Classifying the computational power of stochastic physical oracles

EDWIN BEGGS[1⋆], PEDRO CORTEZ[2], JOSÉ FÉLIX COSTA[2,3†], JOHN V. TUCKER[1‡]

[1] *College of Science, Swansea University, Singleton Park, Swansea, SA2 8PP, U.K.*
[2] *Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal*
[3] *Centro de Filosofia das Ciências da Universidade de Lisboa, Lisboa, Portugal*

Consider a computability and complexity theory in which the classical set-theoretic oracle to a Turing machine is replaced by a physical process, and oracle queries return measurements of physical behaviour. The idea of such physical oracles is relevant to many disparate situations, but research has focussed on physical oracles that were classic deterministic experiments which measure physical quantities. In this paper, we broaden the scope of the theory of physical oracles by tackling non-deterministic systems. We examine examples of three types of non-determinism, namely systems that are: (1) physically nondeterministic, as in quantum phenomena; (2) physically deterministic but whose physical theory is non-deterministic, as in statistical mechanics; and (3) physically deterministic but whose computational theory is non-deterministic caused by error margins. Physical oracles that have probabilistic theories we call *stochastic physical oracles*. We propose a set SPO of axioms for a basic form of stochastic oracles. We prove that Turing machines equipped with a physical oracle satisfying the axioms SPO compute precisely the non-uniform complexity class $BPP//\log\star$ in polynomial time. This result of $BPP//\log\star$ is a computational limit to a great range of classical and non-classical measurement, and of analogue-digital computation in polynomial time under general conditions.

Key words: Physical oracles; Stochastic oracles; Non-uniform complexity

⋆ email: E.J.Beggs@Swansea.ac.uk
† email: fgc@math.tecnico.ulisboa.pt
‡ email: J.V.Tucker@swansea.ac.uk

1 INTRODUCTION

Consider an algorithm that in the course of its computations requests information from an external source. In the theory of computability, the external source is called an oracle. The idea of boosting the power of algorithms – and the terminology of 'oracle' – originates with Alan Turing [1]. He stipulated that an oracle was not a (Turing) machine and used oracles to decide the truth of propositions about natural numbers. Its usefulness in computability theory was demonstrated by Emil Post who studied the computability of sets relative to other sets, and began to develop theories of degrees of unsolvability in [2]. Oracles have had a profound influence on our understanding of computability and complexity since they can be used to classify into hierarchies all sorts of problems that are unsolvable (e.g., Turing degrees [3]) or are unsolvable with feasible resources (e.g., complexity classes and hierarchies [4]). Generalisations and applications of relative computability and complexity have grown; relativised models of computation are so common as to be considered standard.

Physical oracles: However, suppose the external source consulted by an algorithm is not a pure mathematical entity but a physical device, system or environment. Suppose the requests for information are requests to observe physical behaviour and make measurements of physical quantities. We call this external source a *physical oracle*. Situations abound where algorithms with physical oracles may be found, including:

(i) monitoring and controlling devices and machines, such as spatially-aware phones and autonomous cars;

(ii) experiments to measure physical phenomena, such as the half-life of an element;

(iii) enhancing algorithms using physical devices, such as application specific processors, random generators, clocks and biosensors;

(iv) training algorithms to recognise images and behaviour patterns, such as in machine learning with neural nets.

Clearly, the conception of algorithms with physical oracles is immensely broad and relates to established topics of study such as analogue-digital systems, hybrid systems, analogue computers, neural nets, etc.

Since there are infinitely many physical oracles, how can we answer the questions, *What is the computational power of adding a physical oracle? How does the computational theory depend upon the many and varied physical theories and models?*

Starting in [5, 6], we began a theoretical investigation of physical oracles,

focussing on their role in experiments (ii). We examined what was involved in an algorithm requesting and receiving data from a physical process, and especially interface properties to do with

(a) the error margins involved in the data, and

(b) the time taken by the algorithm to acquire the data.

Three types of error margins were analysed: *exact precision*, *unbounded precision*, and *fixed precision*. We also placed complexity constraints on the computations, especially *polynomial time*.

Our method was to study a variety of classical kinematic, electrical, optical and atomic experiments in some detail and, in each case, establish methods and theorems that classified the computational power of Turing machines equipped with these experiments as physical oracles. The computational classification needed non-uniform complexity classes [7], and a number of these complexity classes arose, depending upon the interface properties to do with precision (a) and timing (b). However, two proved to be particularly common: the non-uniform classes $P/\log\star$ and $BPP//\log\star$ (see Section 44.1). After many case studies, we were able to address the questions above by formulating axioms that expressed properties common to a wide class of disparate physical systems with independent physical theories [8]. We could show that whilst the physical oracles satisfying the axioms computed the same sets, they also broke the so called Turing Barrier, i.e., algorithms with physical oracles – and constrained by polynomial time – could decide the halting problem for Turing machines.

One purpose of our programme is create a theory that can combine algorithmic computation with physical processes via their physical theories. The basic architecture of an algorithm with a physical oracle is depicted in Figure 1, in which arrows are used to indicate the direction of flow of information. An important point is to see the physical system as a component distinct from its physical theory. This is natural when one recalls that in modelling a physical system there can be many assumptions that can be chosen and that need to be evaluated with regard to physical accuracy or computational complexity. Even in kinematic systems there are simple modelling choices that lead to very different theoretical outcomes [9]. The physical theory influences the design of the measurements, interface and algorithm. In each case, the experiments studied – and which were the target of the first axiomatisations [8] – were elementary and had classical explanations that needed only simple fragments of physical theories. Keeping examples simple meant that we could focus on the new ideas without being distracted by debates about physical modelling. In particular, the experiments were deterministic.

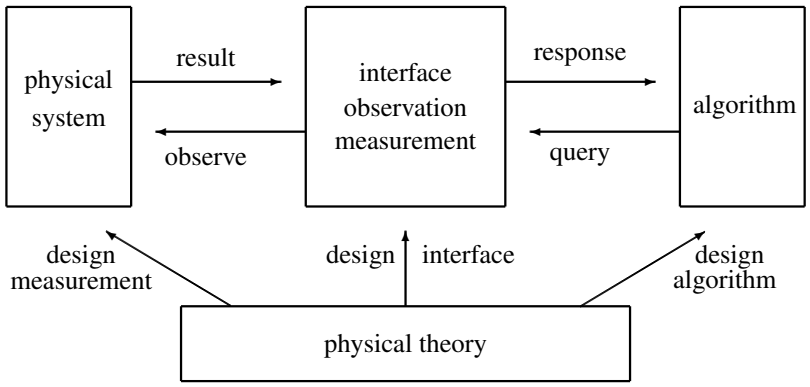

FIGURE 1
An analogue-digital system.

Stochastic oracles: In this paper we analyse physical oracles that can be more complicated and so more realistic or useful. The physical systems will be non-deterministic and require probability theory to model and reason about their behaviour. These we call *stochastic physical oracles*. The nature of non-determinism in physical systems is intriguing and there is a need for a taxonomy of the causes of the non-determinism. The new physical oracles we wish to introduce are physical systems that are:

1. *physically nondeterministic*, such as devices to measure half-life in the atomic decay of an element, and other quantum phenomena;

2. *physically deterministic, but whose physical theory is non-deterministic*, such as brownian motion and thermodynamic phenomena that use statistical methods because of their scale and complexity;

3. *physically deterministic, but whose computational theory is non-deterministic* because of unknowns arising from error margins.

See Figure 2. We encountered the use of probabilities in the case of 3. when dealing with fixed error margins for classical experiments see, e.g., [8].

The stochastic physical oracles greatly extend the role of the theory in the case of experiments (ii), but also are needed to apply the theory to the case of control (i) and learning (iv). Interestingly, independent studies of neural networks had come across non-uniform complexity classes – see subsection 9.2.

Thus, we consider physical systems whose behaviour is modelled probabilistically and depends upon a physical quantity or parameter. Following an examination of examples of non-deterministic physical systems illustrating the three categories 1 - 3, we formulate a set SPO of axioms for a class

of physical systems to qualify as stochastic oracles. We prove the following theorem (Corollary 7.1), which follows from Theorems 5.2 and 7.1:

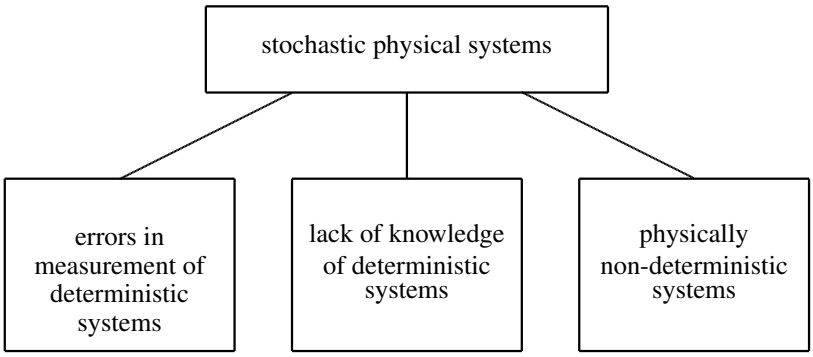

FIGURE 2
Types of stochastic physical oracle.

Theorem. *Let SPO be the axioms for stochastic physical oracles. Let P be a physical system whose behaviour depends upon a physical quantity or parameter σ. Suppose P satisfies the axioms of SPO. Then: a set $A \subset \{0,1\}^\star$ is decidable in polynomial time by a Turing machine with physical oracle P and an unknown parameter σ if, and only if, $A \in BPP//\log\star$.*

Naturally, $BPP//\log\star$ is an established computational entity. Thus, for completeness, we generalize the fair probabilistic Turing machine of BPP to the case of probability of transition $p \in (0,1)$ and prove the following statement (Theorem 8.3):

Theorem. *The class of sets decidable in polynomial time by p-probabilistic Turing machines with bounded error probability is exactly $BPP//\log\star$.*

In Section 2, we introduce the three types of non-deterministic physical process and illustrate them with experiments with atomic decay, brownian motion and a beam balance to measure mass. In Section 3, we formulate the axioms SPO and show the examples in Section 2 satisfy the axioms; for contrast we also consider some physical features that are not covered by the

axioms. Some technical preliminaries are the subject of Section 4; a full account of the concepts of structural complexity that we will be using can be found in [7, 10]. Lower and upper bounds are studied in Sections 5 and 7, respectively. Technical aspects of computation trees, needed for upper bounds, are discussed in Section 6. Finally, in Section 8, we apply our methods to a form of probabilistic Turing machine and end with concluding remarks in Section 9.

2 MEASUREMENT BASED ON PROBABILITIES

We look at some examples where we make measurements of a physical parameter. The class of physical systems we are theorising have two basic properties: observations result in two valued data, and queries give probabilistic results which are independent and identically distributed. The probability of a given result is defined by the physical parameter.

2.1 Radioactive decay

Radioactive decay is a phenomenon whose theory asserts that no measurement made on an atom can alter its decay. Radioactive decay is a classic example of an independent random process. Consider a radioactive isotope with a half-life H, which is the physical quantity we intend to measure. Imagine a single atom is observed for a fixed time T, and we record if it has decayed in this time interval $[0, T]$ (returning 'yes') or not (returning 'no'). The probability of 'yes' is

$$P[\text{yes}] = 1 - \exp(-\log(2)\, T/H)\ . \tag{1}$$

If we have enough observations then we can find error bounds (or *confidence intervals*) for the half-life H. We return to radioactive decay in Section 3.3.

2.2 Random walk with an absorbing surface

A number of physical systems can be modelled or approximated by random walks. Consider a random walk of a particle on 1-dimensional integer lattice. The particle begins at position $x = 1$ at time $t = 0$. Then, at each unit time interval, the particle moves right, with probability σ, or left, with probability $1 - \sigma$, as outlined in Figure 3.

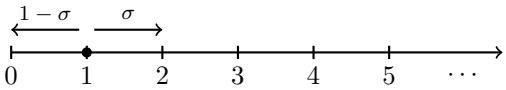

FIGURE 3
Random walk on the line with absorption at $x = 0$.

We would like to measure the probability σ. To do this, we choose to set up an absorbing barrier at $x = 0$, so that the first time the particle reaches $x = 0$ it sticks there, and we record that the particle has been absorbed (see [11]). In time $T = 1$ the probability that the particle has been absorbed is $1 - \sigma$, and in time $T = 3$ the probability of absorption is $1 - \sigma + (1 - \sigma)^2\sigma$. For larger times, we must count the possible paths of a given length from 1 to 1 which do not go through 0. To reach $x = 0$ for the first time at time $n + 1$ the particle must have reached $x = 1$ at time n not via $x = 0$. The cardinality of this set is that of a set of allowable bracketings, where a movement to the right is represented by "(" and a movement to the left is represented by ")". Thus the paths of length 6 from 1 to 1 which do not go through 0 can be denoted by

$$\{((())), (()()), (())(), ()((())), ()()().\}$$

Following this shows that the probability of absorption in time T is given by

$$F(\sigma) = \sum_{\substack{t=1 \\ t\ odd}}^{T} \frac{1}{t} \binom{t}{\frac{t+1}{2}} (1-\sigma)^{\frac{t+1}{2}} \sigma^{\frac{t+1}{2}-1} . \tag{2}$$

This finite step size random walk is used as an approximation for Brownian motion. As the random motions referred to by Titus Lucretius Carus [12] in the 1st Cent. B.C. are caused by air currents, the first proper descriptions of motion of larger particles caused by random motion of atoms was by the botanist Robert Brown [13] and the biologist Jan Ingenhousz, and the detailed theory was due to Einstein [14]. We may imagine that our analysis is a mathematical simplification of observing Brownian motion in natur

2.3 Monte Carlo methods

The theories of many physical systems involving quantum fields have results expressed as integrals over high dimensional spaces. Often, the best way to estimate these integrals is to use Monte Carlo integration [15]. Effectively, the values of the integrals are to be measured experimentally. (Strictly, such systems will be an example of our current axioms only when integrating a

characteristic function of a subset, requiring only a 0 or 1 response.) There are advantages in using physical random number generators (see the article by Tony Warnock in [15]) rather than algorithmically generated ones (the potential problems here are studied in [16]), and there has been recent interest in the theory and practicality of physical random number generators [17].

Unusual examples of experiments whose theory contains a Monte Carlo method include probabilistic experimental calculations for finding π, which have been promoted in schools as part of *Pi day* (March 14th). These include dropping rods onto parallel lines and seeing how many cross the two parallel lines, or making random dots on a square with a circle inscribed, and counting how many dots are inside the circle.

2.4 Experimental error

In any experiment that has error margins, we have an experiment capable of producing different results when repeated. Suppose the result of an experiment that has error margins is an event that can be classified as occurring or not occurring, i.e., 'yes' and 'no' for short. If these results are independent and identically distributed, we could use the methods in this paper to estimate the probabilities of 'yes' and 'no' from a sequence of experimental results. However, to go further and link this error probability to some underlying physical parameter, the theory of the experiment must describe the resulting probability distribution of 'yes' and 'no' as a function F of the parameter σ.

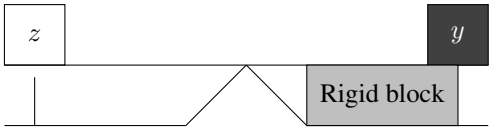

FIGURE 4
The broken balance experiment.

As a concrete example, we consider a simple experiment with an error margin that has one of two events when observed. The broken balance experiment (BBE for short) consists of a balance scale with two plates. In the right plate of the balance is placed a body with unknown mass y that we intend to measure. We place test masses z on the left plate of the balance. If $z < y$, then the plates do not move since the rigid block prevents the right plate from moving down. If $z > y$, then the left plate will move down. If $z = y$, the plates will not move. The scheme of this experiment is represented in Figure 4.

A pressure-sensitive stick is placed below the left plate of the balance such that, if the left plate moves down, it touches the pressure-sensitive stick and it reacts producing a signal. Several physical properties of the experiment are discussed in [18]. If the stick is triggered in fixed time T the result is 'yes', and otherwise 'no'. For every unknown mass y and test mass z, the experimental time of the BBE, i.e., the time between the placement of the mass z and the detection, is given, up to a constant factor, by the function:

$$T_{\exp}(z, y) = \begin{cases} \sqrt{\frac{z+y}{z-y}} & \text{if } z > y \\ \infty & \text{otherwise} \end{cases} .$$

There may be several sources of error in the experiment: for example, errors in the experimental time $T_{\exp}$ or inaccuracy in the pressure-sensitive stick. For a fixed test mass z this might result in identical experiments having non-identical outcomes. As mentioned previously, we could still measure the probability under assumption of independence and identically distributed results. Taking timing as the source of the error, if we had a theory of how the errors in the experimental time $T_{\exp}$ were distributed, we could then proceed to an experimental determination of the unknown mass y.

3 AXIOMS FOR STOCHASTIC PHYSICAL ORACLES

3.1 On nondeterministic physical oracles

Informally, an oracle to an algorithm has the capacity to answer questions generated by the algorithm; the answers may then be used in the algorithm's computations. Traditionally, after Post, oracles have been idealised as sets, oracle queries as set membership, and the queries answered in unit time. Clearly the idealisation can be changed replacing functions with sets and allowing the time to vary, e.g., query time being a function of the length of the question.

A physical oracle to an algorithm assumes that a physical process is involved in answering queries, more specifically that the query causes an observation or measurement of physical behaviour to be made, and the result returned to the algorithm. Queries may or may not contain data (initial conditions, parameters etc.). Physical oracles may be physical devices, experiments or environments.

A nondeterministic oracle can return a number of possible answers for a single query. This can lead to nondeterministic computation where the oracle answers are utilised by the algorithm. A physical oracle can be nondetermin-

istic, in fact given the errors involved in the measurement process this is very likely. The most obvious cases of nondeterminism are:

1) A single answer is returned for each query, but the answer is not uniquely defined by the query, i.e., identical queries may give different answers at different times.

2) A number of possible answers are returned to a single query; indeed the number may change each time a query is made.

In principle, the answer given by a nondeterministic oracle may depend on some complicated series of calculations or observations about which we have no idea. We can dramatically simplify this situation by restricting to *stochastic physical oracles*, by which we mean nondeterministic physical oracles in which the answers to a query obey well defined probabilistic rules. Observing the non-deterministic behaviour of a physical process may not be dependent on a physical theory; the emergence of probabilities is dependent on a choice of physical theory about the process.

3.2 On stochastic physical oracles

Here we specialise the idea of nondeterministic oracles to a fundamental case where we can analyse their behaviour in some detail. Imagine a deterministic Turing machine coupled with an oracle that is a physical system whose behaviour is non-deterministic. We suppose the behaviour is described by probabilities, and that queries to this physical oracle return an answer that may be 'yes' or 'no'. Each query triggers an independent identical experiment, which proceeds in a fixed time to the two possible outcomes; the oracle is not passed any parameters in queries. These outcomes are then independent identically distributed random variables. This means that each outcome is not dependent on the answers to previous queries, or even the number of previous queries.

Further, suppose the probability of the outcomes is determined by a single physical quantity or parameter σ in the physical system.* For convenience let $\sigma \in [0,1]$, as it is easy to renormalise another closed bounded interval with rational end points to $[0,1]$ by changing variables. The difficulties of taking an unbounded range for the parameter will be illustrated by proton decay in Section 3.4.

We suppose that, according to some physical theory PT, the probability of 'yes' is given in terms of a physical parameter $\sigma \in [0,1]$ by a function $F(\sigma)$. Given $F : [0,1] \to [0,1]$, we use data from repeated consultations of the

* We have taken one parameter for simplicity, in principal this could be generalised.

oracle to measure the physical parameter σ. Now we can give the axioms for a basic type of stochastic physical oracle for measurement (SPO for short).

Definition 3.1. *An SPO consists of a physical oracle and (fragment of) a physical theory PT so that,*

1. *The queries to the oracle carry no parameters or additional information.*
2. *The answer to each query is 'yes' or 'no', and is given in a fixed constant time.*
3. *The answer is a random variable which is identically distributed for each query and independent of all previous answers.*

There is a physical parameter $\sigma \in [0,1]$ so that, according to the theory PT, there is a function $F : [0,1] \to [0,1]$ so that the following hold:

4. *Probability. The probability of 'yes' is given by the function $F(\sigma)$, and no other parameters.*
5. *Smoothness. F is a continuously differentiable function, i.e., $F \in C^1[0,1]$.*
6. *Lower Bound. There is a rational number $\nu > 0$ which is a lower bound for $|F'|$ in $[0,1]$.*

The condition on oracles carrying no parameters means that all queries are the same. Since F' is continuous and, for every $\sigma \in [0,1]$, $F'(\sigma) \neq 0$, it follows that F is monotonic (either strictly increasing or strictly decreasing) in the interval.

Next we turn to the connection with Turing machines. For ease of computation, since Turing machines process binary data, we shall use dyadic rationals (rational numbers whose denominator is a power of 2). We say that a dyadic rational $w \in [0,1)$ has length or size n if it can be written as a fraction with denominator 2^n and no smaller power of 2.

Definition 3.2. *A Turing machine $\mathcal{M}$ coupled to a physical oracle with parameter σ and function F satisfying the axioms of SPO is called an SPO-*machine *provided we can approximate F in the following sense: For all integers $n > 0$ and all dyadic rationals $w \in [0,1]$ of length $\leq n$ we can compute an approximation F_{calc} of F with $|F_{\mathrm{calc}}(w) - F(w)| \leq 2^{-n}$ in time $G(n) \in O(2^n)$, i.e., G can have up to exponential growing rate.*[†]

[†] This asymptotic time is not the time that the machine takes to consult the oracle, but only the time that it takes to compute the function F that gives the probability of outcome 'yes'.

We require the exponential bound on the computation time $G(n)$ in Theorem 5.1, we do not need the exact form of $G(n)$.

3.3 Checking the examples satisfy the axioms

The exponential law of radioactive decay was first set out in [19], and this exponential law of 1902 implies that the probability of decay of an atom does not depend on its age, an assumption now widely accepted. Using this as the theory PT we can show that the half-life experiment given earlier satisfies our axioms. In the following lemma we use a single atom as a matter of convenience – we could have used a given fixed number of atoms at the beginning of each experiment.

Lemma 3.1. *A radioactive isotope is known to have a half-life H contained an an interval $[H_{\rm low}, H_{\rm high}]$ where the end points are non-zero positive rational multiples of a given time unit. An experiment observes a single atom of the isotope from time zero to a fixed time T (a rational multiple of the time unit), and returns an answer 'yes' if the atom decays, and 'no' if the atom does not decay. This experimental observation is used as an oracle: by restarting the experiment, with a single atom of the isotope, every time a query is received. This experiment satisfies the axioms of an SPO, and gives an SPO-machine when coupled to a Turing machine.*

Proof: First we normalise the half-life by setting

$$\sigma = \frac{H - H_{\rm low}}{H_{\rm high} - H_{\rm low}} \in [0, 1] \, .$$

Using the formula from (1) we substitute for H in terms of σ to find

$$F(\sigma) = 1 - \exp(-\log(2)\, T/(H_{\rm low} + (H_{\rm high} - H_{\rm low})\sigma)) \, . \qquad (3)$$

The derivative is

$$F'(\sigma) = - \frac{\exp\big(-\log(2)\, T/(H_{\rm low} + (H_{\rm high} - H_{\rm low})\sigma)\big)(H_{\rm high} - H_{\rm low})T\, \log(2)}{(H_{\rm low} + (H_{\rm high} - H_{\rm low})\sigma)^2}$$

so we have $F(\sigma)$ continuously differentiable. Also

$$F'(0) = - \frac{2^{-T/H_{\rm low}}\, |H_{\rm high} - H_{\rm low}|\, T\, \log(2)}{H_{\rm low}^2}$$

$$F'(1) = - \frac{2^{-T/H_{\rm high}}\, |H_{\rm high} - H_{\rm low}|\, T\, \log(2)}{H_{\rm high}^2}$$

and the maximum value of $|F'(\sigma)|$ for $\sigma \in [0,1]$ is achieved at one of these points, so we can calculate a rational number $0 < \nu \leq \min |F'(\sigma)|$. There are standard algorithms for the calculation of $F(\sigma)$ for rational values of σ and the constants satisfying the exponential bound in Definition 3.2. □

Note that the chain reaction for fission of ^{235}U is not an exception to the principle of independence for radioactive decay, as this is caused by slow neutron capture altering the nucleus. However, experiments with exotic forms of β-decay have found cases where the decay rate is influenced by the environment of the atoms [20].

Lemma 3.2. *The random walk experiment with probability σ as parameter satisfies the axioms SPO.*

Proof: We will compare the random walk experiment to each of the four axioms on F. For (1) we call the oracle for a given fixed value of T (e.g., $T = 5$). Then (2) gives a formula for $F(\sigma)$ which is continuously differentiable for $\sigma \in [0,1]$, satisfying (2). For axiom (3) it is convenient to restrict to the case where $\sigma \in [\frac{1}{2}, 1)$. Then the derivative of $(1-\sigma)^{\frac{t+1}{2}} \sigma^{\frac{t+1}{2}-1}$ with respect to σ is

$$(1-\sigma)^{\frac{t-1}{2}} \sigma^{\frac{t-1}{2}-1} \big(\tfrac{t-1}{2} - (2\,\tfrac{t-1}{2} + 1)\sigma\big) \leq -\tfrac{1}{2}\,(1-\sigma)^{\frac{t-1}{2}} \sigma^{\frac{t-1}{2}-1}$$

for $\sigma \in [\frac{1}{2}, 1)$, so $F'(\sigma) \neq 0$ on $[\frac{1}{2}, 1)$. Thus we renormalise the interval and restrict the range of σ to satisfy the axioms. For axiom 4, note that for fixed T we have $F(\sigma)$ a polynomial with rational coefficients. □

3.4 Examples which do not satisfy the axioms

Here we give some variants of the previous examples which do not satisfy the axioms for SPO.

1. Lack of independence. Suppose that we have a starting quantity of radioactive isotope, and that each time a query is made, we observe this same quantity for 60 minutes (i.e., we do not get a new sample on each query), and then return 'yes' for at least one decay in this period, and 'no' for no decays in the period. This will not give identically distributed random variables, as there will be less atoms of the given isotope in the quantity as time increases. In fact they will also not be independent, as 'no' results to previous queries will bias future measurements towards 'yes', as fewer atoms will have decayed in the past.

2. Adding parameters and non-identical experiments. An obvious question is what happens if a parameter is passed to the experiment by the oracle call – e.g., the duration T of the experiment in the case of the single atom

radioactive decay experiment. For example, we could pass the parameters 60, 58, 62, 61,... for the number of minutes to observe the atoms in successive experiments. The results are then (according to our theories of radioactive decay) independent, but no longer identically distributed. It turns out that even this seemingly simple situation requires non-trivial statistics to recover a sensible confidence interval for the half-life, i.e., an interval containing H with a probability of say $\geq 95\%$.

In more generality (e.g., a machine reasoning from a few completely different measurements to determine the likely state of a physical system) it is not clear what the answer would be – probably Bayesian statistics would be the most practical method to use. However, there are differences between the Bayesian approach of credibility intervals and the more usual confidence intervals [21], and it would be necessary to take care in applying them to complexity theory.

3. Problems with an unbounded interval. The half life of the proton given by various theories is in the approximate range $[10^{30}, \infty]$ years [22], where ∞ means that the proton is in fact stable, a possibility consistent with all meaurements so far. The search for physical evidence of proton decay is hampered not only by the very long lifetime (if it indeed is not infinite), but also uncertainty over what the proton would decay into, meaning that no single experiment can cover all possibilities. Measurements put the half life of the proton (for some decay mechanisms) at not below 5.9×10^{33} years [23]. In contrast, devising a means for measuring the half life of Thorium 232 (which has known decay mechanism and a half life of approximately 1.4×10^{10} years) is rather simpler.

4. An example for which we have no lower bound $\nu > 0$ for $F'(\sigma)$ for $\sigma \in [0, 1]$. Take the random walk with an absorbing surface, for $\sigma \in [0, 1]$. In this case, for $T = 3$ we have $F(\sigma) = (1-\sigma) + (1-\sigma)^2\sigma$, and so $F'(0) = 0$.

4 CANTOR SETS AND ADVICE FUNCTIONS

4.1 Non-uniform complexity classes

Here we describe non-uniform complexity classes based on Turing machines processing words over $\{0, 1\}$, in particular we explain the class $BPP//\log\star$ mentioned in the Theorems in the introduction. A *non-uniform complexity class* has notation $\mathcal{B}/\mathcal{F}$, where $\mathcal{B}$ is a complexity class (such as P or Exp), and $\mathcal{F}$ is a class of advice functions. An advice function is a function $f : \mathbb{N} \to \{0, 1\}^*$, which roughly speaking gives additional information for solving a problem of given size; f is not necessarily computable.

Using the standard word acceptance and Turing machine model of complexity, we can make this more precise: Suppose we have a class $\mathcal{B}$ of sets of words which are to be accepted. Given a word w, we append $f(|w|)$, the advice associated with words of length $|w|$, to w, and write the combined word on the input tape, and ask if that word is in class $\mathcal{B}$. That is, the Turing machine is given some extra help in its computation, and that *advice* only depends on the length of the word.

Note that too much advice makes every possible subset of words computable, as the advice could be a list of all words of the given length that should be accepted. However, in general this would require the length of the advice $|f(|w|)|$ to be exponential in $|w|$. Thus to get interesting classes we place restrictions on the length of the advice.

Let $\mathcal{B}$ be a class of sets and $\mathcal{F}$ a class of total functions. The non-uniform class $\mathcal{B}/\mathcal{F}$ is the class of sets A for which some $B \in \mathcal{B}$ and some $f \in \mathcal{F}$ are such that, for every w, $w \in A$ if, and only if, the concatenation $\langle w, f(|w|)\rangle \in B$. For example, take poly to be the class of advice functions which have length bounded by a polynomial, and log to be the class with length bounded by a multiple of $\log_2$. Thus we have non-uniform classes such as P/poly, which has polynomially bounded advice and is computable in polynomial time.

We also need prefix non-uniform complexity classes. For these classes we use prefix functions, i.e., such that $f(n)$ is a prefix of $f(n+1)$. The idea behind prefix non-uniform complexity classes is that the advice for inputs of size n is also useful to decide smaller inputs. The notation is $\mathcal{B}/\mathcal{F}\star$, e.g., $P/\log\star$.

A probabilistic Turing machine T is a Turing machine with a fair coin toss oracle (i.e., probability 50% and independence of queries). For a set A of words, a probabilistic Turing machine T, and an input $w \in \Sigma^*$, the *error probability* of T for input w is the probability of T rejecting w if $w \in A$, or the probability of T accepting w if $w \notin A$. We say that T decides A with *bounded error probability* if there is a number $\gamma < \frac{1}{2}$, such that the error probability of T for any input w is smaller than γ. We write BPP to stand for the class of sets decidable by T in polynomial time with bounded error probability. Here is the primary complexity class of interest:

Definition 4.1. *$BPP//\log\star$ is the class of sets A for which a probabilistic Turing machine T, a prefix function $f \in \log\star$, and a constant $\gamma < \frac{1}{2}$ exist such that, for every length n and input w with $|w| \leq n$, T rejects $\langle w, f(n)\rangle$ with probability at most γ if $w \in A$ and accepts $\langle w, f(n)\rangle$ with probability at*

most γ *if* $w \notin A$.

The classes $BPP//\log\star$ and $BPP//\log$ both use advice of logarithmic size for a query of size n, but the difference is that the advice for size n for $BPP//\log\star$ is also good advice for all shorter queries. To be more precise, for $BPP//\log\star$ there is a single infinite word so that the initial segment of length $\log_2(an+b)$ (for some constants a, b) is the advice for words of length n. If we used polynomial size instead of logarithmic the star would be redundant, but for logarithmic size the ideas differ as the sum of $\log_2(am+b)$ from $m = 1$ to $m = n$ is not logarithmic in n. We should also point out that the value of the error probability $\gamma < \frac{1}{2}$ is not in itself significant. To get an error probability of less than any specified amount all we have to do is to repeat the entire process several times (assuming the repeats are independent) and take the majority verdict. The condition $\gamma < \frac{1}{2}$ is just what we need to ensure that the total error for the repeated process tends to zero (in a controlled manner) as the number of repeats tends to infinity. Some definitions specify the value $\gamma = \frac{1}{3}$.

The reader should also note that we use $BPP//\log\star$ which has a slightly different definition from $BPP/\log\star$. Given the standard definition of single / in complexity theory, for $BPP/\log\star$ we would have to specify a value of γ which was true for *all* good advice functions $f \in \log\star$! For $BPP//\log\star$ we only have to find a γ for one choice of good advice, which is far more practical. However, $BPP//\log\star$ seems to be little understood as a complexity class [24].

4.2 Encoding the advice

To prove that SPO-machines compute some non-uniform complexity class, we encode the advice functions into the real numbers of the Cantor set

$$\mathcal{C}_3 = \left\{ \sum_{k=1}^{\infty} x_k 2^{-3k} : x_k \in \{1, 2, 4\} \right\} ,$$

where every number has a binary expansion composed of triplets 001, 010, or 100. The advantage of this particular encoding is that there is no need to distinguish between arbitrarily close representations such as 0.10^{ω} and 0.01^{ω} to get the most significant bit. This a very common technique in the field of computing with real numbers (see [25, 26, 27]). Our encoding comes from [5, 18] and the proof of the following theorem can be found in [18, 28].

Proposition 4.1 (Beggs et al. [18])**.** *For every* $x \in \mathcal{C}_3$ *and for every dyadic rational* z *with size* $|z| = m$*,*

1. *if* $|x - z| \leq \frac{1}{2^{i+5}}$*, then the binary expansions of* x *and* z *coincide on the first* i *bits;*

2. $|x - z| > \frac{1}{2^{m+10}}$. □

The encoding of a word $w \in \{0,1\}^\star$ is denoted by $c(w)$ and is obtained replacing every 0 in w by 100 and every 1 by 010. Given an advice function $f : \mathbb{N} \to \{0,1\}^\star$ in $\log\star$, the encoding $y(f) = \lim y(f)(n)$ of f is recursively defined beginning with $y(f)(0) = 0.c(f(0))$. Next, as $f(n)$ is a prefix of $f(n+1)$ we write $f(n+1) = f(n)s$, and then define

$$y(f)(n+1) = \begin{cases} y(f)(n)c(s) & \text{if } n+1 \text{ is not a power of 2} \\ y(f)(n)c(s)001 & \text{if } n+1 \text{ is a power of 2} \end{cases}$$

By construction we have that $y(f) \in \mathcal{C}_3$, since the procedure above corresponds to replacing the bits of f by 100 and 010 and adding separators 001 at the end of the codes for every $f(2^k)$, $k \in \mathbb{N}$. To extract the value of $f(n)$, we just have to find the number $m \in \mathbb{N}$ such that $2^{m-1} < n \leq 2^m$ and then read $y(f)$ in triplets, until we find the $(m+1)$-th separator. Then, it is only needed to ignore the separators and replace each 100 triplet by 0 and each 010 triplet by 1. Since f has logarithmic size bound, we know that $|f(2^m)| = O(\log 2^m) = O(m)$, so that we need $3O(m)+3(m+1) = O(m)$ bits to get the value of $f(2^m)$ and, consequently, $O(\log n)$ bits to get the value of $f(n)$.

5 LOWER BOUNDS

We show now how an SPO-machine recovers the advice relative to the current input size. The following lemma generalizes results found in [5, 8, 18, 29] and in [27], and will be used in this section to establish lower bounds on the computational power of SPO-machines.

Theorem 5.1. *An* SPO*-machine with 'yes' probability* $F(\sigma)$*, with* $\sigma \in \mathcal{C}_3$*, given an integer* $n \geq 1$*, can read* $O(\log(n))$ *bits of the unknown parameter* σ *in polynomial time in* n*, with error probability bounded by any given* $\gamma \in (0,1)$.

Proof: Each oracle call has an associated probability $P(\text{`}yes\text{'}) = F(\sigma)$, so if we make ξ oracle calls, the number of times α that the experiment returns 'yes' is a random variable with binomial distribution. Let $X = \alpha/\xi$ be the relative frequency of 'yes'. The expected value of X is $\mathbb{E}[X] =$

$\mathbb{E}[\alpha]/\xi = \xi F(\sigma)/\xi = F(\sigma)$ and the variance of X is $\mathbb{V}[X] = \mathbb{V}[\alpha]/\xi^2 = \xi F(\sigma)(1-F(\sigma))/\xi^2 = F(\sigma)(1-F(\sigma))/\xi$. Chebyshev's inequality states that, for every $\delta > 0$,

$$P(|X - \mathbb{E}[X]| > \delta) \leq \frac{\mathbb{V}[X]}{\delta^2} \leq \frac{F(\sigma)(1-F(\sigma))}{\xi\delta^2} \leq \frac{1}{\xi\delta^2} .$$

If $|\sigma - u| \leq 2^{-k-6}$ then there is a dyadic rational of length $k+5$ with $|z-u| \leq 2^{-k-6}$, so $|\sigma - z| \leq 2^{-k-5}$ and by Proposition 4.1 (1) the first k bits of σ are then the first k bits of z. Suppose that we have chosen u so that there is a v with $|v-u| \leq 2^{-k-7}$ with $F(v) = X$. Now by the Mean Value Theorem,

$$|\sigma - u| \leq |\sigma - v| + |v-u| \leq \frac{|F(\sigma) - F(v)|}{\nu} + |v-u| \leq \frac{|F(\sigma) - X|}{\nu} + 2^{-k-7} .$$

From Chebyshev's inequality

$$P(|X - F(\sigma)| > \nu\, 2^{-k-7}) \leq \frac{2^{2k+14}}{\xi\nu^2} .$$

If we want an error probability of at most γ, we must make a number of oracle calls given by

$$\xi \geq \frac{2^{2k+14}}{\gamma\nu^2} .$$

Then, to a probability of error $\leq \gamma$, we have $|\sigma - u| \leq 2^{-k-6}$, and we can then find the first k digits of σ. After having made ξ oracle calls and having calculated X, we now need to find u as required, i.e., so that there is a v with $|v-u| \leq 2^{-k-7}$ with $F(v) = X$, and we do this by bisection. First if $X \notin F([0,1])$,[‡] then we set X to the minimum or maximum value of $F([0,1])$, depending on X being below or above $F([0,1])$.

Now remember that we can only compute F_{calc}, an approximation to F. We look for sign changes in $F_{\mathrm{calc}} - X$, where $|F_{\mathrm{calc}}(w) - F(w)| \leq \nu\, 2^{-k-8}$ for all w. To simplify things we assume that F is increasing (decreasing is similar). By bisection, we find $v_1 \leq v_2$ so that $F_{\mathrm{calc}}(v_1) \leq X \leq F_{\mathrm{calc}}(v_2)$ and $|v_2 - v_1| \leq 2^{-k-7}$. This requires constant plus k evaluations of F_{calc}, at an accuracy requiring time $G(\mathrm{const} + k)$ each. Now set $u = (v_1 + v_2)/2$. Now if the value v with $F(v) = 0$ is in the interval $[v_1, v_2]$ we are done. If not, we suppose $v \geq v_2$ (the other case $v \leq v_1$ is very similar). Given the accuracy we calculate F_{calc} at, we have

$$X = F(v) \geq F(v_2) \geq X - \nu\, 2^{-k-8} ,$$

[‡] The continuous image of a compact set is a compact set.

so $|F(v) - F(v_2)| \leq \nu\, 2^{-k-8}$, so $|v - v_2| \leq 2^{-k-8}$, so $|v - u| \leq 2^{-k-7}$.

In conclusion, using Axiom 4 of SPO in Section 3 we see that we can find logarithmically many digits of σ in polynomial time, with a given error probability. □

Note that we considered that the oracle responses are independent and identically distributed (as stated in the axioms), from which Chebyshev's inequality follows. Otherwise, we have no guarantee that there is convergence of the sequence of relative frequencies (see [30] for an interesting discussion on the existence of a limit for relative frequencies).

We now define the concept of a set decided by an SPO-machine in the context of this paper:

Definition 5.1. *We say that an SPO-machine $\mathcal{M}$ with unknown parameter σ and with 'yes' probability $F(\sigma)$ decides $A \subseteq \{0,1\}^\star$ if there exists $\gamma < 1/2$ such that, for every input w, $\mathcal{M}$ halts and*

- *If $w \in A$, $\mathcal{M}$ accepts w with error probability bounded by γ;*
- *If $w \notin A$, $\mathcal{M}$ rejects w with error probability bounded by γ.*

Now, we can state and prove the lower bound theorem for our SPO-machines.

Theorem 5.2. *Every set A in $BPP//\log\star$ is decidable by an SPO-machine with 'yes' probability $F(\sigma)$, for some $\sigma \in \mathcal{C}_3$, in polynomial time. I.e. $w \in A$ is decidable in time polynomial in its length $|w|$.*

Proof: Let A be an arbitrary set in $BPP//\log\star$ and $\mathcal{M}$ a probabilistic Turing machine with advice $f \in \log\star$, which decides A in polynomial time with error probability bounded by $\gamma_1 \in (0, 1/2)$.

Let $\mathcal{M}'$ be an SPO-machine with parameter $\sigma = y(f)$, where $y(f)$ represents the encoding of f and let γ_2 be a positive real number such that $\gamma_1 + \gamma_2 < 1/2$. Let w be a word of size n. We know, by Theorem 5.1, that $\mathcal{M}'$ can estimate a logarithmic number of bits of $y(f)$ and read the same amount of bits of $f(n)$ in polynomial time with an error probability bounded by γ_2.

We need now to simulate independent unbiased coin tosses. With probability at least $1 - \gamma$, we can use a sequence of independent biased coin tosses of length

$$\frac{s}{F(\sigma)(1 - F(\sigma))}\left(1 + \frac{1}{\gamma^{1/2}}\right)$$

to produce a sequence of s independent fair coin tosses (see the proof e.g. in [5, 28]).

Let γ be such that $\gamma_1 + \gamma_2 + \gamma < 1/2$. Machine $\mathcal{M}'$ can produce a sequence of s fair coin tosses in time linear in $|s|$ to within a probability of γ. Therefore, $\mathcal{M}'$ can decide A in polynomial time, with error probability bounded by $\gamma_1 + \gamma_2 + \gamma < 1/2$. □

6 PROBABILISTIC TREES

In this section, we introduce the concept of a probabilistic tree, which been used in, e.g., [6, 18, 28]. Here, we clarify the concept for our particular use.

During the computation of an SPO-machine $\mathcal{M}$ on a particular input w, the finite control of the machine may eventually reach the query state triggering the physical experiment, and, after the scheduled time, it will be in one of two states, the 'yes' state or the 'no' state. Then the machine performs some deterministic computations before a new call to the oracle is made. Therefore, the oracle calls of the SPO-machine can be represented in a binary tree.

Definition 6.1. *A query tree T is an ordered tree (V, E), associated with an oracle Turing machine, where each node in V is a configuration of the machine in the query state or in a halting state and each edge in E is a deterministic computation of the Turing machine between consecutive oracle calls or between oracle calls and halting configurations. Moreover, every node with zero children is called a leaf and is labeled with 'A' or 'R', depending on whether it abstracts an accepting or rejecting configuration.*

In a query tree, every internal node represents a call to the oracle. As our oracle is stochastic, there is a probability associated to each outcome of the oracle. Thus, we can assign probabilities to the edges of the query tree. A single computation of a Turing machine, over the input w, corresponds to a path in the tree, beginning in the root and ending in a leaf. If the path ends in a leaf labeled with 'A', the machine halts in an accepting state, otherwise, it halts in a rejecting state.

Definition 6.2. *A probabilistic tree is a pair (T, D) where T is a query tree, V being its set of nodes and E its set of edges, and $D : E \to [0, 1]$ is an assignment of probabilities to the edges of T, such that the probabilities of the outgoing edges of every internal node sum to* 1.

Let $\rho(T)$ be the set of all possible assignments of probabilities to the edges of T.

Definition 6.3. *Given a probabilistic tree (T, D), the probability $P_A(T, D)$ of acceptance is the sum of the products of probabilities along the edges in every path that leads to an accepting node, i.e.,*

$$P_A(T, D) = \sum_{c \in C_A} \left(\prod_{i=1}^{m_c} D(c[i]) \right),$$

where C_A is the set of all paths that end in an accepting leaf, m_c is the length of the path c and $c[i]$ is the i-th edge of that path. Similarly, we define probability $P_R(T, D)$ of rejection as the sum of the products of probabilities along the edges in every path that leads to a rejecting node.

$$P_R(T, D) = \sum_{c \in C_R} \left(\prod_{i=1}^{m_c} D(c[i]) \right),$$

where C_R is the set of all paths that end in a rejecting leaf. (We use the convention that the sum over an empty set is 0, and a product over an empty set is 1.)

Note that if the tree has a bound on its depth, then we can assume that all leaves are at the same depth, i.e., all the paths from the root to the leaves have the same length, because, even if some computation ends without making m calls to the oracle, we can continue the computation, doing nothing besides the oracle calls, and assigning to all the resulting leaves the same label, depending on the state in which the computation had ended.

Definition 6.4. *The distance between probabilistic trees (T, D) and (T, D') with the same query tree T, denoted by $d(D, D')$, is given by $d(D, D') = \max_{e \in E}\{D(e) - D'(e)\}$.*

We can restrict the domain of trees that we want to work with by focusing on t-ary probabilistic trees, i.e., probabilistic trees such that each internal node of the query tree has exactly t children. Denote by $\mathcal{T}_m^t$ the set of all t-ary probabilistic trees of depth m. If $m = 0$, the probabilistic tree is composed of one node, corresponding to the case where the machine does not consult the oracle.

Definition 6.5. *For every $m, t \in \mathbb{N}$ and $\beta \in [0, 1]$, define a function $f_t : \mathbb{N} \times [0, 1] \to [0, 1]$, that gives the largest possible difference in the probability of acceptance for two probabilistic trees with the same t-ary query tree of depth m, such that their distance is bounded by β:*

$$f_t(m, \beta) = \sup \big\{ |P_A(T, D) - P_A(T, D')| : \\ T \in \mathcal{T}_m^t, D,\ D' \in \rho(T),\ d(D, D') \leq \beta \big\} .$$

The proof of the following proposition is provided in [18, 28].

Theorem 6.1. *For every $m, t \in \mathbb{N}$, for every $\beta \in [0,1]$,*

$$f_t(m,\beta) \leq (t-1)m\beta \ .$$

The following statement generalizes results in [6, 8]:

Theorem 6.2. *Let $\mathcal{M}$ be an SPO-machine with unknown parameter σ and with 'yes' probability $F(\sigma)$, deciding a set A in time $t(n)$ with error probability bounded by $\gamma < 1/4$. Let $\mathcal{M}'$ be an identical SPO-machine, with the exception that the parameter is $\tilde{\sigma}$ and the probability of 'yes' is $\tilde{F}(\tilde{\sigma})$. If*

$$|F(\sigma) - \tilde{F}(\tilde{\sigma})| < \frac{1}{8t(n)},$$

then for any word of size $\leq n$, the probability of $\mathcal{M}'$ making an error when deciding A is less than or equal to $3/8$.

Proof: Since in time $t(n)$ at most $t(n)$ calls to the oracle can be made, we conclude that the query tree T associated to $\mathcal{M}$ and $\mathcal{M}'$ has maximum depth $t(n)$. Note that the query tree of $\mathcal{M}$ and $\mathcal{M}'$ is the same, since the machines only differ in the probabilities associated with the calls to the oracle. Let w be a word such that $|w| \leq n$. Let $D \in \rho(T)$ be the assignment of probabilities to the edges of T corresponding to the parameter σ and 'yes' probability $F(\sigma)$ and $D' \in \rho(T)$ the assignment of probabilities given by parameter $\tilde{\sigma}$ and 'yes' probability $\tilde{F}(\tilde{\sigma})$. Since $|F(\sigma) - \tilde{F}(\tilde{\sigma})| < 1/(8t(n))$, the difference between any particular probability is at most $\mu = 1/(8t(n))$. Two and only two cases can arise:

- $w \notin A$: In this case, an incorrect result corresponds to $\mathcal{M}'$ accepting w. Applying Theorem 6.1, we can bound the probability of acceptance for $\mathcal{M}'$ in the following way:

$$\begin{aligned} P_A(T,D') &\leq P_A(T,D) + |P_A(T,D') - P_A(T,D)| \\ &\leq \gamma + t(n)\mu \\ &\leq \gamma + t(n) \times \frac{1}{8t(n)} \\ &= \frac{1}{4} + \frac{1}{8} \\ &= \frac{3}{8} \end{aligned}$$

- $w \in A$: In this case, an incorrect result corresponds to $\mathcal{M}'$ rejecting w. Applying Theorem 6.1, we can bound the probability of rejection for $\mathcal{M}'$ in the following way:

$$\begin{aligned} P_R(T, D') &\leq P_R(T, D) + |P_R(T, D') - P_R(T, D)| \\ &\leq \gamma + t(n)\mu \\ &\leq \gamma + t(n) \times \frac{1}{8t(n)} \\ &= \frac{1}{4} + \frac{1}{8} \\ &= \frac{3}{8} \end{aligned}$$

In both cases, the error probability is bounded above by $3/8$. □

Theorem 6.2 has the following corollary, where we denote by $a\lfloor_\ell$ the first ℓ digits of a, if a has ℓ or more than ℓ digits, otherwise it represents a padded with k zeros, for some k such that $|a0^k| = \ell$.

Corollary 6.1. *Let $\mathcal{M}$ be an SPO-machine with unknown parameter σ and with 'yes' probability $F(\sigma)$, deciding some set in time $t(n)$ with error probability bounded by $\gamma < 1/4$. Let $\mathcal{M}_n$ be an identical SPO-machine, also with unknown parameter σ, but with the exception that the probability of 'yes' is given by $F(\sigma)\lfloor_{\log t(n)+3}$. Then, $\mathcal{M}_n$ decides the same set as $\mathcal{M}$, also in time $t(n)$, but with error probability bounded by $3/8$.*

7 UPPER BOUNDS

In the proof of the next theorem, we build a probabilistic tree as defined in Section 6, from the computation tree of a probabilistic Turing machine.

Theorem 7.1. *Every set decided by an SPO-machine with unknown parameter σ clocked in polynomial time is in $BPP//\log\star$.*

Proof: Let A be a set decided in polynomial time $t(n)$ by an AD machine $\mathcal{M}$ with unknown parameter σ and with error probability bounded by $1/4$. We specify a probabilistic Turing machine $\mathcal{M}'$ to decide A in polynomial time with the help of the advice function $f(n) = F(\sigma)\lfloor_{\log t(n)+3} \in \log\star$.

By Corollary 6.1, we know that an SPO-machine with 'yes' probability $f(n)$ decides $A^{\leq n}$, the finite subset of A comprising all the words of length less or equal to n, with error probability $\leq 3/8$. Machine $\mathcal{M}'$ simulates $\mathcal{M}$ but with 'yes' probability $f(n)$.

Since $f(n) = F(\sigma)\downarrow_{\log t(n)+3}$ is a dyadic rational with denominator given by $2^{\log t(n)+3}$, we conclude that $m = 2^{\log t(n)+3} f(n) \in [0, 2^{\log t(n)+3}]$ is an integer. Machine $\mathcal{M}'$ tosses a fair coin $k = \log t(n) + 3$ times, obtaining the binary sequence $\tau_1\tau_2 \ldots \tau_k$ to be viewed as the binary representation of an integer. If the test $\tau_1\tau_2 \ldots \tau_k < m$ is true, $\mathcal{M}'$ interprets the result as 'yes', otherwise as 'no'. The probability of returning 'yes' is $m/2^k = f(n)$, as required. The time taken is polynomial in n.

We now prove that the probabilistic Turing machine $\mathcal{M}'$ has an error probability bounded by $3/8$.

The computation tree of $\mathcal{M}'$ has maximum depth $t(n)(\log t(n)+3)$, since there are at most $t(n)$ calls to the oracle and $k = \log t(n) + 3$ tosses of a coin for each call. Moreover, those are the only steps of the computation in which $\mathcal{M}'$ behaves probabilistically. To obtain a probabilistic tree (as in Section 6) that corresponds to this computation tree, we proceed in the following way: for every level $i \in \mathbb{N}$, we create a new level in the probabilistic tree, assigning probability $f(n)$ to every left edge and $1 - f(n)$ to every right edge. The obtained tree has depth $t(n)$.

Now, let us see that the probabilities involved in the probabilistic tree and in the probabilistic Turing machine are the same. In level k of the computation tree, we have 2^k nodes, and we know that m of those correspond to the result 'yes' and, consequently, their subtrees are equal. For each of those m nodes, if we descend k more levels in their subtrees, we have m nodes that correspond to 'yes' result. Thus, at level $2k$, we know that there are m^2 nodes corresponding to a 'yes' in the two first experiments, and so on. When we have traversed the whole tree, we know that in the last level, we have $2^{kt(n)}$ leaves, and consequently $m^{t(n)}$ leaves correspond to having had 'yes' in all the experiments. We conclude that the probability of having 'yes' in all the experiments is

$$\frac{m^{t(n)}}{2^{kt(n)}} = \left(\frac{m}{2^k}\right)^{t(n)} = f(n)^{t(n)} .$$

In the probabilistic tree, this probability would be the product of $t(n)$ 'yes' probabilities, so $f(n)^{t(n)}$, as expected. The probabilities of 'no' coincide as well. □

Using Theorems 5.2 and 7.1, we get the following corollary:

Corollary 7.1. *The class of sets decidable in polynomial time by SPO-machines with unknown parameter σ is exactly $BPP//\log\star$.*

8 RELATION TO P-PROBABILISTIC TURING MACHINES

What is the relationship between our stochastic physical oracles and the many varied forms of probabilistic computation, especially probabilistic Turing machines? Given a specification that requires a probabilistic computation, does there exist a physical oracle which computes it? To try to answer these questions, we look at different models of random computation involving Turing machines.

8.1 Probabilistic Turing machines and probabilistic oracles

The idea of probabilistic Turing machines was introduced in [31]. A p-machine was defined to be a Turing machine with a special probabilistic print instruction, printing a 1 with probability p and 0 with probability $1-p$. This probabilistic print command is equivalent to a probabilistic oracle, which on being queried returns 1 with probability p and 0 with probability $1-p$ (with each query being independent of all the others). It was noted that the sets computable by these machines depended on whether p was computable or not. Probabilistic complexity classes, including BPP were introduced in [32], and their relation to NP discussed.

In [8] some axioms for a probabilistic oracle were given, and it was shown that such a probabilistic oracle had a computational power in polynomial time of $BPP//\log\star$. We shall adopt the following definition for the sake of being definite:

Definition 8.1. *For $p \in (0,1)$, a p-probabilistic Turing machine is a Turing machine equipped with an oracle that to every query returns, in unit time, 1 with probability p and 0 with probability $1-p$, independently of any other queries. For a word acceptance problem we can assume that all computations on the same input require the same number of oracle calls, and that every computation ends with reject or accept.*

The tree of computations of a p-probabilistic Turing machine is a full binary tree, i.e., all leaves are at the same depth, and a computation of the machine consists of a path from the root to a leaf being an accepting or rejecting computation, depending on whether it ends in *reject* or *accept*, respectively. We associate with each computation a probability which is the product of the probabilities at each computation step. The probability of accepting an input is the sum of the probabilities associated with the accepting computations of the machine on the given input. The error probability of the machine on some input is the sum of the probabilities associated with the computations that give the wrong answer.

Definition 8.2. *We say that a p-probabilistic Turing machine $\mathcal{M}$ clocked in polynomial time decides the set $A \subseteq \{0,1\}^\star$ with bounded error probability if $\mathcal{M}$ halts in polynomial time on all inputs and there exists $\gamma < 1/2$ such that, for every input w,*

- *If $w \in A$, $\mathcal{M}$ accepts w with error probability bounded γ;*
- *If $w \notin A$, $\mathcal{M}$ rejects w with error probability bounded by γ.*

8.2 Stochastic physical oracles

We analyse the relation between p-probabilistic Turing machines and our SPO-machines. While we could do this by equating the calculated complexity classes, it is more informative to examine the correspondence directly.

Theorem 8.1. *Every set decided by an SPO-machine clocked in polynomial time with unknown parameter σ and with 'yes' probability $F(\sigma)$ is decided in polynomial time by a p-probabilistic Turing machine with bounded error probability.*

Proof: Let A be a set decided by an AD machine $\mathcal{M}$ with unknown parameter σ and 'yes' probability $F(\sigma)$ in polynomial time $t(n)$ and with error probability bounded by $\gamma < 1/2$. Let $p = F(\sigma)$. We construct a p-probabilistic Turing machine $\mathcal{M}'$ to decide A.

A p-probabilistic machine can perform deterministic computations, being sufficient that it makes the same operations on both outcomes of each computation step. For the oracle calls, machine $\mathcal{M}'$ just has to make one transition, in which, with probability $F(\sigma)$ it simulates the transition of $\mathcal{M}$ relative to the outcome 'yes', and with probability $1 - F(\sigma)$ it simulates the transition of $\mathcal{M}$ relative to the outcome 'no'.

The machine $\mathcal{M}'$ makes the same number of transitions than machine $\mathcal{M}$ and therefore runs in polynomial time. Also, as it makes the exactly same operations than $\mathcal{M}$, with the same probability, $\mathcal{M}'$ decides A with error probability bounded by the same $\gamma < 1/2$. □

To prove the reverse statement, we can take a special case of SPO-machine.

Theorem 8.2. *Every set decided by a p-probabilistic Turing machine clocked in polynomial time with bounded error probability is decided in polynomial time by an SPO-machine.*

Proof: Let A be a set decided by a generalized probabilistic Turing machine $\mathcal{M}$ with transition probability p in polynomial time and with error probability bounded by $\gamma < 1/2$.

Let $\mathcal{M}'$ be an SPO-machine with $F : [0,1] \to [0,1]$ being the identity, and set $\sigma = p$. Then calls to the stochastic oracle are the same as the oracle calls we can use to define the p-probabilistic Turing machine. □

Then we can state the following theorem:

Theorem 8.3. *The class of sets decidable by generalized probabilistic Turing machines with bounded error probability clocked in polynomial time is exactly $BPP//\log\star$.*

Proof: From Theorems 8.1 and 8.2, we see that generalized probabilistic machines have exactly the same computational power as SPO-machines with 'yes' probability $F(\sigma) = \sigma$. By Corollary 7.1, we know that these machines decide exactly $BPP//\log\star$. □

9 CONCLUSION

We considered a Turing machine coupled with a non-deterministic physical system as an oracle. Observing the physical system results in two possible events as outcomes, conveniently labelled 'yes' or 'no' with certain probabilities. No parameters are needed to observe the physical system. Thus, thinking of the physical system as an experiment to measure a quantity, no data is passed in order to initialise or sustain the experiment. After reflecting on examples of three types of experiments with non-deterministic behaviour, we proposed a single set of axioms SPO for stochastic physical oracles that imposed conditions on the calculation of the probabilities from physical quantities or parameters. Then, we proved lower and upper bounds on the computational power in polynomial time of Turing machines with physical oracles satisfying SPO, obtaining in both cases the non-uniform class $BPP//\log\star$.

For completeness, we considered the mathematical relationship between Turing machines with physical oracles satisfying SPO and the classic probabilistic Turing machines. Introducing the p-probabilistic Turing machine that is a probabilistic machine with transition probabilities $p \in (0,1)$ and $1-p$. we showed p-probabilistic machines clocked in polynomial time compute exactly $BPP//\log\star$.

9.1 Background on deterministic physical oracles

Our work on a theory of physical oracles began in [5, 6] and has focussed on classic, simple, deterministic experiments to measure quantities. In modelling these experiments, we analysed the interface that enables a physical process

to exchange data with an algorithm. The mathematical theory that took shape was different from that of oracles in computability and complexity theory, because physical oracle queries are subject to

(a) error margins in data used to make measurements, specifically the three types *exact precision*, *unbounded precision*, and *fixed precision*; and

(b) time delays needed to make measurements.

In keeping with the physical objectives of the theory, computations were also subject to

(c) feasible runtimes, especially that of *polynomial time*.

As the portfolio of experiments grew, results emerged that enabled us to develop axiomatic methods to show how physical processes could become physical oracles and, furthermore, to classify precisely their computational power as physical oracles [8].

Thus, by means of many case studies, and general mathematical methods, we showed that the computational power of a large class of deterministic experiments, acting as physical oracles to Turing machines running in polynomial time, is

$P/\log\star$, if queries operate with exact precision or unbounded precision; and

$BPP//\log\star$, if queries operate with fixed precision.

The relevance to the physical world of the Church-Turing thesis has been the subject of an enormous number of commentaries and speculations, especially in the wake of Kreisel's physical questions about the Church-Turing thesis (see the collection [33]). The area has received more sophisticated attention in recent years. The computational power of dynamical systems has been extensively mapped by Bournez; for comments on the Church-Turing thesis see [34] on analog and hybrid systems. Computation relative to particular theories of physics have been proposed in the methodologies of our [35] and in Ziegler's thorough analysis [36]. The results led us to speculate on a Church-Turing thesis for analogue-digital computation [37].

Our investigation also yielded insight into the process of measurements [29, 38, 39]. A survey of our previous deterministic work is [28].

For some time, the emergence of probability in the fixed error case seemed to be an anomaly: Could it be an artefact of the different probabilistic methods we needed to tackle this tricky case? However, in this paper we have shown that the methods we used to solve the fixed precision case uncover and address subtle problems of non-determinism in error handling. Furthermore, we have shown that the methods can be extended to tackle the next phase of theory building, namely the use of more advanced non-deterministic experi-

ments as physical oracles. Our results here support the proposal in [37] that $BPP//\log\star$ is the limit of physical computation.

9.2 Scope of Physical Oracles

We have emphasised that the idea of physical oracles – depicted in Figure 1 – has wide scope. Indeed, the purpose of this paper is to widen the scope by extending the theory to stochastic oracles. As the Figure 1 shows, there are four ingredients: the physical process; its observation and measurement; the algorithm that calls on the measurements; and the theory that enables us to specify, design and reason about the entire set-up. Our development of the theory of physical oracles has been shaped by thinking about measurement in physics. It would be interesting to analyse applications in other areas of science and technology, to see how the theory applies and adapts; we mentioned three other domains in the introduction.

Our idealisations of physical oracles and algorithms are expressed in a mathematical theory. Therefore, a more pure mathematical form of the theory can also be developed, about abstractly conceived dynamical systems involving continuous and discrete data. Stripping out the application domain leaves a pure mathematical theory of algorithms that depend on real number parameters in various ways.

Interestingly, whilst we have worked independently on physical oracles, we have benefited from work on algorithms that depend on real number parameters. An example is the *artificial recurrent neural net* (ARNN), which is a model of computation in discrete time with real valued parameters, where computation happens through programming and learning. The ARNN system satisfies the classical constraints of computation theory, namely, (a) input is discrete (binary) and finite, (b) output is discrete (binary) and finite, and (c) the system control is finite. Infiniteness enters with the real number weights and parameters; a real value that can be a physical magnitude directly affecting the computation or a probability or any other value held in the system.

Three classes of neural nets are worth noting. In the first class $Net[\mathbb{Z}]$ the weights are integers, and the nets are equivalent to classical finite automata. In the second class $Net[\mathbb{Q}]$ the weights are rationals, and the nets are equivalent to Turing machines. Finally, in the third class $Net[\mathbb{R}]$ the weights are reals. In polynomial time, these latter networks compute some non-recursive functions such as (a unary encoding of) the halting problem of Turing machines. A net can process a real-valued weight, reading it bit by bit, making the class of sets decided by the deterministic systems in $Net[\mathbb{R}]$ in polynomial time coincide with P/poly; or the net can process a probability of transition, making

the class of sets decided by the stochastic systems in $Net[\mathbb{R}]$ in polynomial time coincide with $BPP//\log\star$. In both cases this measurement sounds unrealistic since the function involved in computing the so-called activation of the neurones (processors) is the well-behaved piecewise linear function. In an attempt to recover the classical analytic sigmoid function, in [40] the power of the deterministic ARNN in polynomial time drops to $P/\log\star$ as shown in [41].

The association of non-uniform complexity classes that break the Turing barrier has led to controversies, especially in the case of ARNNs where popular science took up speculations on hypercomputation. In [42], Martin Davis states that the only way a machine can go beyond the Turing limit is being provided with non-computable information, and in [43] argued that, even if a machine could compute beyond the Turing limit, we would not be able to certify that fact (a phenomenon that can be well understood in [30]). In [44], Younger et al. still fight for the realization of super-Turing machines with their electronic engineering project.

REFERENCES

[1] Alan Turing. Systems of logic based on ordinals. *Proceedings of the London Mathematical Society*, Second series, 45:161–228, 1939.

[2] Emil Post. Degrees of recursive unsolvability. *Bulletin of the American Mathematical Society*, 54, 641–642, 1948.

[3] Robert I. Soare. *Recursively Enumerable Sets and Degrees: A Study of Computable Functions and Computably Generated Sets*. Springer-Verlag, 1987.

[4] Scott Aaronson, `https://complexityzoo.uwaterloo.ca/Complexity_Zoo`

[5] Edwin Beggs, José Félix Costa, Bruno Loff, and John V. Tucker. Computational complexity with experiments as oracles. *Proceedings of the Royal Society, Series A (Mathematical, Physical and Engineering Sciences)*, 464(2098):2777–2801, 2008.

[6] Edwin Beggs, José Félix Costa, Bruno Loff, and John V. Tucker. Computational complexity with experiments as oracles II. Upper bounds. *Proceedings of the Royal Society, Series A (Mathematical, Physical and Engineering Sciences)*, 465(2105):1453–1465, 2009.

[7] José Luis Balcázar, Josep Días, and Joaquim Gabarró, *Structural Complexity I*, Springer-Verlag, 2nd edition, 1988, 1995.

[8] Edwin J. Beggs, José Félix Costa, and John V. Tucker. Axiomatizing physical experiments as oracles to algorithms. *Philosophical Transactions of the Royal Society, Series A (Mathematical, Physical and Engineering Sciences)*, 370(12):3359–3384, 2012.

[9] Edwin Beggs, José Félix Costa, and John V. Tucker. The impact of models of a physical oracle on computational power. *Mathematical Structures in Computer Science*, 22(5):853–879, 2012.

[10] José Luis Balcázar and Montserrat Hermo. The structure of logarithmic advice complexity classes. *Theoretical Computer Science*, 207(1):217–244, 1998.

[11] Salvador Elias Venegas-Andraca. *Quantum Walks for Computer Scientists*. Morgan and Claypool Publishers, 2008.

[12] Titus Lucretius Carus, *Of the Nature of Things*, (Translator) William Ellery Leonard, Project Gutenberg Release #785, 1st Cent. B.C.

[13] Brown, Robert, A brief account of microscopical observations made in the months of June, July and August, 1827, on the particles contained in the pollen of plants; and on the general existence of active molecules in organic and inorganic bodies, *Philosophical Magazine* 4: 161-173, 1828.

[14] Albert Einstein. Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen. *Annalen der Physik*, 322 (8) 1905.

[15] Los Alamos Science no. 15, Special Issue, *Stanislaw Ulam 1909-1984*, 1987.

[16] Peter Hellekalek, Good random number generators are (not so) easy to find. *Mathematics and Computers in Simulation* 46, 485-505, 1998.

[17] Cristian S. Calude, Michael J. Dinneen, Monica Dumitrescu & Karl Svozil. Experimental evidence of quantum randomness incomputability. *Physical Review A*, 82, 022102, 2010.

[18] Edwin Beggs, José Félix Costa, Diogo Poças, and John V. Tucker. Oracles that measure thresholds: the Turing machine and the broken balance. *Journal of Logic and Computation*, 23(6):1155–1181, 2013.

[19] Ernest Rutherford and Frederick Soddy, The Cause and Nature of Radioactivity, Part I, *Philosophical Magazine 4*, 370-96, 1902.

[20] Fritz Bosch, Measurement of Mass and Beta-Lifetime of Stored Exotic Nuclei, The Euroschool Lectures on Physics with Exotic Beams, Vol. I, *Lect. Notes Phys. 651*, 137-168, Springer, 2004.

[21] E. T. Jaynes and Oscar Kempthorne, Confidence intervals vs Bayesian intervals. In W. L. Harper and C. A. Hooker (eds.) *Foundations of Probability Theory, Statistical Inference, and Statistical Theories of Science*, pages 175-257, Springer, 1976.

[22] B.V. Sreekantan, Searches for Proton Decay and Superheavy Magnetic Monopoles, *J. Astrophys. Astr. 5*, 251-271, 1984.

[23] K. Abe et al. (Super-Kamiokande Collaboration), Search for proton decay via $p \rightarrow \nu K^+$ using 260 kiloton.year data of Super-Kamiokande, *Phys. Rev. D 90*, 072005 - 14 October 2014

[24] Edwin Beggs, Pedro Cortez, José Félix Costa, Bruno Loff, and John V. Tucker, A Hierarchy for $BPP//\log\star$ Based on Counting Calls to an Oracle, in *Emergent Computation, A Festschrift for Selim G. Akl*, Editor Andrew Adamatzky, Emergence, Complexity and Computation Volume 24, Springer, 2017.

[25] Olivier Bournez and Michel Cosnard. On the computational power of dynamical systems and hybrid systems. *Theoretical Computer Science*, 168(2):417–459, 1996.

[26] Pascal Koiran, Michel Cosnard, and Max Garzon. Computability with low-dimensional dynamical systems. *Theoretical Computer Science*, 132:113–128, 1994.

[27] Hava T. Siegelmann. *Neural Networks and Analog Computation: Beyond the Turing Limit*. Birkhäuser, 1999.

[28] Tânia Ambaram, Edwin Beggs, José Félix Costa, Diogo Poças, and John V. Tucker. An analogue-digital model of computation: Turing machines with physical oracles. *Advances in Unconventional Computing*, Springer, 2016.

[29] Edwin Beggs, José Félix Costa, and John V. Tucker. Limits to measurement in experiments governed by algorithms, *Mathematical Structures in Computer Science*, 20(06):1019–1050, 2010.

[30] Kevin T. Kelly. *The Logic of Reliable Inquiry (Logic and Computation in Philosophy)*, Oxford University Press, January 1996.

[31] K. De Leeuw, E.F. Moore, C.E. Shannon, N. Shapiro. Computability by probabilistic machines. In *Automata studies*, Annals of Mathematics Studies 34, pages 183–212. Princeton University Press, 1956.

[32] J. Gill. Computational Complexity of Probabilistic Turing Machines. *SIAM Journal on Computing*, 6(4):675-695, 1977.

[33] Piergiorgio Odifreddi (Editor), *Kreiseliana: About and Around Georg Kreisel*, A.K. Peters / CRC Press, 1996.

[34] Olivier Bournez, How much can analog and hybrid systems be proved (super-)Turing?, *Applied Mathematics and Computation, 178:1*, 58-71, 2006.

[35] Edwin Beggs and John V. Tucker., Can Newtonian systems, bounded in space, time, mass and energy compute all functions? *Theoretical Computer Science 371:1*, 4-19, 2007.

[36] Martin Ziegler, Physically-relativized Church-Turing Hypotheses, *Applied Mathematics and Computation 215:4*, 1431-1447, 2009.

[37] Edwin J. Beggs, José Félix Costa, Diogo Poças, and John V. Tucker. An analogue-digital Church-Turing thesis. *International Journal of Foundations of Computer Science*, 25(4):373–390, 2014.

[38] Edwin Beggs, José Félix Costa, and John V. Tucker. Computational models of measurement and Hempel's axiomatization. In A. Carsetti, editor, *Causality, Meaningful Complexity and Embodied Cognition*, volume 46 of *Theory and Decision Library A*, pages 155–183, Springer, 2010.

[39] Edwin J. Beggs, José Félix Costa, and John V. Tucker. Three forms of physical measurement and their computability. *The Review of Symbolic Logic*, 7 (12):618–646, 2014.

[40] Joe Kilian and Hava T. Siegelmann. The dynamic universality of sigmoidal neural networks. *Information and Computation*, 128(1):48–56, 1996.

[41] José Félix Costa and Raimundo Leong. The ARNN model relativises $P = NP$ and $P \neq NP$. *Theoretical Computer Science*, 499(1):2–22, 2013.

[42] Martin Davis. The myth of hypercomputation. In Christof Teuscher (Editor) *Alan Turing: the life and legacy of a great thinker*, pages 195–212, Springer, 2006.

[43] Martin Davis. Why there is no such discipline as hypercomputation. *Applied Mathematics and Computation*, 178(1):4–7, 2006.

[44] Arthur Steven Younger, Emmett Redd, and Hava T. Siegelmann. Development of physical super-turing analog hardware. In O. H. Obara et. al. (editors) *Unconventional Computation and Natural Computation - 13th International Conference (UCNC 2014)*, Lecture Notes in Computer Science Volume 8553, pages 379–391, Springer, 2014.