Immerman. Relational queries computable in polynomial time. *Information and Control*, 68:86–104, 1986.

[19] N. Immerman. *Descriptive Complexity Theory*. Graduate Texts in Computer Science. Springer-Verlag, 1999.

[20] R. Impagliazzo and A. Wigderson. P = BPP if E requires exponential circuits: Derandomizing the XOR lemma. In *Proceedings of the 29th ACM Symposium on Theory of Computing*, pages 220–229, 1997.

[21] Russell Impagliazzo. Can every randomized algorithm be derandomized? In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, STOC '06, pages 373–374, 2006.

[22] P. Kaye. A logical characterisation of the computational complexity class BPP. Technical report, University of Waterloo, 2002.

[23] P. G. Kolaitis and M. Y. Vardi. Infinitary logics and 0-1 laws. *Information and Computation*, 98:258–294, 1992.

[24] C. Lautemann. BPP and the polynomial hierarchy. *Information Processing Letters*, 17(4):215–217, 1983.

[25] L. Libkin. *Elements of Finite Model Theory*. Texts in Theoretical Computer Science. Spinger-Verlag, 2004.

[26] J.F. Lynch. On sets of relations definable by addition. *Journal of Symbolic Logic*, 47(3):659–668, 1982.

[27] J. A. Makowski. Algorithmic uses of the feferman-vaught theorem. *Annals of Pure and Applied Logic*, 126(1-3):159–213, April 2004.

[28] J.C. Mitchell, M. Mitchell, and A. Scedrov. A linguistic characterization of bounded oracle computation and probabilistic polynomial time. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, pages 725–733, 1998.

[29] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[30] M. Müller. Valiant-vazirani lemmata for various logics. *Electronic Colloquium on Computational Complexity (ECCC)*, 15(063), 2008.

[31] N. Nisan and A. Wigderson. Hardness vs randomness. *Journal of Computer and System Sciences*, 49:149–167, 1994.

[32] Noam Nisan. Pseudorandom bits for constant depth circuits. *Combinatorica*, 11(1):63–70, 1991.

[33] M. Otto. *Bounded Variable Logics and Counting*. Lecture Notes in Logic. Springer-Verlag, 1996.

[34] N. Rescher. Plurality quantification. *Journal of Symbolic Logic*, 27(3):373–374, 1962.

[35] Nicole Schweikardt. On the expressive power of monadic least fixed point logic. *Theor. Comput. Sci.*, 350(2-3):325–344, 2006.

[36] L. Stockmeyer. The polynomial hierarchy. *Theoretical Computer Science*, 3:1–22, 1977.

[37] S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 20(5):865–877, 1991.

[38] M.Y. Vardi. The complexity of relational query languages. In *Proceedings of the 14th ACM Symposium on Theory of Computing*, pages 137–146, 1982.

[39] Emanuele Viola. The complexity of constructing pseudorandom generators from hard functions. *Computational Complexity*, 13:147–188, 2004.