Electronic Colloquium on Computational Complexity, Report No. 56 (2010)

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
eickmeye,grohe@informatik.hu-berlin.de

Abstract. We study probabilistic complexity classes and questions of derandomisation from a logical point of view. For each logic L we introduce a new logic BPL, *bounded error probabilistic* L, which is defined from L in a similar way as the complexity class BPP, bounded error probabilistic polynomial time, is defined from P.
Our main focus lies on questions of derandomisation, and we prove that there is a query which is definable in 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. We prove similar results for ordered structures and structures with an addition relation, showing that certain uniform variants of 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 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 P vs. 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 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 P and BPP are indeed equal. This belief is supported by deep results due to Nisan and Wigderson [1] and Impagliazzo and Wigderson [2], which link the derandomisation question to the existence of one-way functions and to circuit lower bounds. 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.

ISSN 1433-8092

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 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 L we introduce a new logic BPL, *bounded error probabilistic* L, which is defined from L in a similar way as BPP is defined from P. The randomness is introduced to the logic by letting formulas of vocabulary τ speak about *random expansions* of τ-structures to a richer vocabulary $\tau \cup \rho$. We also introduce variants RL, co-RL with one-sided bounded error and PL with unbounded error, corresponding to other well known complexity classes, but focus on BPL in this conference paper.

Our main technical results are concerned with questions of derandomisation. We prove that there is a query that is definable in 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. Note that these results are in sharp contrast to the general belief that most standard complexity classes can be derandomised.

We then investigate whether BPFO can be derandomised on classes of structures with built-in relations, such as ordered structures. Behle and Lange [3] showed that the expressive power of FO on classes of ordered structures with certain predefined relation symbols corresponds to uniform subclasses of AC^0, the class of circuit families of bounded depth, unbounded fan-in and polynomial size. In fact, for a set $\mathcal{R}$ of relation symbols whose interpretation is prescribed (such as linear orders, addition and multiplication relations) they show that $\mathsf{FO}[\mathcal{R}]$ captures $\mathsf{FO}[\mathcal{R}]$-uniform AC^0. We show that on additive structures, BPFO can not be derandomised, and on ordered structures it is not even contained in MSO.

Arguably the most intensively studied uniformity condition on AC^0 is *dlogtime-uniform* AC^0, which corresponds to $\mathsf{FO}[+,\times]$ by Barrington et al. [4]. The question of whether dlogtime-uniform BPAC^0 can be derandomised is still open, but there is a conditional derandomisation by Viola [5]. There are less uniform variants of BPAC^0 that can be proved to be derandomisable by standard arguments in the style of Adleman [6]. We prove the more uniform $\mathsf{FO}[+]$-uniform AC^0 to be non-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 BPIFP+C, the probabilistic version of fixed-point logic with counting, captures the complexity class BPP, even on unordered structures. For ordered structures, this result is a direct consequence of the Immerman-Vardi Theorem [7, 8], and for arbitrary structures it follows from the observation that we can define a random order with high probability in 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 $\mathsf{P} = \mathsf{BPP}$. The caveat is that the logic $\mathsf{BPIFP{+}C}$ does not have an effective syntax and thus is not a "logic" according to Gurevich's [9] definition underlying the question for a logic that captures P. Nevertheless, we believe that $\mathsf{BPIFP{+}C}$ 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 [10] that gives a logical characterisation of BPP on ordered structures. Müller [11] 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* [12], 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 [13] give a logical characterisation of BPP by a higher-order typed programming language in this context.

Outside of descriptive complexity theory and finite model theory, probabilistic logics have received wide attention in mathematical logic and computer science, particularly in artificial intelligence and also in database theory. However, all this work has little in common with ours, both on a conceptual and technical level. A few pointers to the literature are [14–17].

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 τ of relation symbols of fixed arities. A *τ-structure* A consists of a 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 σ, τ be vocabularies with $\sigma \subseteq \tau$. Then the *σ-restriction* of a τ-structure B is the σ-structure $B|_\sigma$ with universe $V(B|_\sigma) := V(B)$ and relations $R(B|_\sigma) := R(B)$ for all $R \in \sigma$. A *τ-expansion* of a σ-structure A is a τ-structure B such that $B|_\sigma = A$. For every class $\mathcal{C}$ of structures, $\mathcal{C}[\tau]$ denotes the class of all τ-structures in $\mathcal{C}$. A *renaming* of a vocabulary τ is a bijective mapping r from τ to a vocabulary τ' 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 τ-structure then A^r is the τ'-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. Whenever any of these relations symbols appear in a vocabulary τ, we demand that they be interpreted by a linear order and ternary addition and multiplication relations, respectively, in all τ-structures. To be precise, let $[n]$ be the set $\{0, 1, \ldots, n\}$ for $n \geq 0$, and denote by $\mathcal{N}_n$ the $\{\leqslant, +, \times\}$-structure with

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

and demand $A|_{\{\leqslant,+,\times\}\cap\tau} \cong (\mathcal{N}_{|A|-1})|_{\{\leqslant,+,\times\}\cap\tau}$ for all τ-structures A. We call structures whose vocabulary contains any of these relation symbols *ordered*, *additive* and *multiplicative*, respectively.

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

2.2 Logics

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

(i) For all $\varphi \in \mathsf{L}[\tau]$ the global relation $\mathcal{Q}_\varphi^{\mathsf{L}[\tau]}$ is a τ-query.

(ii) If $\sigma \subseteq \tau$ then $\mathsf{L}[\sigma] \subseteq \mathsf{L}[\tau]$, and for all formulas $\varphi \in \mathsf{L}[\sigma]$ and all τ-structures A it holds that $\mathcal{Q}_\varphi^{\mathsf{L}[\sigma]}(A|_\sigma) = \mathcal{Q}_\varphi^{\mathsf{L}[\tau]}(A)$.

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

A query $\mathcal{Q}$ is *definable* in a logic L if there is an L-formula φ such that $\mathcal{Q} = \mathcal{Q}_\varphi$. 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 L_1 is *weaker* than a logic L_2 (we write

$\mathsf{L}_1 \leqq \mathsf{L}_2$) if every query definable in L_1 is also definable in L_2. Similarly, we define it for L_1 and L_2 to be *equivalent* (we write $\mathsf{L}_1 \equiv \mathsf{L}_2$) and for L_1 to be *strictly weaker* than L_2 (we write $\mathsf{L}_1 \lneqq \mathsf{L}_2$). The logics L_1 and L_2 are *incomparable* if neither $\mathsf{L}_1 \leqq \mathsf{L}_2$ nor $\mathsf{L}_2 \leqq \mathsf{L}_1$.

Remark 1. Our notion of logic is very minimalistic, usually logics are required to meet additional conditions (see [18] 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. [19]); 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* FO, *second-order logic* SO and its fragments Σ^1_k, *monadic second-order logic* MSO, *transitive closure logic* TC and its *deterministic* variant DTC, *least*, *inflationary*, and *partial fixed-point logic* LFP, IFP, and PFP, and *finite variable infinitary logic* $\mathsf{L}^\omega_{\infty\omega}$. For all these logics except LFP there are also *counting versions*, which we denote by $\mathsf{FO}{+}\mathsf{C}$, $\mathsf{TC}{+}\mathsf{C}$, ..., $\mathsf{PFP}{+}\mathsf{C}$ 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:

$$\begin{array}{ccccccccccc}
\mathsf{FO} & \lneqq & \mathsf{DTC} & \lneqq & \mathsf{TC} & \lneqq & \mathsf{LFP} & \equiv & \mathsf{IFP} & \lneqq & \mathsf{PFP} & \lneqq & \mathsf{L}^\omega_{\infty\omega} \\
\lneqq & & \lneqq & & \lneqq & & \lneqq & & \lneqq & & \lneqq & & \lneqq \\
\mathsf{FO}{+}\mathsf{C} & \lneqq & \mathsf{DTC}{+}\mathsf{C} & \lneqq & \mathsf{TC}{+}\mathsf{C} & \lneqq & & & \mathsf{IFP}{+}\mathsf{C} & \lneqq & \mathsf{PFP}{+}\mathsf{C} & \lneqq & \mathsf{C}^\omega_{\infty\omega}.
\end{array} \tag{1}$$

Furthermore, MSO is strictly stronger than FO and incomparable with all other logics displayed in (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 P and NP. Let us briefly review the class BPP, *bounded error probabilistic polynomial time*, and other probabilistic complexity classes: A language $L \subseteq \Sigma^*$ is in 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 α, β 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 BPP is the class of all problems that can be solved by a randomised polynomial time algorithm with bounded error probabilities. 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 co-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, 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 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 [20], the class P^{PP} contains the full polynomial hierarchy. By the Sipser-Gács Theorem (see [21]), 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 τ 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 τ structures we associate the language $L(\mathcal{C}) := \{s(B) \mid B \in \mathcal{C}\} \subseteq \{0,1\}^*$. For a Boolean τ-query $\mathcal{Q}$, let $\mathcal{Q}_\leqslant := \left\{B \in \mathcal{O}[\tau \cup \leqslant] \;\middle|\; B|_\tau \in \mathcal{Q}\right\}$ 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 K if the language $L(\mathcal{Q}_\leqslant)$ is contained in K. We say that a logic L *captures* K if for all Boolean queries $\mathcal{Q}$ it holds that $\mathcal{Q}$ is definable in L if and only if $\mathcal{Q}$ is decidable in K. We say that L is *contained* in K if all Boolean queries definable in L are decidable in K.

Remark 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 [22–25].

3 Randomised logics

Throughout this section, let τ and ρ be disjoint vocabularies. Relations over ρ will be "random", and we will reserve the letter R for relation symbols from ρ. We are interested in *random* $(\tau \cup \rho)$*-expansions* of τ-structures. For a τ-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 $\boldsymbol{a} \in V(A)^k$ with probability $1/2$ whether $\boldsymbol{a} \in R(X)$. 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 τ-structure A satisfies a sentence φ of vocabulary $\tau \cup \rho$ of some logic.

Definition 1. *Let* 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* τ*-structures* A *and all* $\boldsymbol{a} \in V(A)^k$ *it holds that*

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

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

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

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

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

It is easy to see that for every logic L and all α, β 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 shows that for reasonable L the strength of the logic $\mathsf{P}_{(\alpha,\beta]}\mathsf{L}$ does not depend on the exact choice of the parameters α, β. This justifies the arbitrary choice of the constants $1/3, 2/3$ in the definitions of RL and BPL.

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

We omit the straightforward proof.

Remark 3. For the rest of this conference paper, we focus entirely on logics BPL with two-sided bounded error. Many of our results have a version for logics RL with one-sided error as well. The logics PL are considerably more expressive and behave quite differently. For example, PFO contains the existential fragment Σ^1_1 of second-order logic. on all structures with at least one definable element (like the minimal element of a linear order). More results about the logics RL and PL will appear in the full version of this paper.

Remark 4. As we mentioned several times, our motivation for this work is to study complexity theoretic questions related to randomisation (and derandomisation) in a logical context. Thus we designed our logics to be faithful images of probabilistic complexity classes. Furthermore, we focus on "complexity theoretic" questions, ignoring natural "logical" questions such as which closure properties our logics have. This is not meant to say that these questions are not worthwhile to be studied.

From a logical point of view, it may seem more natural to work with probabilistic quantifiers, but we believe that this leads to somewhat different questions than those we are interested in here. Our logics seem to be very well-suited to study probabilistc complexity classes, questions of derandomisation, pseudorandom generators, and other topics arising in this context. We believe that our results, in particular those stating that strongly uniform variants of AC^0 cannot be randomised, fully justify our descriptive theoretic approach to these topics.

3.1 First observations

We start by observing that the syntax of BPFO and thus of most other logics BPL is undecidable. This follows easily from Trakhtenbrot's Theorem (see [22] for similar undecidability proofs):

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

For each n, let S_n be the $\emptyset$-structure with universe $V(S_n) := \{1, \dots, n\}$. Recall the 0-1-law for first order logic [26, 27]. In our terminology, it says that for each vocabulary ρ 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, BPFO has the same expressive power as FO. As there is also a 0-1-law for the logic $\mathsf{L}^{\omega}_{\infty\omega}$ [28], we actually get the following stronger statement:

Observation 2. *Every formula $\varphi \in \mathsf{BPL}^{\omega}_{\infty\omega}[\emptyset]$ is equivalent to a formula $\varphi' \in \mathsf{FO}[\emptyset]$.*

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

Finally, we note that the Sipser-Gács theorem [21] that $\mathsf{BPP} \subseteq \Sigma^p_2 \cap \Pi^p_2$, the fact that the fragment Σ^1_2 of second-order logic captures Σ^p_2 [29, 30], and the observation that $\mathsf{BPFO} \leqq \mathsf{BPP}$ imply the following:

Observation 3. $\mathsf{BPFO} \leqq \Sigma^1_2$.

4 Separation results for BPFO

In this section we study the expressive power of the randomised logic BPFO. Our main results are the following:

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

It turns out that we need three rather different queries to get these separation results. For the first two queries this is immediate by the fact that *any* query on ordered structures is axiomatisable in $\mathsf{C}^{\omega}_{\infty\omega}$. The third query (on additive structures) is readily seen to be axiomatisable in MSO.

In fact, any BPFO-axiomatisable query on additive structures can be axiomatised in MSO. To see this, we first use Nisan's pseudorandom generator for constant depth circuits [31] to reduce the number of random bits to $m := \mathrm{polylog}(n)$, where n is the size of the input structure. We then use an expander random walk to generate $\sqrt{n}$ many blocks of m pseudorandom bits each, using a seed of only $s := m + O(\sqrt{n})$ bits, see [32]. Taking a majority vote over the $\sqrt{n}$ many trials, for large enough n, the error drops down to below $2^{-s/3}$, and we use an argument similar to that by Goldreich and Zuckerman [33]. Note that these pseudorandom generators are expressible in MSO on additive structures, essentially because we can quantify over binary relations on the first $\sqrt{n}$ numbers of the structure. Details of this proof will appear in the full version of this paper.

4.1 BPFO 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 φ"). For example, the class of finite structures of even cardinality can be axiomatised 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 1. *There is a class $\mathcal{TCFI}$ of structures that is definable in* BPFO*, but not in $\mathsf{C}^{\omega}_{\infty\omega}$.*

Recall that by Observation 2 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 1 is based on a well-known construction due to Cai, Fürer, and Immerman [34], 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 proofs by Dawar, Hella, and Kolaitis [35] for results on implicit definability in first-order logic, and obtain a query $\mathcal{TCFI}$ definable in BPFO, 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 axiomatise 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. For details we refer to Appendix A.

4.2 BPFO on ordered structures is not contained in MSO

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

Theorem 2. *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:
$\mathcal{B}'$ 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

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

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 the following lemma:

Lemma 2 (Birthday Paradoxon). *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*
$$\Pr(F \text{ is injective}) \leq \epsilon_1$$
2. *For any $\epsilon_2 > 0$, if $m \geq \frac{n^2}{2\epsilon_2}$, then*
$$\Pr(F \text{ is injective}) \geq 1 - \epsilon_2$$

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

$$\Pr(F \text{ not injective}) = \Pr\left(\bigcup\nolimits_{1 \leq i < j \leq n} \{F(i) = F(j)\}\right) \leq \sum_{i<j} \frac{1}{m} \leq \frac{n^2}{2m}.$$

□

Proof (Theorem 2). 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. Suppose that we have a linearly ordered structure of the form $A \cup B$, and the two parts A, B are disjoint and not related except by the linear order, which puts A completely before B. Let $q \geq 0$ and A' another linearly ordered structure that is indistinguishable from A by all MSO-sentences of quantifier rank at most q. Then the structure $A' \cup B$ is indistinguishable from $A \cup B$ by all MSO-sentences of quantifier rank at most q. 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 $A \in \mathcal{K}$ 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, n in such a way that $A \notin \mathcal{K}'$ and $A' \in \mathcal{K}'$.

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

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

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

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

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 $\mathcal{K}$, 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 φ_{inj} in all sufficiently large structures in $\mathcal{K}$, by lemma 2 with $\epsilon_1 = 0.2$, $\epsilon_2 = 0.5$ and $c = 4$. The remaining finitely many structures in $\mathcal{K}$ can be dealt with separately. □

4.3 BPFO 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 $([|A| - 1], \{(a, b, c) \mid a + b = c\})$.

Theorem 3. *There is a class $\mathcal{A}$ of additive structures that is definable in* BPFO*, but not in* FO*.*

Our proof uses the following result:

Theorem 4 (Lynch [36]). *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* 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 1. *Let k, A_k, d_k, Q_0 and Q_1 be as above. Then the (finite) structures $([\max Q_0], +, Q_0)$ and $([\max Q_1], +, Q_1)$ satisfy exactly the same* FO*-sentences of quantifier rank at most k.*

We call a set $Q \subset \mathbb{N}$ *sparse* if $|Q \cap \{n, \dots, 3n\}| \leq 1$ 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 φ_{sparse} such that

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

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

Proof (Proof of Theorem 3). We define the following class of additive $\{+, P\}$-structures:

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

with $+$ defined as usual. It follows immediately from Corollary 1 that $\mathcal{A}$ is not definable in FO.

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

Each element $a \in [\max Q]$ 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., [37]; or use a union-bound argument), if $\max Q$ is large enough, every subset of Q is defined by some element of $[\max Q]$. Thus we may quantify over subsets of Q. Since we can define a linear order on the structure $\mathcal{Q}$ from the addition, we can now easily express evenness of Q in FO. □

5 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 [12].

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 [38] 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 [39], see [38].

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 $\varphi_1(\boldsymbol{x})$ and $\varphi_2(\boldsymbol{x})$, where $\boldsymbol{x}$ is a k-tuple of variables, we form a new formula

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

Its semantics is defined by

$$A \models \mathcal{J}\boldsymbol{x}.\varphi_1(\boldsymbol{x})\varphi_2(\boldsymbol{x}) \quad \text{iff} \\ \left|\{\boldsymbol{a} \in V(A)^k \mid A \models \varphi_1[\boldsymbol{a}]\}\right| \leq \left|\{\boldsymbol{a} \in V(A)^k \mid A \models \varphi_2[\boldsymbol{a}]\}\right|. \quad (3)$$

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

Lemma 3. *Let R be a 6-ary relation symbol. There is a formula $\varphi_\leq(x,y) \in \mathsf{FO}(\mathcal{J})[\{R\}]$ such that*

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

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

A proof of this lemma can be found in Appendix B.

Theorem 5. *The logic $\mathsf{BPIFP}(\mathcal{J})$ captures BPP.*

Proof. $\mathsf{BPIFP}(\mathcal{J})$ is contained in 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 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 BPIFP-sentence ψ_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 ψ_M by the formula $\varphi_\leq(x,y)$ of Lemma 3, which with high probability defines a linear order on the universe. □

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

Corollary 2. $\mathsf{BPIFP{+}C} = \mathsf{BPIFP}(\mathcal{J})$*, and both capture* BPP.

Remark 5. By similar arguments, we obtain logical characterisations of other randomised complexity classes. For example, $\mathsf{BPL} = \mathsf{BPDTC}(\mathcal{J}) = \mathsf{BPDTC{+}C}$. (Here L does not denote a generic logic, but the complexity class logspace.)

Furthermore, it also follows from Lemma 3 that $\mathsf{BPL}^\omega_{\infty\omega}(\mathcal{J}) = \mathsf{BPC}^\omega_{\infty\omega}$. Actually, it follows that all queries are definable in $\mathsf{BPL}^\omega_{\infty\omega}(\mathcal{J})$.

Acknowledgements

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

References

1. Nisan, N., Wigderson, A.: Hardness vs randomness. Journal of Computer and System Sciences **49** (1994) 149–167
2. Impagliazzo, R., Wigderson, A.: P = BPP if E requires exponential circuits: Derandomizing the xor lemma. In: Proceedings of the 29th ACM Symposium on Theory of Computing. (1997) 220–229
3. Behle, C., Lange, K.J.: FO[<]-uniformity. In: IEEE Conference on Computational Complexity. (2006) 183–189
4. Barrington, D.A.M., Immerman, N., Straubing, H.: On uniformity within NC^1. J. Comput. Syst. Sci. **41**(3) (1990) 274–306
5. Viola, E.: The complexity of constructing pseudorandom generators from hard functions. Electronic Colloquium on Computational Complexity (ECCC) (020) (2004)
6. Adleman, L.M.: Two theorems on random polynomial time. In: FOCS. (1978) 75–83
7. Immerman, N.: Relational queries computable in polynomial time. Information and Control **68** (1986) 86–104
8. Vardi, M.: The complexity of relational query languages. In: Proceedings of the 14th ACM Symposium on Theory of Computing. (1982) 137–146
9. Gurevich, Y.: Logic and the challenge of computer science. In Börger, E., ed.: Current trends in theoretical computer science. Computer Science Press (1988) 1–57
10. Kaye, P.: A logical characterisation of the computational complexity class BPP. Technical report, University of Waterloo (2002)
11. Müller, M.: Valiant-vazirani lemmata for various logics. Electronic Colloquium on Computational Complexity (ECCC) **15**(063) (2008)
12. Hella, L., Kolaitis, P., Luosto, K.: Almost everywhere equivalence of logics in finite model theory. The Bulletin of Symbolic Logic **2**(4) (December 1996) 422–443
13. Mitchell, J., Mitchell, M., Scedrov, A.: A linguistic characterization of bounded oracle computation and probabilistic polynomial time. In: Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science. (1998) 725–733
14. Bacchus, F.: Representing and Reasoning with Probabilistic Knowledge. MIT Press (1990)
15. Dalvi, N., Ré, C., Suciu, D.: Probabilistic databases: diamonds in the dirt. Communications of the ACM **52**(7) (2009) 86–94
16. Fagin, R., Halpern, J., Megiddo, N.: A logic for reasoning about probabilities. Information and Computation **87**(1/2) (1990) 78–128
17. Keisler, H.: Probability quantifiers. In Barwise, J., Feferman, S., eds.: Model–Theoretic Logics. Springer-Verlag (1985) 509–556
18. Ebbinghaus, H.D.: Extended logics: The general framework. In Barwise, J., Feferman, S., eds.: Model–Theoretic Logics. Springer-Verlag (1985) 25–76
19. Barwise, J., Feferman, S., eds.: Model Theoretic Logics. Perspectives in Mathematical Logic. Springer-Verlage (1985)
20. Toda, S.: PP is as hard as the polynomial-time hierarchy. SIAM Journal on Computing **20**(5) (1991) 865–877

21. Lautemann, C.: BPP and the polynomial hierarchy. Information Processing Letters **17**(4) (1983) 215–217
22. Ebbinghaus, H.D., Flum, J.: Finite Model Theory. 2nd edn. Perspectives in Mathematical Logic. Springer-Verlag (1999)
23. Grädel, E., Kolaitis, P., Libkin, L., Marx, M., Spencer, J., Vardi, M., Venema, Y., Weinstein, S.: Finite Model Theory and Its Applications. Texts in Theoretical Computer Science. Springer-Verlag (2007)
24. Immerman, N.: Descriptive Complexity Theory. Graduate Texts in Computer Science. Springer-Verlag (1999)
25. Libkin, L.: Elements of Finite Model Theory. Texts in Theoretical Computer Science. Spinger-Verlag (2004)
26. Fagin, R.: Probabilities on finite models. Journal of Symbolic Logic **41** (1976) 50–58
27. Glebskiĭ, Y., Kogan, D., Liogon'kiĭ, M., Talanov, V.: Range and degree of realizability of formulas in the restricted predicate calculus. Kibernetika **2** (1969) 17–28 Englisch translation, Cybernetics 5:142–154,1969.
28. Kolaitis, P.G., Vardi, M.Y.: Infinitary logic and 0-1 laws. Information and Computation **98** (1992) 258–294
29. Fagin, R.: Generalized first-order spectra and polynomial-time recognizable sets. In Karp, R.M., ed.: Complexity of Computation. Volume 7 of SIAM-AMS Proceedings. (1974) 43–73
30. Stockmeyer, L.: The polynomial hierarchy. Theoretical Computer Science **3** (1977) 1–22
31. Nisan, N.: Pseudorandom bits for constant depth circuits. Combinatorica **11**(1) (1991) 63–70
32. Zuckerman, D.: Simulating bpp using a general weak random source. Algorithmica **16**(4/5) (1996) 367–391
33. Goldreich, O., Zuckerman, D.: Another proof that bpp subseteq ph (and more). Electronic Colloquium on Computational Complexity (ECCC) **4**(45) (1997)
34. Cai, J.Y., Fürer, M., Immerman, N.: An optimal lower bound on the number of variables for graph identifications. Combinatorica **12**(4) (1992) 389–410
35. Dawar, A., Hella, L., Kolaitis, P.G.: Implicit definability and infinitary logic in finite model theory. In: ICALP. Volume 944 of LNCS., Springer Verlag (1995) 624–635
36. Lynch, J.: On sets of relations definable by addition. Journal of Symbolic Logic **47**(3) (1982) 659–668
37. Motwani, R., Raghavan, P.: Randomized Algorithms. Cambridge University Press (1995)
38. Otto, M.: Bounded Variable Logics and Counting. Lecture Notes in Logic. Springer-Verlag (1996)
39. Rescher, N.: Plurality quantification. Journal of Symbolic Logic **27**(3) (1962) 373–374
40. Feller, W.: An Introduction to Probability Theory and Its Aplications. Volume I. John Wiley & Sons (1957)
41. Babai, L., Erdös, P., Selkow, S.: Random graph isomorphism. SIAM Journal on Computing **9**(3) (1980) 628–635

A The query $\mathcal{TCFI}$

We first review the construction of [34] 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.

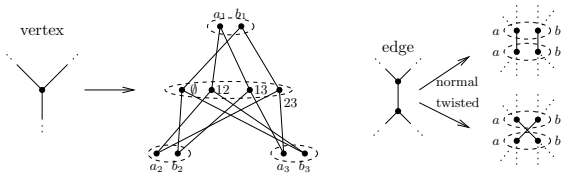

Fig. 1. The gadgets for CFI-graphs. Dashed ellipses indicate groups of equivalent vertices. Vertex labels are not part of the actual structure.

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 $\{1, 2, 3\}$. Permuting the pairs of nodes results in a permutation of the four centre nodes, so we get isomorphic graphs regardless of which way we order the pairs. 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 the centre nodes and the nodes in each pair.

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.

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 graphs, 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.

Now, for every $\mathsf{C}^\omega_{\infty\omega}$-sentence φ, if the original graph G is complicated enough, the two isomorphism classes can not be told apart by φ [34]. In 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

$$\text{(no. of odd centre groups} + \text{no. of twisted edgets) is odd.}$$

We aim for a BPFO-sentence which defines exactly the twisted connected 3-regular CFI-graphs. In view of the above P-algorithm, we are done if we can

- express connectedness of the graph,
- count 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:[1] Let τ 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 and the nodes of one edget or one centre group, or consists of a single non-atom of $\mathfrak{B}$.

Theorem 6. *The class $\mathcal{CFI}$ is definable in FO. The subclass $\mathcal{TCFI}$ of twisted CFI-graphs is definable in 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 [34]; 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 BPFO. We pick one vertex from each edge group by viewing a random binary relation R as assigning

[1] It has been pointed out to us that a somewhat similar construction appears in [35], but there the starting point is a linear order rather than a CFI-graph.

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 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 ξ succeeds in picking exactly one vertex from each edge group, and 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 close to one. □

B Proof of lemma 3

Proof (of lemma 3). We let

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

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

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

Then

$$A \models \varphi_{\leq}(a, b) \quad \text{iff} \quad X_a \leq X_b,$$

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

$$\begin{aligned}
\Pr(X_a = X_b) &= \sum_{k=0}^{m} \left(\frac{1}{2^m}\binom{m}{k}\right)^2 \\
&= \frac{1}{2^{2m}} \sum \binom{m}{k}^2 = \frac{1}{2^{2m}} \sum \binom{m}{k}\binom{m}{m-k} \\
&= \frac{1}{2^{2m}}\binom{2m}{m} = \Theta\left(\frac{1}{\sqrt{m}}\right),
\end{aligned}$$

where the final approximation can be found, for example, in [40]. The second part now follows by a union bound over the $\binom{n}{2} = \Theta(m^{2/5})$ pairs $a \neq b$. □

Remark 6. While using a 6-ary relation makes the above analysis of the success probability particularly simple, in IFP it is also possible to define an order with high probability using a binary random relation and Rescher quantifier [41] or a binary random relation and an even quantifier [12].

ECCC ISSN 1433-8092

http://eccc.hpi-web.de