Oracles that measure thresholds: the Turing machine and the broken balance

EDWIN BEGGS, *Department of Mathematics, College of Science, Swansea University, Singleton Park, Swansea, SA2 8PP, Wales, UK.*
E-mail: e.j.beggs@swansea.ac.uk

JOSÉ FÉLIX COSTA, *Instituto Superior Técnico, Universidade Técnica de Lisboa, Portugal; Centro de Matemática e Aplicações Fundamentais do Complexo Interdisciplinar da Universidade de Lisboa; Centro de Filosofia das Ciências da Universidade de Lisboa*
E-mail: fgc@math.ist.utl.pt

DIOGO POÇAS, *Instituto Superior Técnico, Universidade Técnica de Lisboa, Portugal; Centro de Matemática e Aplicações Fundamentais do Complexo Interdisciplinar da Universidade de Lisboa; Centro de Filosofia das Ciências da Universidade de Lisboa*
E-mail: diogopocas1991@gmail.com

JOHN V. TUCKER, *Department of Computer Science, College of Science, Swansea University, Singleton Park, Swansea, SA2 8PP, Wales, UK.*
E-mail: j.v.tucker@swansea.ac.uk

Abstract

What can algorithms compute with the help of information provided by an oracle that is a physical system? We have developed a theory that combines Turing machines with experiments that perform physical measurements in which queries are governed by subtle timing protocols and provide the equipment with numerical data with (i) infinite precision, (ii) finite but unbounded precision or (iii) finite but fixed precision. Here, we consider the measurement of physical quantities that are *thresholds*, whose values are obtained by a sequence of approximate measurements that converge either from above or from below. The thresholds may be authentic physical properties or artefacts of the methods and equipment that performs the measurement. Using a canonical example of a threshold oracle, the broken beam balance for measuring mass, we develop methods to cope with thresholds and classify the computational power in polynomial time of this physical oracle using non-uniform complexity classes. Surprisingly, new complexity classes arise illuminating the influence of the operation of the equipment. All classes break the Turing Barrier.

Keywords: Physical oracle, measurement, non-uniform complexity.

1 Introduction

Computation and measurement are intimately connected in all sorts of ways. The computations we make involve data that commonly come from measurements in the world's work. The nature of measurement is the subject of a large and fascinating technical subject, created by

Published online September 18, 2013 doi:10.1093/logcom/ext047

philosophical problems of the physical and behavioural sciences. The theory explains how numerical representations of qualitative attributes are possible. This theory of measurement begins in the nineteenth century, in [17], and acquires a logical basis early on in the twentieth century, making it mathematical, axiomatic and abstract. A synthesis of different approaches, made in [23], led to a coherent mathematical representation theory of measurement,[1] using the methods of mathematical logic (see, e.g. [15, 16, 18]). The theory is laid out in the magnum opus [20]. An interesting survey of the challenges facing the theory is [21].

However, the theory of measurement does not involve computation. We are developing a new theory that combines measurement and computation, in a series of papers (see [2–9, 11, 12]). The theory views the process of making a measurement from an algorithmic perspective and, further, examines what can be computed with the data obtained from measurement. At the heart of our theory is the idea that an experimenter measures a physical quantity by applying an experimental procedure to equipment and that an experimental procedure is an algorithm of some kind. We model this idea as follows:

The experimental procedure is modelled as a Turing machine. The equipment is modelled as a physical device whose behaviour is governed by a physical theory. The equipment is connected to the Turing machine as a physical oracle.

The Turing machine abstracts the experimental procedure, encoding the experimental actions as a programme; it also is able to process the data obtained by measurement. This mathematical approach accommodates

(i) the measurement process;
(ii) the use the data from measurement in subsequent computations; and, indeed,
(iii) arbitrary sequences of interactions with equipment.

Some implications of this computational model for the axiomatic theory have been considered in [7].[2]

The development of the theory has been shaped by the careful study of case studies: various physical experiments involving statics, dynamics, electricity, optics, atomic theory. The standard oracle to a Turing machine is simply a set that contains information to boost the power and efficiency of computation: a query is a question about set membership that is answered in one time step. However, physical oracles require queries based upon rational numbers (specifically dyadic rationals denoted by finite binary strings) that initialize the equipment. The initialization may be imprecise so errors must be considered. The behaviour of the equipment determines the time taken to answer the queries, which is very rarely constant and usually depends upon the query data, so timing must be considered. Recalling our formative case studies, we note that the measurement of distance taught us that oracles involve information with possible error (such an experiment is fully analysed in [2, 5, 12]). The measurement of a mass taught us that oracles may take considerable time to consult (such an experiment is fully analysed in [8, 12]). Actually, an important difference between the classical oracle and the physical oracle is the need of a timer as a component of the Turing machine, which is a cost function T of the size of the query.

To measure the value of a physical quantity, i.e. a real number y, the experimenter (= the Turing machine) proceeds to construct approximations, which are generated by *oracle consultations*. Whenever possible, a measurement procedure should approximate the unknown quantity from above and from below, generating a series of experimental values that converges to the quantity. We call

[1] Of extensive quantities.
[2] Scientific activity based upon Turing machines can be found in computational learning theory (see [19].)

such experiments *two-sided*. Two-sided measurement is the focus of our previous work (see [9, 12]), and that of axiomatic measurement theory (see [15, 16, 18]).

However, not all measurements can be made this way. Some quantities by their physical nature, or by the nature of the equipment used to measure them, are *thresholds that can only be approximated either just from below or just from above*. Examples are experiments on activation thresholds for the neurone and Rutherford scattering. In this article,[3] we study *threshold experiments*, which are complex and are not yet addressed in the literature. We show that threshold experiments are oracles of a new kind and turn out to have different properties from the two-sided experiments. The results about two-sided oracles do not apply to these systems.

In Section 2, we discuss in some detail four threshold experiments in order to establish the significance of this class of measurements. In Section 3, we introduce the protocols needed to interface a Turing machine and threshold physical oracle. In Section 4, we focus on the *broken beam balance* for measuring (mass, and use BBE for Broken Balance Experiment for short). This we take as the simplest canonical example of a threshold oracle, the threshold being an artefact of the equipment rather than of the physical quantity. Sections 5 and 6 contain properties of the broken beam balance in preparation for the main results. In Sections 7 and 8, we prove lower and upper bounds on the computational power of polynomial-time Turing machines with the broken beam balance. The new theorems reveal differences with the less complex two-sided case:

THEOREM 1.1

(1) If a set A is decided by an oracle Turing machine coupled with a BBE machine of infinite precision, then $A \in P/\log^2\star$.[4] If a set A is in $P/\log\star$, then A is decided by a oracle Turing machine coupled with a threshold oracle of infinite precision.
(2) If a set A is decided by an oracle Turing machine coupled with a BBE of unbounded or fixed precision, then $A \in BPP//\log^2\star$.[5] If a set A is in $BPP//\log\star$, then A is decided by a oracle Turing machine coupled with a threshold oracle.

The upper bound known so far for the two-sided oracles with non-infinite precision is $BPP//poly$ (except for a particular type of two-sided oracles considered in [5] and [11] for which is P/poly and $BPP//\log\star$, respectively). The new results raise questions about the stability of the complexity classes that arise when using physical oracles, which we address in Section 9 of concluding remarks.

2 Threshold experiments

We will begin by listing some examples of threshold experiments[6] and then we will focus on one particular experiment, the *broken balance experiment*.

[3] A short account of the threshold oracles appeared in a conference paper (see [13]).

[4] Let $\mathcal{B}$ be a class of sets and $\mathcal{F}$ a class of functions. The advice class $\mathcal{B}/\mathcal{F}$ is the class of sets A for which there exists $B \in \mathcal{B}$ and some $f \in \mathcal{F}$ such that, for every word w, $w \in A$ if and only if $\langle w, f(|w|)\rangle \in B$. For the prefix advice class $\mathcal{B}/\mathcal{F}\star$ some (prefix) function $f \in \mathcal{F}$ must exist such that, for all words w of length less or equal to n, $w \in A$ if and only if $\langle w, f(n)\rangle \in B$.

[5] $BPP//\mathcal{F}\star$ is the class of sets A for which a probabilistic Turing machine $\mathcal{M}$, a prefix function $f \in \mathcal{F}\star$, and a constant $\gamma < \frac{1}{2}$ exist such that, for every length n and input w with $|w| \leq n$, $\mathcal{M}$ 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 $w \notin A$. Moreover, we use $\log^2\star$ to denote the class of advice functions such that $|f(n)| \in \mathcal{O}((\log(n))^2)$.

[6] Some of these examples have been already introduced in [10, 13].

2.1 The squid giant motor neuron

We may think of the *neuron* as a cell with three regions: the *dendritic region*, the cell body or *soma* and the *axon*. The dendrites receive electric signals from other neurons depolarizing the *membrane* of the cell. This process occurs due to a leak of sodium and potassium ions across membrane channels. When the membrane potential reaches a threshold value, an increase in voltage across the membrane occurs that propagates along the axon.

The Spinal Neuron Experiment is designed to measure the threshold of firing of a given neuron. This first threshold experiment is inspired in the spiking neuron (see [1]), such like the squid giant motor neuron, and it is designed to measure the threshold of activation: an electric current ι is injected into the cell and the action potential, once generated, can be detected along the axon. Suppose that the rest (membrane) potential is ν_0 ($\nu_0 \sim -65mV$) and that the threshold electric current is ι_0 ($\iota_0 \sim 2nA$). The goal is to measure the threshold ι_0, for some concentration of the ions: (a) if $\iota < \iota_0$, then no signal is sent along the axon and (b) if $\iota \geq \iota_0$, a series of action potentials is propagated along the axon. Once the current is switched off, the rest potential is reset.

2.2 The Photoelectric Effect Experiment

The equipment consists of a metallic surface, a source of monochromatic light and an electron detector (see Figure 1). Each photon of the light beam has energy $E = hf$, where h is the Planck constant and f is the frequency of the light. On the other hand, the metallic surface is characterized by a value $\phi = hf_0$ of energy, where f_0 is the minimum (threshold) frequency required for photoelectric emission. The goal is to measure f_0, and to that end we can send a light beam with frequency f: (a) if $f \leq f_0$, then no electron escapes the surface and (b) if $f > f_0$, then the electrons are ejected with kinetic energy $E = h(f - f_0)$. In this way, the photoelectric experiment is a threshold experiment, since we only get a response whenever the light beam frequency exceeds the threshold frequency.

2.3 The BBE

The experiment[7] consists of a balance scale with two pans (see Figure 2). In the right pan we have some body with an unknown mass y. To measure y we place test masses z on the left pan of the balance: (a) if $z < y$, then the scale will not move since the rigid block prevents the right pan from moving down; (b) if $z > y$, then the left pan of the scale will move down, which will be detected in some way; and (c) if $z = y$, then we assume that the scale will not move since it is in equilibrium. We assume the following characteristics of the apparatus inter alia: (a) y is a real number in $[0, 1]$, (b) the mass z can be set to any dyadic rational in the interval $[0, 1]$, (c) a pressure-sensitive stick is placed below the left side of the balance, such that, when the left pan touches the pressure-sensitive stick, it reacts producing a signal, (d) the mass z can be set so that the procedure starts from absolute rest, (e) the friction between the masses and the pans is large enough so that these will not slide away from their original position once the scale is in motion, and (f) the bar on which the masses are placed is made of an homogeneous material, so that the two pans have exactly the same weight. Assuming that the test mass weighs z and the unknown mass weighs y, the cost of the experiment, $T_{exp}(z, y)$,

[7] Suggested by Manuel João Morais, from Instituto Superior Técnico.

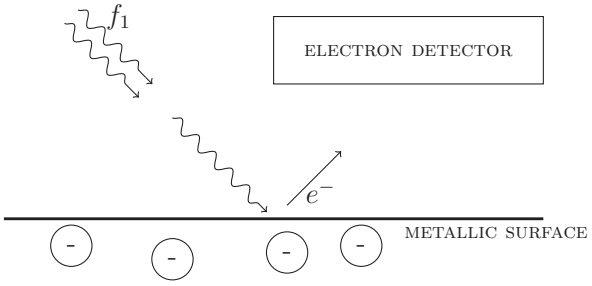

FIGURE 1. Schematic description of the Photoelectric Effect Experiment.

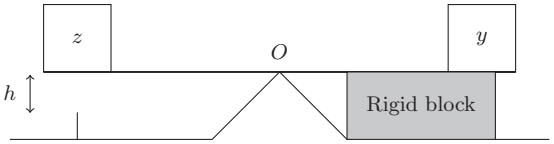

FIGURE 2. Schematic representation of BBE.

which is the time taken for the left pan of the balance to touch the pressure stick, is given by:[8]

$$T_{\text{exp}}(z,y) = \text{Const} \times \sqrt{\frac{z+y}{\max(0,z-y)}} \text{ for all } y,z \in \mathbb{R}. \tag{1}$$

2.4 *Rutherford scattering in an electric field*

Our new experiment consists of scattering a beam of electrical charged particles in a Coulombian field, as in Figure 3. This Rutherford experiment will stand in this article as an extended detailed example of an abstract description of a real physical experiment. In this case, the basic qualitative procedure, in the sense of Hempel (see [18]), is not two sided but a threshold one as the broken balance, i.e. in this case, the unknown value can only be approximated from below.

Collisions described here involve projecting particles towards other particles that are at rest in the system of the laboratory. Let y denote the mass of the particles at rest in the system of the laboratory and z denote the mass of the impinging proof particles (which are dyadic rational numbers in some system of units).

The problem that we will solve first is how to transform the *cross section* $\chi(\theta'_z)$, computed in the system of the center of mass, shown in Figure 4, back into the system of the laboratory. To do this, we have to transform the deflection angles θ'_z of the scattered particles, measured in the reference system of the center of mass, in deflection angles θ_z in the system of the laboratory (see Figure 5). *Deflection angles are measured from the direction of the impinging beam to the right (negative) or to the left (positive).* As a result, in the system of the laboratory, we have:

(1) Suppose that $z < y$, i.e. the bombarding particle is lighter than the particle that is being struck. Then the maximum angle of scattering θ_z is always π for both sides.

[8]This expression for the time, namely exhibiting an exponential growth on the precision of z with respect to the unknown y, is typical in physical experiments, regardless of the concept being measured.

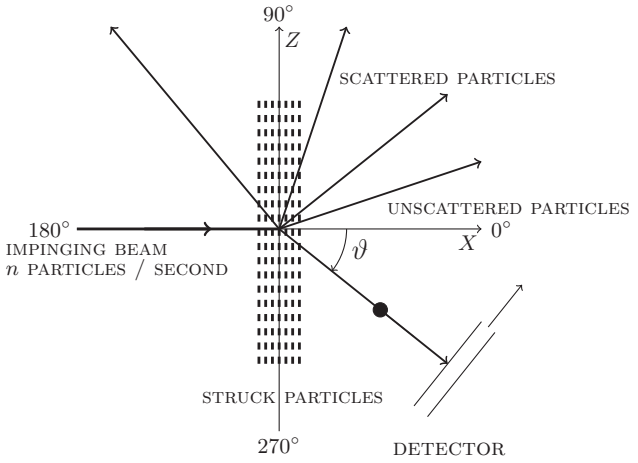

FIGURE 3. Scattering machine experiment. The detector has a non-negligible area and rotates around the block of matter.

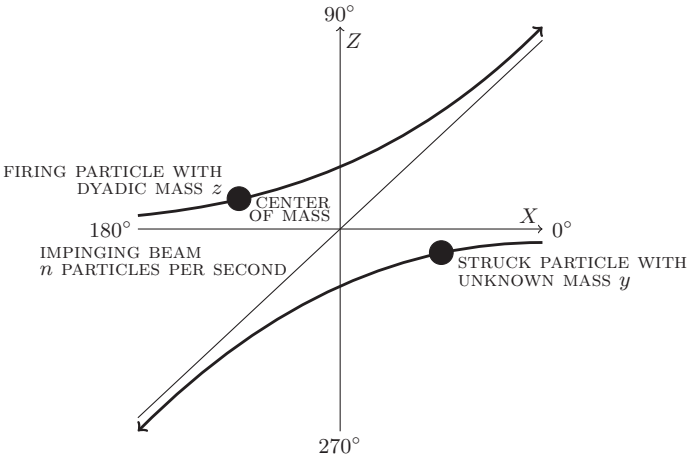

FIGURE 4. Elements of trajectory of an electric particle in a Coulombian field.

(2) Suppose that $z > y$. In this case we can easily see that the maximum of θ_z is less than $\pi/2$ for both sides.
(3) Suppose that $z = y$. The maximum of θ_z is $\pi/2$ for both sides.

There is an abrupt discontinuity in the cross-section, since for $z = y$ the maximum value of θ_z is $\pi/2$, while for z very slightly less than y, the maximum value of θ_z suddenly jumps to π. This is not a real physical discontinuity because the cross section itself, $\chi(\theta_z)$, approaches zero for angles greater than $\pi/2$, as z approaches y.

The non-normalized cross section when z is relatively close to y is given by:

$$\chi(\theta_z) = \frac{\pi(q_y q_z)^2}{4E^2} \frac{\cos(\theta_z/2)}{\sin^3(\theta_z/2)}, \tag{2}$$

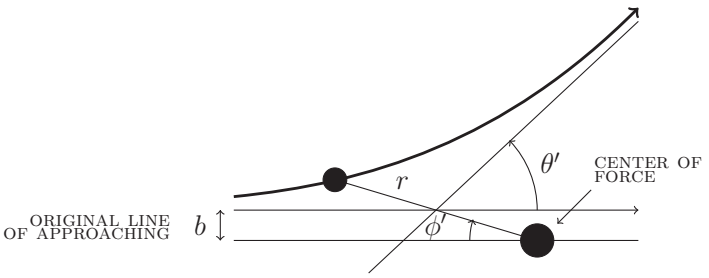

FIGURE 5. Scattering a particle: view from the system of reference of the centre of mass. When the particle of mass z is fired, the particle of unknown mass y is at rest in the system of reference of the laboratory. The angle of deflection is θ'.

where q_y and q_z are the electric charges of the struck and projected particles, respectively, and E is the kinetic energy of each particle of the beam. We compute next the total cross-section for the scattering between the angles $\cos^{-1}(-z/y)$ and $2\pi - \cos^{-1}(-z/y)$, corresponding, in the system of the laboratory, to the angles $\frac{\pi}{2}$ and $\frac{3}{2}\pi$.

Since z and y are very close, it means that $\cos^{-1}(-z/y) \sim \pi$. Hence, after several calculations, we obtain the form:

$$\chi(\theta_z) = \frac{\pi^2 (q_y q_z)^2}{16 y E^2} \, |y - z| \, . \tag{3}$$

The number of particles collected in the angular section $[\frac{\pi}{2}, \frac{3\pi}{2}]$ after time t is then given by the product of the total cross-section and the number of particles projected by the beam, that is $N_0 t$, where N_0 is the number of particles projected per unit time. A given threshold of detection $N_0\uparrow$ is reached after time t, given by:

$$N\uparrow = t \, N_0 \, \frac{\pi^2 (q_y q_z)^2}{16 y E^2} \, |y - z| \, . \tag{4}$$

The time taken for detection of particles of mass $z < y$ is then inversely proportional to the difference of masses, i.e.

$$t = \frac{16 y E^2 N\uparrow}{\pi^2 N_0 (q_y q_z)^2} \, \frac{1}{|y - z|} \, . \tag{5}$$

The experimental algorithm is of a different kind of those considered in [2, 7, 8]. Let $T : \mathbb{N} \to \mathbb{N}$ be the time for the experiment to take place as a (total) function of the size of the sequence of bits setting the value of the mass of the bombarding particles. The function T can be seen as a *schedule*, i.e. in order to perform the experiment with proof particles of mass z, $T(|z|)$ gives the amount of time steps that *the experimenter accepts to wait* until resuming the experimental conditions.

This new Scatter Machine Experiment (SME for short) works as an oracle to a Turing machine in the same way we described in [2, 6, 8, 9, 11]. The Turing machine is connected to the SME in the same way as it would be connected to an oracle: we replace the query state with a *scattering state* (q_s), the 'YES' state with a *less than* state (denoted also by q_{YES}), and the 'NO' state with a 'TIMEOUT' state (denoted by q_{TIMEOUT}). The resulting computational device is called the *analogue-digital scattering machine*, and we refer to the *unknown mass* of the struck particles of an analogue-digital scattering machine when meant to discuss the unknown mass of the corresponding SME. After setting the mass

z, the SME will fire particles of mass z, wait $T(|z|)$ time units, and then check if some particles have been detected in the angular section $(\frac{\pi}{2}, \frac{3\pi}{2})$; in this case the Turing machine computation will be resumed in the state q_{YES}. If no particles have been detected in the angular section $[\frac{\pi}{2}, \frac{3\pi}{2}]$, i.e. if the threshold of detection has not been attained, then the Turing machine computation will be resumed in the state q_{TIMEOUT}.

The unknown mass can encode for an advice to the Turing machine as in [2, 8]. In those cases, a schedule T can be designed to read the digits of the unknown mass in a two-sided experiment, where values to be measured can be approximated both from below and from above. But our current experiment is one sided: we cannot distinguish between 'TIMEOUT' due to the fact that y and z became close each other, being $z < y$, and 'NO DETECTION' due to the fact that $z > y$. Note that in both cases we always have particles in the interval $[-\frac{\pi}{2}, \frac{\pi}{2}]$.

Changing the mass of the particles in the beam also changes their electrical charge, assuming that electrical charge is uniformly distributed, but changing the charge does not disturb the qualitative answer the machine gets from the experimental apparatus, namely when dealing with masses z very close to the value of y. Fundamental measurement (see [18]) is based on *events* happening during the experiment and not on values taken meanwhile (in a derived measurement[9]).

3 Protocols

3.1 Query word and precision

A larger variety of experiments could have been mentioned earlier (such as Rutherford's scattering experiment). However, since the BBE is fairly simple to analyse and understand, and as it displays the properties of threshold experiments, we will focus on it. Just as in previous investigations (see, e.g. [2, 8, 11]), we will consider different types of precision, i.e. different communication protocols between the experimenter/Turing machine and the oracle/analogue device. The query word $z \in \{0,1\}^{|z|}$ of length $|z|$ is converted to a dyadic rational in $[0, 1)$ by taking zero point (the query word) in binary notation. There are three cases:

(a) *infinite precision*: when the dyadic z is read in the query tape, a test mass $z' = z$ is simultaneously placed in the left pan.
(b) *unbounded precision*: when the dyadic z is read in the query tape, a test mass z' is simultaneously placed in the left pan such that $z - 2^{-|z|} \leq z' \leq z + 2^{-|z|}$. Here $z' \in \mathbb{R}$ is independently and uniformly distributed in the interval.
(c) *fixed precision* $\epsilon > 0$: when the dyadic z is read in the query tape, a test mass z' is simultaneously placed in the left pan such that $z - \epsilon \leq z' \leq z + \epsilon$. Here $z' \in \mathbb{R}$ is independently and uniformly distributed in the interval.

In what follows the suffix operation $\lfloor_n$ on a word w, $w\lfloor_n$, denotes the prefix of size n of the ω-word $w0^\omega$, no matter the size of w. Mass$(m\lfloor_\ell)$ denotes the action that triggers the BBE experiment with mass (query word) $m\lfloor_\ell$. Depending on the context, the experiment is performed either with infinite, unbounded or finite precision, as explained above. For the remainder of this section, we write $\mathcal{M}$ for any analogue-digital machine using either infinite precision or unbounded precision or fixed precision ϵ. We can also refer to $\mathcal{M}$ as the corresponding oracle Turing machine.

[9] Although, in [20], it is questioned if fundamental measurement is indeed fundamental.

3.2 The time schedule

To the oracle Turing machine model $\mathcal{M}$ we associate a schedule $T:\mathbb{N}\to\mathbb{N}$. On submitting the query z, the Turing machine waits a time $T(|z|)$, and then receives the answer to the query. By default, if no other answer is provided, the answer 'TIMEOUT' is returned. We suppose that $T(\ell)$ is a time constructible function, i.e. that the Turing machine can itself count its own waiting time, a condition we might call busy waiting.

The threshold oracles have answers 'YES' or 'TIMEOUT', resulting in a transition of $\mathcal{M}$ to the state q_{YES} or q_{TIMEOUT}, respectively. For $y\in(0,1)$, the broken balance experiment BBE with unknown mass y is characterized by following property: For a test mass $z'\in[0,1)$ the experiment, having z' approaching y from above, takes a time $T_{exp}(z',y)$, which we assume is constrained by the inequality, for $c,d>0$ constants

$$\frac{d}{\sqrt{z'-y}}<T_{exp}(z',y)<\frac{c}{\sqrt{z'-y}}.$$

If the experiment completes (by touching the pressure sensitive stick), then $z'>y$ and the outcome of the experiment is $\text{Mass}(z)=\text{YES}$. If the experiment does not complete (i.e. the experimental time $T_{exp}(z',y)$ exceeds the time schedule $T(|z|)$), the outcome of the experiment is $\text{Mass}(z)=\text{TIMEOUT}$. We now give the procedure 'Mass' for the BBE for some unknown mass y and some time schedule T. It comes in three cases, for infinite, unbounded and finite precision.

PROTOCOL IP: "MASS": Infinite Precision Case
Receive as input the description of a dyadic rational z (possibly padded with 0s); Place a mass z in the left pan; Wait $T(\|z\|)$ units of time; Check if the pressure stick has sent a signal. If so, return 'YES', otherwise 'TIMEOUT'.

PROTOCOL UP: "MASS": Unbounded Precision Case
Receive as input the description of a dyadic rational z (possibly padded with 0s); Place a mass z' in the left pan, where $z'\in(z-2^{-\|z\|},z+2^{-\|z\|})$; Wait $T(\|z\|)$ units of time; Check if the pressure stick has sent a signal. If so, return 'YES', otherwise 'TIMEOUT'.

PROTOCOL FP: "MASS": Finite Precision Case(ϵ)
Receive as input the description of a dyadic rational z (possibly padded with 0s); Place a mass z' in the left pan, where $z'\in(z-\epsilon,z+\epsilon)$; Wait $T(\|z\|)$ units of time; Check if the pressure stick has sent a signal. If so, return 'YES', otherwise 'TIMEOUT'.

3.3 A non-uniform complexity class and boundary numbers

We start by defining a new non-uniform complexity class that did not appear in our previous papers, namely [8, 12]. Observe that $\log^2\star$ is the class of prefix functions f such that $|f(n)|\in\mathcal{O}((\log(n))^2)$.

DEFINITION 3.1
$BPP//\log^2\star$ is the class of sets A for which there exist a probabilistic Turing machine $\mathcal{M}$ with polynomial running time, a function $f\in\log^2\star$ and a constant $\gamma<1/2$ such that, for every $n\in\mathbb{N}$ and

for every word w such that $|w| \leq n$, (a) if $w \in A$, then $\mathcal{M}$ rejects $\langle w, f(n)\rangle$ with probability at most γ and (b) if $w \notin A$, then $\mathcal{M}$ accepts $\langle w, f(n)\rangle$ with probability at most γ.

Previously, our results involved the prefix functions such that $|f(n)| \in \mathcal{O}(\log(n))$. Before we go into the details of how this new class is obtained, we should explain why the difference occurs. Previously, the experiment was to measure a physical quantity y, and the involvement of the time schedule $T(\ell)$ was minimal—it determined the time needed for a sufficient number of experiments to be able to find y to a specified accuracy. It was not necessary to have precise knowledge of the time schedule—if one would suffice, then so would any larger one.

This is not the case with the new classes, there $T(\ell)$ is involved with the experiment to a much greater degree. The experiment is not run to find y, its value is almost irrelevant. For a given k, let z_k be the real solution to $T_{exp}(z_k, y) = T(k)$. Then for protocol time $T(k)$, if $z \leq z_k$ the result is TIMEOUT and for $z > z_k$ the result is YES. It is these boundary numbers z_k, or their truncations, which are used in the algorithm. We expect a unique such boundary number, because if a test mass z tips the balance in a time $T(k)$, then so should any greater test mass.

To consider the difference in philosophy between the two approaches, consider two teams of experimenters. They are both given identical masses, and are told to find the value of the mass. They both produce experimental apparatus to do this. They write out a time schedule, based on how long they expect the experiment to take to reach a given accuracy. Now both of these experimental groups can solve the class of problems $P/\log\star$ with advice given by the fixed mass (assuming, for simplicity, that we are in the infinite precision case), if both their time schedules are exponential in the query word length.

However, these two teams would almost certainly not be able to replicate results on the class $BPP//\log^2\star$. There would be no reason why their boundary numbers would coincide. They might have used completely different experimental methods to determine the mass. Even if they had both used a broken balance machine, the masses of the arms in the machine might be different. Even if the apparatus seemed identical, the acceleration due to gravity g might be different in their locations. But surely they could write out time schedules so that their boundary numbers would coincide? Probably not—that would require a very large amount of knowledge about the physical system, probably including the *a priori* unknown value of the given mass, and there are only countably many descriptions of time constructible functions that could be used. Only *absolutely identical* experiments would be reasonably expected to give the same boundary numbers.

So how 'real' are the physical oracles giving the class $BPP//\log^2\star$? Of course, by specifying given forms for the experimental time, examples of experiments could be given with specified boundary value behaviour. This may be allowed by the rules of thought experiments, but in terms of what is 'practically' measurable with physical theories it is an interesting question.

4 The BBE machine as a means to measure real numbers

4.1 Measurement in the infinite and unbounded precision cases

The algorithm Binary Search measures a mass, in the cases of infinite or unbounded precision. This is an improvement to an algorithm previously presented in [10]. The experimental procedure Mass is either deterministic (for the infinite precision case) or stochastic (for the unbounded precision case) and takes the scheduled time $T(\ell)$, where ℓ is the size of the query and T an arbitrary time constructible function.

ALGORITHM 'BINARY SEARCH'

Input number $\ell \in \mathbb{N}$; `% Number of places to the right of the left leading 0`
  $x_0 := 0$; $m := 0$, $x_1 := 1$;
  While $x_1 - x_0 > 2^{-\ell}$ **Do Begin**
    $m := (x_0 + x_1)/2$;
    $s :=$ Mass$(m\downarrow_\ell)$; `% Procedure Mass takes time` $T(\ell)$
    If $s =$ 'YES' **Then** $x_1 := m$ **Else** $x_0 := m$;
  End While;
Output x_0.

PROPOSITION 4.1
Let s be the result of Mass(m), for an unknown mass y and time schedule T. In the infinite precision scenario,

(a) if $s =$ 'YES', then $y < m$ and
(b) if $s =$ 'TIMEOUT', then $y > m - (c/T(|m|))^2$.

In the unbounded precision scenario,

(a) if $s =$ 'YES', then $y < m + 2^{-|m|}$ and
(b) if $s =$ 'TIMEOUT', then $y > m - 2^{-|m|} - (c/T(|m|))^2$.

PROPOSITION 4.2
For any unknown mass y and any time schedule T,

(a) the time complexity of the algorithm 'Binary Search', both in the infinite and unbounded precision scenarios, for input ℓ, is $\mathcal{O}(\ell T(\ell))$,
(b) in both cases, for all $k \in \mathbb{N}$, if there exists $\ell \in \mathbb{N}$ such that $\ell \geq k+1$ and $T(\ell) \geq c2^{(k+1)/2}$, then the output is a dyadic rational m such that $|y - m| < 2^{-k}$.

PROOF. In the infinite precision case, all through the algorithm we have an interval $[x_0, x_1]$ where x_0 gives TIMEOUT and x_1 gives YES. At the end of the algorithm, $x_1 = x_0 + 2^{-\ell}$, so from Proposition 4.1 we have

$$x_0 + \frac{1}{2^\ell} > y > x_0 - \left(\frac{c}{T(\ell)}\right)^2.$$

In the unbounded precision case, all through the algorithm we have an interval $[x_0, x_1]$ where x_0 can give TIMEOUT and x_1 can give YES. At the end of the algorithm, $x_1 = x_0 + 2^{-\ell}$, so from Proposition 4.1 we have

$$x_0 + \frac{2}{2^\ell} > y > x_0 - \left(\frac{c}{T(\ell)}\right)^2 - \frac{1}{2^\ell}.$$

To ensure $|y - x_0| < 2^{-k}$ it suffices to have $\ell \geq k+1$ and $T(\ell) \geq c2^{(k+1)/2}$. ■

4.2 *Measurement in the finite precision case*

We consider now the fixed precision ϵ. Instead of setting the unknown mass with the value μ, we reassign it the modified value $y = 1/2 + \epsilon - 2\mu\epsilon$ in a way such that $1/2 - \epsilon \leq y \leq 1/2 + \epsilon$. Fixing a

time schedule T, we call the procedure Mass($1\downarrow_\ell$), where ℓ is a natural number, i.e. we call the test mass $z=0.5$ on the left pan and wait $T(\ell)$ units of time. The experiment will be performed with mass $z' \in (1/2-\epsilon, 1/2+\epsilon)$. We can split this interval in two: (a) an interval $(1/2-\epsilon, y+\varphi)$ in which the result of the experiment is always 'TIMEOUT' and (b) an interval $(y+\varphi, 1/2+\epsilon)$ in which the result of the experiment is always 'YES', where φ is such that $T_{exp}(y+\varphi, y)=T(\ell)$, from where we deduce that $\varphi \leq (c/T(\ell))^2$, for some constant c. We will guess the digits of μ (and, consequently, the digits of y) by calling Mass($1\downarrow_\ell$) a number ζ of times. Each time the procedure is called we get an answer: (a) 'YES' with probability $p=(1/2+\epsilon-y-\varphi)/2\epsilon=\mu-\varphi/2\epsilon$ and (b) 'TIMEOUT' with probability $1-p=1-\mu+\varphi/2\epsilon$. The number α of 'YES's is a random variable with binomial distribution. Consider the estimator $X=\alpha/\zeta$. We have the expectation or mean of X, $\mathbb{E}[X]=\mathbb{E}[\alpha]/\zeta=\mu-\varphi/(2\epsilon)$. Then the variance of X is $\mathbb{V}[X]=\mathbb{V}[\alpha]/\zeta^2 \leq 1/(4\zeta)$. We use Chebyshev's inequality, to conclude that, for every $\Delta > 0$,

$$P(|X-\mathbb{E}[X]| > \Delta) \leq \frac{\mathbb{V}[X]}{\Delta^2} \leq \frac{1}{4\zeta\,\Delta^2}.$$

We take ℓ such that $T(\ell) > c2^{(k+1)/2}/\sqrt{2\epsilon}$ to ensure that $\varphi/(2\epsilon) < 1/2^{k+1}$. Thus, if $|X-\mu| > 1/2^k$, then $|X-(\mu-\varphi/(2\epsilon))| > 1/2^{k+1}$. The probability of error is then

$$P\left(|X-\mu| > \frac{1}{2^k}\right) \leq P\left(\left|X-(\mu-\frac{\varphi}{2\epsilon})\right| > \frac{1}{2^{k+1}}\right) \leq \frac{2^{2k}}{\zeta}.$$

We need $\zeta > 2^{2k}/\delta$ in order to bound such a probability by δ. This means that we can specify a BBE machine to read μ by performing a large number of experiments with test mass $1/2$ and returning the relative frequency of those that produced result 'YES'. Furthermore, for any δ and k, we need at least $2^{2k}/\delta$ experiments with scheduled time $T(\ell) > c2^{(k+1)/2}/\sqrt{2\epsilon}$ to obtain an approximation of k bits of μ with an error probability less than δ. This is used to make an algorithm 'Search(ϵ, h)' to measure a mass μ with fixed precision ϵ. The integer number h is used to bound the probability of error.

ALGORITHM 'SEARCH(ϵ, h)'

Input number $\ell \in \mathbb{N}$; `% number of places to the right of the left leading 0`
  $c:=0$; $\zeta := 2^{2\ell+h}$; `% h is used to bound the probability of error`
  Repeat ζ times
    s:=Mass($1\downarrow_\ell$); `% Recall that this step takes T(ℓ) units of time`
    If $s=$ 'YES' **Then** $c:=c+1$;
  End Repeat;
Output c/ζ.

PROPOSITION 4.3
For all $\mu \in (0,1)$, $\epsilon \in (0, 1/2)$, $h \in \mathbb{N}$, and time schedule T,

(a) for all $k \in \mathbb{N}$, if there exists $\ell \in \mathbb{N}$ such that $T(\ell) > c2^{(k+1)/2}/\sqrt{2\epsilon}$ then, with probability of error 2^{-h}, the output of the algorithm is a dyadic rational m such that $|\mu-m| \leq 2^{-k}$, and
(b) the time complexity of algorithm 'Search(ϵ, h)' for input ℓ is $\mathcal{O}(2^{2\ell+h}T(\ell))$.

5 The BBE machine as a means to produce fair coin tosses

In both cases of unbounded and finite precision, the experiment becomes probabilistic and we can use it to simulate independent coin tosses and to produce random strings. This is similar to the procedure in [2].

LEMMA 5.1
For all unknown masses y and all time schedules T there is a dyadic rational z and a real number $\delta \in (0,1)$ such that the result of Mass(z) is a random variable that produces 'YES' with probability δ and 'TIMEOUT' with probability $1-\delta$.

PROOF. Consider first the unbounded precision case. Let ℓ be such that $2^{-\ell} \leq y$ and $T_{exp}(1,y) < T(\ell)$. This means that, if we perform the experiment with test mass $1\lfloor_\ell$ and infinite precision we would obtain the answer of 'YES' within $T(\ell)$ units of time. After fixing the unknown mass y and the schedule T, we know that there exists some value y' such that $T_{exp}(y',y) = T(\ell)$. Thus let z' be the dyadic rational of size ℓ such that $z' \leq y' < z' + 2^{-\ell}$. Observe that $y < y' < 1$ and so $0 < z' < 1$. Consider performing the experiment Mass(z'). We know that the mass placed on the left pan lies on $(z'-2^{-\ell}, z'+2^{-\ell})$. Also, the probability of returning 'YES' is given by

$$\delta = \frac{z'+2^{-\ell}-y'}{z'+2^{-\ell}-(z'-2^{-\ell})} = \frac{1}{2} - \frac{y'-z'}{2\times 2^{-\ell}}$$

and so $0 < \delta < 1$.

For the finite precision case, the proof is similar. Let ℓ be a positive integer such that $2^{-\ell} \leq y$, $2^{-\ell} \leq \epsilon$ and $T_{exp}(1,y) < T(\ell)$. Then there exists y' such that $T_{exp}(y',y) = T(\ell)$. By choosing z' as a dyadic rational of size ℓ such that $z' \leq y' \leq z' + 2^{-\ell}$, we guarantee that the result of experiment Mass(z') is 'YES' with probability $\delta \in (0,1)$. ■

The previous and the next lemma are a variation of those relative to the two-sided oracles (see [2, 5]). After fixing the position of the dyadic rational provided by Lemma 5.1, the results of sequential experiments can be seen as independent biased coin tosses with probability δ of giving result 'YES' and $1-\delta$ of giving result 'TIMEOUT'.

LEMMA 5.2
Take a biased coin with probability of heads $\delta \in (0,1)$ and let $\gamma \in (0,1/2)$. Then, there is an integer N such that, with probability of failure at most γ, we can use a sequence of independent biased coin tosses of length Nn to produce a sequence of length n of independent fair coin tosses.

6 The Cantor set $\mathcal{C}_3$

We denote by $\mathcal{C}_3$ (the *Cantor numbers*) the set of real numbers x such that $x = \sum_{k=1}^{\infty} x_k 2^{-3k}$, where $x_k \in \{1,2,4\}$, i.e. the numbers composed by triples of the form 001, 010, or 100.

PROPOSITION 6.1
For every $x \in \mathcal{C}_3$ and for every dyadic rational $z \in (0,1)$ with size $|z| = m$, (a) if $|x-z| \leq 1/2^{i+5}$, then the binary expansions of x and z coincide in the first i bits and (b) $|x-z| > 1/2^{m+10}$.

PROOF.

(a) First suppose that z and x coincide on the first $i-1$ bits and differ on the ith bit. We have two relevant cases.

$z < x$: In this case $z_i = 0$ and $x_i = 1$. In the worst cases, the binary expansion for z after the i-th position begins with a sequence of 1s and the binary expansion for x after the i-th position begins with a sequence of 0s:

	i	lower bound of $\lvert x-z\rvert$
z	$\cdots 011111\cdots$	
x (case $i \equiv_3 0$)	$\cdots 100100\cdots$	$> 2^{-(i+3)}$
x (case $i \equiv_3 1$)	$\cdots 100001\cdots$	$> 2^{-(i+5)}$
x (case $i \equiv_3 2$)	$\cdots 100010\cdots$	$> 2^{-(i+4)}$

$z > x$: In this case $z_i = 1$ and $x_i = 0$. In the worst cases, the binary expansion for z after the i-th position begins with a sequence of 0s and the binary expansion for x after the i-th position begins with a sequence of 1s:

	i	lower bound of $\lvert x-z\rvert$
z	$\cdots 1000\cdots$	
x (case $i \equiv_3 0$)	$\cdots 0100\cdots$	$> 2^{-(i+2)}$
x (case $i \equiv_3 1$)	$\cdots 0101\cdots$	$> 2^{-(i+2)}$
x (case $i \equiv_3 2$)	$\cdots 0110\cdots$	$> 2^{-(i+3)}$

We conclude that in any case $|x-z| > 2^{-(i+5)}$. Thus, if $|x-z| \leq 2^{-(i+5)}$, then x and z coincide in the first i bits.

(b) Since the binary expansion of z after the mth bit is exclusively composed of 0s and any Cantor number $x \in \mathcal{C}_3$ has at most four consecutive 0s in its binary expansion, we conclude that, in the best fit, z and x can not coincide in the $m+5$th bit. Thus, by (a), $|x-z| > 2^{-(m+10)}$. ■

Now we will encode a given advice function $f : \mathbb{N} \to \{0,1\}^\star$ into a real number in $(0,1)$.

DEFINITION 6.2
The encoding of a word $w \in \Sigma^\star$, over some alphabet Σ, denoted by $c(w)$, is the binary expression of the real number obtained first by converting w to a string of 0's and 1's, and then replacing every 0 by 100 and every 1 by 010. Given a function $f \in \log\star$, we denote the encoding of f by the real number $\mu(f) = \lim \mu(f)(n)$, recursively defined by (a) $\mu(f)(0) = 0 \cdot c(f(0))$, (b) $\mu(f)(n+1) = \mu(f)(n)c(s)$ whenever $f(n+1) = f(n)s$ and $n+1$ is not a power of two, and (c) $\mu(f)(n+1) = \mu(f)(n)c(s)001$, whenever $f(n+1) = f(n)s$ and $n+1$ is a power of two.

The encoding above consists of replacing the bits of f by triples 100 and 010, adding 001 at the end of each code of $f(2^k)$, with $k \in \mathbb{N}$. Observe that, by construction, $\mu(f) \in \mathcal{C}_3$. To obtain $f(2^k)$, we just have to read the bits of $\mu(f)$ in triples until the $(k+1)$-th triple 001 is found. Consequently, one may say that, whenever $f \in \log\star$, by knowing $\mathcal{O}(k)$ bits of the real number $\mu(f)$, we can know $f(2^k)$.

7 Lower bounds on the BBE machine

DEFINITION 7.1
We say that a set A is decidable by an infinite precision BBE machine in polynomial time if there is an oracle Turing machine $\mathcal{M}$, an unknown mass y and a time schedule T such that $\mathcal{M}$ decides A and runs in polynomial time.

We say that a set A is decidable by an unbounded (or fixed) precision BBE machine in polynomial time if there is an oracle Turing machine $\mathcal{M}$ running in polynomial time, an unknown mass y, a time schedule T, and some $0<\gamma<1/2$ such that, for any input word w,

(a) if $w\in A$, then $\mathcal{M}$ accepts w with probability at least $1-\gamma$ and
(b) if $w\notin A$, then $\mathcal{M}$ rejects w with probability at least $1-\gamma$.

THEOREM 7.2
If $A\in P/\log\star$, then A is decidable by a BBE machine with infinite precision in polynomial time. Moreover, the unknown mass y can be chosen to be in $\mathcal{C}_3$ and T to be exponential.

PROOF. Let f be a prefix function in $\log\star$ and $\mathcal{M}'$ be a Turing machine running on polynomial time such that, for any natural number n and any word w such that $|w|\leq n$, $w\in A$ if and only if $\mathcal{M}'$ accepts $\langle w,f(n)\rangle$. Let $y=\mu(f)$ using the encoding of Definition 6.2, and T be any exponential time schedule. Since $f\in\log$, there are constants $a,b\in\mathbb{N}$ such that, for all n, $|f(n)|\leq a\lceil\log(n)\rceil+b$. For each $n\in\mathbb{N}$, let $k_n=3(a+1)\lceil\log(n)\rceil+3b+8$. Resorting to Proposition 4.2 (b), there is a value of ℓ, linear in k_n (and thus linear in $\lceil\log(n)\rceil$), such that the result of running the algorithm Binary Search for input ℓ is a dyadic rational m such that $|y-m|<2^{-k_n}$. Then, by Cantor $\mathcal{C}_3$ properties of Proposition 6.1, m and y coincide in the first $k_n-5=3(a+1)\lceil\log(n)\rceil+3(b+1)$ bits, which means that m can be used to decode $f(2^{\lceil\log(n)\rceil})$. The oracle machine $\mathcal{M}$ that reads the dyadic m and then simulates $\mathcal{M}'$ for the input word $\langle w,f(2^{\lceil\log(n)\rceil})\rangle$ decides A. Furthermore, from Proposition 4.2 (a) and the fact that $A\in P/\log\star$, the time complexity of these activities is polynomial in n. ■

THEOREM 7.3
If $A\in BPP//\log\star$, then A is decidable by a BBE machine with unbounded precision in polynomial time. Moreover, the unknown mass y can be chosen to be in $\mathcal{C}_3$ and T to be exponential.

PROOF. Let $\mathcal{M}'$ be the advice Turing machine working in polynomial time p_3, $f\in\log\star$ the prefix function and $\gamma_3\in(0,1/2)$ the constant witnessing that $A\in BPP//\log\star$. Let $a,b\in\mathbb{N}$ be such that, for all n, $|f(n)|\leq a\lceil\log(n)\rceil+b$. Let γ_2 be such that $\gamma_3+\gamma_2<1/2$. Let $y=\mu(f)$ and consider any exponential time schedule T. By Lemma 5.1, there is a dyadic rational z that can be used to produce independent coin tosses with probability of heads $\delta\in(0,1)$. This rational depends only on y and T and can be hard-wired into the machine. By Lemma 5.2, we can take an integer N (depending on δ and γ_2) such that we can use Nn biased coin tosses to simulate n fair coin tosses, with probability of failure at most γ_2.

For each $n\in\mathbb{N}$, let $k_n=3(a+1)\lceil\log(n)\rceil+3b+8$. By Proposition 4.2 (b), taking T to be exponential there is ℓ, linear in k_n, such that the result of the algorithm Binary Search in the case of unbounded precision, for input ℓ, is a dyadic rational m such that $|y-m|<2^{-k_n}$, so that, by the Cantor set properties of Proposition 6.1, m can be used to decode $f(2^{\lceil\log(n)\rceil})$. We design a oracle machine $\mathcal{M}$ that, on input w of size n, starts by running Binary Search for input ℓ and then uses the result to decode the advice $\mu(f)$. In the next step, the machine uses the dyadic rational z to produce a sequence of $Np_3(n)$ independent biased coin tosses and extract from it a new sequence of $p_3(n)$ independent fair coin tosses. If it fails (which may happen with probability at most γ_2), then the machine rejects w. Otherwise the machine simulates $\mathcal{M}'$ on input $\langle w,f(2^{\lceil\log(n)\rceil})\rangle$ using the sequence of $p_3(n)$ fair coin tosses to decide the path of the computation of $\mathcal{M}'$.

The machine $\mathcal{M}$ decides A in polynomial time (see Figure 6). If $w\in A$, then $\mathcal{M}$ rejects w if it failed to produce the sequence of fair coin tosses or if $\mathcal{M}'$ rejected w. The probability of rejecting w is bounded by $\gamma_2+\gamma_3$. On the other hand, if $w\notin A$, then $\mathcal{M}$ accepts w if it produced a sequence of fair

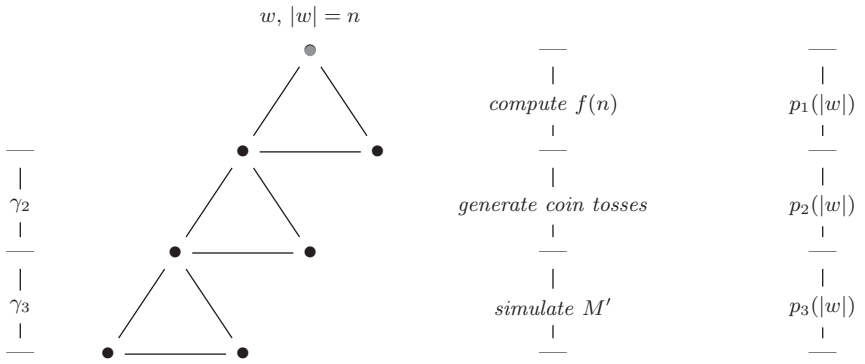

FIGURE 6. Schematic description of the behaviour of the BBE machine.

coin tosses and if $\mathcal{M}'$ accepted w, and this happens with probability at most γ_3. This means that the error probability of $\mathcal{M}$ is bounded by constant $\gamma_2+\gamma_3$ which is less than $1/2$. By Proposition 4.2 (a), the time complexity of the first step is $\mathcal{O}(\ell T(\ell))$. Since ℓ is logarithmic in n and T is exponential in ℓ, the result is bounded by some polynomial in n, $p_1(n)$. The time complexity of the second step is also bounded by some polynomial p_2 in n, since we require only a polynomial amount of $Np_3(n)$ biased coin tosses. Finally, since $\mathcal{M}'$ runs in polynomial time p_3, we conclude that $\mathcal{M}$ runs in polynomial time $\mathcal{O}(p_1+p_2+p_3)$.

■

THEOREM 7.4
If $A \in BPP//\log\star$ and $\epsilon \in (0,1/2)$, then A is decidable by a BBE machine with fixed precision ϵ in polynomial time. Moreover, the unknown mass μ can be chosen to be in $\mathcal{C}_3$ and T to be exponential.

PROOF. The proof follows the lines of the proof of Theorem 7.3, mutatis mutandis, using Proposition 4.3. ■

8 Upper bounds on the BBE machine

Given a threshold oracle (i.e. an oracle with two possible random answers), we can depict the sequence of the answers in a binary tree, where each path is labeled with its probability. The leaves of these trees are marked with an accept or reject. Then, to get the probability of acceptance of a particular word, we simply add the probabilities for each path that ends in acceptance. The next basic idea is to think of what would happen if we change the probabilities in the tree. This means that we are using the same procedure of the Turing machine, but now with a different probabilistic oracle. Suppose that the tree has depth m and there is a real number β bounding the difference in the probabilities labelling all pairs of corresponding edges in the two trees. Proposition 2.1 of [5], states that the difference in the probabilities of acceptance of the two trees is at most $2m\beta$. We need to state and prove a result equivalent to this one in [5] but for dyadic trees, since now each experiment has only two possible results. In [5] we defined $f_d(m,\beta)$ as the largest possible difference in probabilities of acceptance for two different assignments of probabilities with difference at most β in a d-adic probabilistic tree of height m.

DEFINITION 8.1
By a d-adic probabilistic tree, being $d \geq 2$ an integer, we mean a pair $(\mathcal{T}, D)$ where:

- $\mathcal{T}$ is a tree with some set of nodes or vertices V, some set of edges E and some set of leaves $L \subseteq V$;
- $D : E \to [0,1]$ is a map that assigns to each edge u a probability $D(u)$;
- $\mathcal{T}$ is a d-adic tree, i.e. each inner node has exactly d children; moreover, if $u_1, \ldots, u_d$ are its outgoing edges then $D(u_1) + \ldots + D(u_d) = 1$;
- Each leaf is either an accepting node (labelled with 'A') or a rejecting node (labelled with 'R').

When $d = 2$, we may use the expression *binary probabilistic tree* to refer to a 2-adic probabilistic tree. Thus, the set of computations of an BBE machine with non-infinite precision, for a given input, may be represented using a binary probabilistic tree. Given two d-adic probabilistic trees $(\mathcal{T}, D)$ and $(\mathcal{T}, D')$ with the same d-adic tree $\mathcal{T}$, we define the distance $d(D, D')$, as $d(D, D') = \sqcup_{u \in E} |D(u) - D'(u)|$. Let $\mathcal{T}_m^d$ denote the set of d-adic trees of height at most m and $\mathcal{T} \in \mathcal{T}_m^d$. We define a function $f_d : \mathbb{N} \times [0,1] \to [0,1]$ by

$$f_d(m, \beta) = \bigsqcup_{d(D,D') \leq \beta} |P(\mathcal{T}, D) - P(\mathcal{T}, D')|$$

Thus, $f_d(m, \beta)$ gives the largest possible difference in probabilities of acceptance for two different assignments of probabilities with difference at most β.

PROPOSITION 8.2
For any $m \in \mathbb{N}$ and $\beta \in [0,1]$, $f_2(m, \beta) \leq m\beta$.

PROOF. By induction on m. The result is true for $m = 0$. Assuming that $f_2(m, \beta) \leq m\beta$, we shall prove that $f_2(m+1, \beta) \leq (m+1)\beta$. Let $\mathcal{T} \in \mathcal{T}_{m+1}^2$. If $\mathcal{T}$ has height 0 then the result is also true. Otherwise we conclude that there are two edges u_1 and u_2 leaving the root node and two trees $T_1, T_2 \in \mathcal{T}_m^2$ that correspond to the left and right subtrees of $\mathcal{T}$ (see Figure 7). Now let D and D' be two probability assignments for $\mathcal{T}$ such that $d(D, D') \leq \beta$.

We have that $P(\mathcal{T}, D) = pP(T_1, D_1) + qP(T_2, D_2)$ and $P(\mathcal{T}, D') = p'P(T_1, D_1') + q'P(T_2, D_2')$. Furthermore, using the fact that $q = 1 - p$ and $q' = 1 - p'$,

$$\begin{aligned} P(\mathcal{T}, D) - P(\mathcal{T}, D') = {} & (p - p')(P(T_1, D_1) - P(T_2, D_2)) \\ & + p'(P(T_1, D_1) - P(T_1, D_1')) + q'(P(T_2, D_2) - P(T_2, D_2')). \end{aligned}$$

Finally, using the definition of $f_2(m, \beta)$, and observing that the difference of two real numbers in $[0,1]$ must lie in $[-1,1]$, we have $|P(\mathcal{T}, D) - P(\mathcal{T}, D')| \leq |p - p'| + p'f_2(m, \beta) + q'f_2(m, \beta) \leq f_2(m, \beta) + \beta$. Hence, using the induction hypothesis, we conclude that $f_2(m+1, \beta) \leq \beta + f_2(m, \beta) \leq (m+1)\beta$, as we wanted to prove. ■

FIGURE 7. Proving Proposition 8.2.

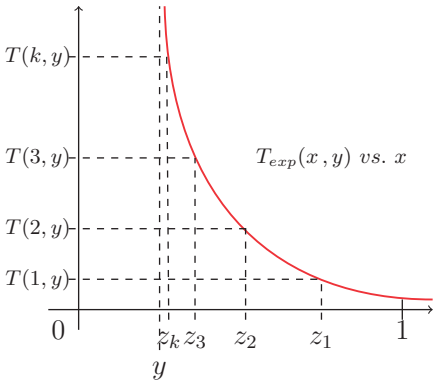

FIGURE 8. Boundary numbers. The value y stands for a fixed unknown mass and each z_i denotes an approximation to y.

8.1 $P/\log^2\star$ is an upper bound for the infinite precision case

DEFINITION 8.3
Let $y \in (0,1)$ be the unknown mass and T a time schedule. Then, for all $k \in \mathbb{N}$, we define $z_k \in (0,1)$ as the number such that $T_{exp}(z_k, y) = T(k)$. These numbers are illustrated in Figure 8, and were the boundary numbers discussed in Subsection 3.3.

For any oracle query z of size k, (a) if $z \leq z_k$,[10] then the result of the experiment is 'TIMEOUT' and (b) if $z > z_k$, then the result of the experiment is 'YES'. Notice that $z_k\downarrow_k$ is precisely the result of the algorithm for input k and as such, by knowing $z_k\downarrow_k$, we can obtain the result of any experiment of size k (in the infinite precision case) without having to perform it. This is the core idea of the two following proofs.

THEOREM 8.4
If A is a set decidable by a BBE machine with infinite precision in polynomial time and the chosen time schedule is exponential, then $A \in P/\log^2\star$.

PROOF. Suppose that A is decided by a BBE machine $\mathcal{M}$ in polynomial time, with exponential time schedule T. Since T is exponential and the running time is polynomial, we conclude that the size of the oracle query grows at most logarithmicly in the size of the input word, i.e. there are constants $a, b \in \mathbb{N}$ such that, for any input word of size n, the computation of $\mathcal{M}$ only queries words with size less or equal to $a\lceil\log(n)\rceil + b$. Consider the advice function f such that $f(n)$ encodes the concatenation of words $z_1\downarrow_1 \# z_2\downarrow_2 \# \cdots \# z_t\downarrow_t$, where $t = a\lceil\log(n)\rceil + b$. We observe that f is a prefix function and $|f(n)| \in \mathcal{O}(t - 1 + \sum_{i=1}^{t} i) = \mathcal{O}(t^2) = \mathcal{O}(\log^2(n))$. Furthermore, we can use $f(n)$ to determine the answer to any possible oracle query of size less than $a\lceil\log(n)\rceil + b$. To decide the set A in polynomial time with advice f, simply simulate the original machine $\mathcal{M}$ on the input word and, whenever $\mathcal{M}$ is in the query state, simulate the experiment by comparing the query word with the appropriate z_i in the advice function. As this comparison can be done in polynomial time and $\mathcal{M}$ runs in polynomial time too, we conclude that A can be decided in polynomial time with the given advice. ■

[10]This comparison can be seen either as a comparison between reals—the mass values— or as a comparison between binary strings in the lexicographical order—the corresponding dyadic rationals.

Observe that $z_k \searrow y$, where y is the unknown mass. As we are going to see, under some extra assumptions on the time schedule, the value of $z_{k+1}\lfloor_{k+1}$ can be obtained by adding to the word $z_k\lfloor_k$ a very few bits of information, shortening the encoding to $\mathcal{O}(\log(n))$ bits.

THEOREM 8.5
If A is a set decidable by a BBE machine with infinite precision in polynomial time and the chosen time schedule is $T(k) \in \Omega(2^{k/2})$,[11] then $A \in P/\log\star$.

PROOF. Since $T(k) \in \Omega(2^{k/2})$, it follows that there exist constants $\varphi, k_0 \in \mathbb{N}$ such that $T(k) \geq \varphi 2^{k/2}$, for $k \geq k_0$. By Proposition 4.2, we can ensure that the value of the boundary number z_k is such that $y < z_k < y + 2^{-k+c}$, for some constant $c \in \mathbb{N}$ and for $k > k_0$. This means that, when we increase the size of k by one bit, we also increase the precision on y by one bit. Let us write the dyadic rational $z_k\lfloor_k$ as the concatenation of two strings, $z_k\lfloor_k = x_k \cdot y_k$, where y_k has size c and x_k has size $k-c$. Note that $z_k - 2^{-k+c} < x_k < z_k$, i.e. $|x_k - y| < 2^{-k+c}$. The bits of x_k provide information about the possibilities for the binary expansion of y. We show that we can obtain x_{k+1} from x_k with just two more bits of information. Suppose that x_k ends with the sequence $x_k = \cdots 10^\ell$. The only two possibilities for the first $k-c$ bits of y are $\cdots 10^\ell$ or $\cdots 01^\ell$. Thus, x_{k+1} must end in one of the following: $x_{k+1} = \cdots 10^\ell 1$ or $x_{k+1} = \cdots 10^\ell 0$ or $x_{k+1} = \cdots 01^\ell 1$ or $x_{k+1} = \cdots 01^\ell 0$. That is, *even though x_k is not necessarily a prefix of x_{k+1}*, it still can be obtained from x_k by appending some information that determines which of the four possibilities occur.[12] Suppose now that x_k ends with the sequence $x_k = \cdots 01^\ell$. The only two possibilities for the first $k-c$ bits of y are $\cdots 01^{\ell-1}1$ or $\cdots 01^{\ell-1}0$. Thus, x_{k+1} must end in one of the following: $x_{k+1} = \cdots 01^\ell 0$ or $x_{k+1} = \cdots 01^\ell 1$ or $x_{k+1} = \cdots 01^{\ell-1}00$ or $x_{k+1} = \cdots 01^{\ell-1}01$. In the same way, x_{k+1} still can be obtained from x_k by appending some information that determines which of the four possibilities occur.

We define the function $f(n)$ as follows: (a) if $n < k_0$, then $f(n) = z_1\lfloor_1 \# z_2\lfloor_2 \# \cdots \# z_n\lfloor_n$, (b) $f(k_0) = f(k_0 - 1)\# x_{k_0} \# y_{k_0}$, and (c) if $n > k_0$, then $f(n) = f(n-1)\# b_1 b_2 y_n$, where the bits $b_1 b_2$ are used to determine one of the four possibilities for x_n with respect to x_{n-1}. Observe also that from $f(n)$ one can recover the values of $z_k\lfloor_k$, for all $k \leq n$. Moreover, $|f(n)|$ is linear in n, since all y_k have size d. Since A is decided by a BBE machine $\mathcal{M}$ in polynomial time and T is exponential, the size of the oracle query grows at most logarithmic in the size of the input word. There are constants $d, e \in \mathbb{N}$ such that, for any input word of size n, the computation of $\mathcal{M}$ only queries for words with size less or equal to $d\lceil \log(n) \rceil + e$. We define the advice function $g : \mathbb{N} \to \{0,1\}^\star$ such that $g(n) = f(d\lceil \log(n) \rceil + e)$. Note that $|g(n)| = \mathcal{O}(\log(n))$ and $g(n)$ can be used to determine the result of any oracle query for any computation for any input word of size less or equal to n. Then, as in the proof of Theorem 8.4, we can devise a Turing machine that decides A in polynomial time using g as advice, witnessing that $A \in P/\log\star$. ■

8.2 *$BPP//\log^2\star$ is an upper bound for the unbounded precision case*

Our next step is to prove that any set decidable using a BBE machine with unbounded precision in polynomial time can also be decided in polynomial time using an advice of a particular size. Given a BBE machine $\mathcal{M}$, we construct an advice function f with the following properties: (a) for any n, $f(n)$ contains enough information to answer all queries occurring during the computation of $\mathcal{M}$ on a word

[11] We define $\Omega(g)$ as the class of functions f such that there exist $p \in \mathbb{N}$ and $r \in \mathbb{R}^+$ such that, for all $n \geq p$, $f(n) \geq rg(n)$.

[12] The following example helps to clarify the argument. Suppose that $y = 0.1100011000\ldots$ The sequence x_k can be taken as follows: $x_1 = 1$, $x_2 = 11$, $x_3 = 111$, $x_4 = 1101$, $x_5 = 11001$, $x_6 = 110010$, $x_7 = 1100100$, $x_8 = 11000111$, $x_9 = 110001100$, ...

of size n and (b) the size of $f(n)$ grows as slowly as we can accomplish. In the previous section, we made the observation that a dyadic rational $z_n\downarrow_n$ of size n could be used to answer all oracle queries of size up to n. Thus, using an exponential time schedule, we could simulate any polynomial time computation having the oracle replaced by an advice containing a logarithmic number of $z_n\downarrow_n$'s.

Remember that $z_k\downarrow_k$ is the result of the algorithm Binary Search for input k, or alternatively, is the biggest dyadic rational of size k for which the result of the experiment is 'TIMEOUT' (see Proposition 4.1). The second property is that, when performing the experiment Mass$(z_k\downarrow_k)$ (unbounded precision case), since the mass z' is uniformly sampled from the interval $(z_k\downarrow_k - 2^{-k}, z_k\downarrow_k + 2^{-k})$, the probability of obtaining result 'YES' is precisely $(z_k\downarrow_k + 2^{-k} - z_k)/(2\times 2^{-k}) = 1/2 - (z_k - z_k\downarrow_k)/(2\times 2^{-k})$. From these facts we can conclude that, if we know the first $k+d$ bits of z_k, then we can obtain an approximation of the probability of answer 'YES' when performing experiment Mass$(z_k\downarrow_k)$ with an error of at most 2^{-d}. The same reasoning can be made for the experiment Mass$(z_k\downarrow_k + 2^{-k})$, which is the other dyadic rational of size k for which the experiment is not deterministic. In this case, the probability of answer 'YES' is $1-(z_k - z_k\downarrow_k)/(2\times 2^{-k})$, and this value can also be approximated by knowing the first $k+d$ bits of z_k, with an error of at most 2^{-d}.

THEOREM 8.6
If A is a set decided by a BBE machine $\mathcal{M}$ in polynomial time with unbounded precision and exponential time schedule T, then $A \in BPP//\log^2\star$.

PROOF. Suppose that $\mathcal{M}$ runs in polynomial time $\mathcal{O}(n^a)$ and $T(k)$ is exponential in k. As in the proof of Theorem 8.4, we can assume that, for input words of size n, all oracle queries that occur in the computations are of size at most $d\log(n)+e$. Also, there must be another constant α such that at most αn^a oracle queries are made in the computation of any word of size n. This means that the probabilistic tree of computations on a word of size n has a depth of at most αn^a. Let γ be the bound on the error probability associated with $\mathcal{M}$ and b be an integer such that $2^b > 2\alpha/(1/2-\gamma)$. Our goal is to approximate all probabilities of all oracle queries by at most $2^{-b-a\log(n)}$. Then the difference in the probability of acceptance (Proposition 2.1 of [5]) will be of at most $2\times\alpha n^a \times 2^{-b-a\log(n)} \leq 2\alpha/2^b < 1/2-\gamma$; the bound can thus be achieved by knowing the first $k+b+a\log(n)$ bits of z_k, for $1\leq k\leq d\log(n)+e$ (see remark earlier in this section).

Let $\xi = max(a,d)$ and f be such that (a) $f(0)$ is the concatenation of the first $b+e$ bits of the reals $z_1,\ldots,z_e$ and (b) $f(t+1)$ is the concatenation of $f(t)$, the bits $b+e+2\xi t+1,\ldots,b+e+2\xi t+2\xi$ of the reals $z_1,\ldots,z_{\xi t+e}$, and the first $b+e+2\xi t+2\xi$ bits of the reals $z_{e+\xi t+1},\ldots,z_{e+\xi t+\xi}$. Observe that $f(t)$ can be decoded in order to find the approximations of z_k, for $1\leq k\leq \xi t+e$. After reading $f(t)$ a Turing machine can read $\xi t+e$ blocks of size 2ξ to update the approximations of z_i, $1\leq i\leq \xi t+e$. Then the machine reads ξ blocks of size $b+e+2\xi t+2\xi$ to get approximations of $z_{e+\xi t+i}$, $1\leq i\leq\xi$; $f(t)$ contains exactly the first $b+e+2\xi t$ bits of the reals z_i, $1\leq i\leq e+\xi t$, and so $|f(t)| = (b+e+2\xi t)(e+\xi t) = \mathcal{O}(t^2)$. The advice function required is the prefix function $g(n) = f(\lceil\log(n)\rceil)$ of size $|g(n)|\in\mathcal{O}(\log^2(n))$. The advice $g(n)$ can provide approximations of all z_k, for $k\leq d\log(n)+e$, with at least $k+b+a\log(n)$ bits of precision.

Consider the advice Turing machine $\mathcal{M}'$ that first writes on a tape the approximations of z_k, for $k\leq d\log(n)+e$, simulates $\mathcal{M}$ for the input word and, whenever $\mathcal{M}$ is in a query state, compares the query word with the appropriate z_k. The machine $\mathcal{M}'$ uses the approximation of z_k to compute the probability of a 'YES' with error less than $2^{-b-d\log(n)}$. Then $\mathcal{M}'$ simulates a coin toss with that probability, which can be done by performing $b+d\log(n)$ coin tosses. The corresponding probabilistic tree is similar to that of the original machine, with depth lower than αn^a and edge difference less than $2^{-b-d\log(n)}$. The difference in the probabilities of acceptance is bounded by $1/2-\gamma$ and the

probability that $\mathcal{M}'$ gives a wrong answer is less than $\gamma+1/2-\gamma=1/2$. Since all the simulations run in polynomial time, we conclude that $\mathcal{M}'$ decides A in polynomial time. ■

8.3 *BPP*//$\log^2\star$ *is an upper bound for the finite precision case*

We now establish an upper bound for the class of sets decided by BBE machines with finite precision in polynomial time. Theorem 8.8 has a proof that follows the same lines of the previous proof. We discuss now how the bits of the probability distribution can be computed. The numbers z_k are defined as in the beginning of Section 8.2. The following proposition is straightforward:

PROPOSITION 8.7
For any dyadic rational z of size k let $P(z)$ be the probability of obtaining answer 'YES' when performing the experiment with test mass z, unknown mass y, finite precision ϵ, and time schedule T.

$$P(z)=\begin{cases} 0 & \text{if } z<z_k-\epsilon \\ \frac{1}{2}+\frac{z-z_k}{2\epsilon} & \text{if } z_k-\epsilon\le z<z_k+\epsilon \\ 1 & \text{if } z_k+\epsilon\le z \end{cases}.$$

As before, given the BBE machine $\mathcal{M}$, we define the sequence of real numbers z_k such that $T_{exp}(z_k,y)=T(k)$. Our advice function will contain dyadic rational approximations of z_k and ϵ that will be used to compute approximations to $P(z)$. We will use in this section and the next the facts that, if $\xi,\upsilon\in\mathbb{R}$ are such that $0<\xi<\upsilon$ and $\Delta\xi,\Delta\upsilon\in\mathbb{R}$ are such that $\Delta\upsilon>0$, then

$$|(\xi+\Delta\xi)(\upsilon+\Delta\upsilon)-\xi\upsilon|\le\upsilon|\Delta\xi|+\xi\Delta\upsilon+|\Delta\xi|\Delta\upsilon \tag{6}$$

$$\left|\frac{\xi+\Delta\xi}{\upsilon+\Delta\upsilon}-\frac{\xi}{\upsilon}\right|<\frac{|\Delta\xi|}{\upsilon}+\frac{\Delta\upsilon}{\upsilon}, \tag{7}$$

to compute how many digits of z_k and ϵ are required to obtain an approximation of $P(z)$ up to 2^{-e}, for some $e\in\mathbb{N}$ and for any dyadic rational z of size k. Let c be an integer such that $2^{-c}\le\epsilon$, and let z_k' and ϵ' be z_k and ϵ rounded up to the first $c+e$ and $c+e+1$ bits, respectively. We can then compute $(z-z_k)/2$ with precision 2^{-c-e-1}. By means of inequality (7) above, with ϵ as υ and $(z-z_k)/2$ as ξ, the quotient $1/2+(z-z_k')/(2\epsilon')$ provides an approximation of $P(z)$ with error less than 2^{-e}. The number of digits required grows linearly with the precision desired on $P(z)$ that in its turn increases logarithmically with the size of the input word. We conclude that, for queries of size less or equal to that of z, only a logarithmic amount of bits of $P(z)$ is required.

THEOREM 8.8
If A is a set decided by a BBE machine in polynomial time with fixed precision and an exponential time schedule, then $A\in BPP//\log^2\star$.

PROOF. This proof is similar to the proof of Theorem 8.6. However, there is one more value to take into account, which is the value of the precision ϵ. Suppose that $\mathcal{M}$ runs in polynomial time $\mathcal{O}(n^a)$ and $T(k)$ is exponential in k. As in the proof of 8.4, we can assume that, for an input word of size n, all oracle queries that occur on the computations are of size at most $d\log(n)+e$. Also, there must be another constant α such that at most αn^a oracle queries are made in the computation of any word of size n. This means that the binary probabilistic tree has depth at most αn^a. Let γ be the bound on the error probability of $\mathcal{M}$ and let b be such that $2^b>2\alpha/(1/2-\gamma)$. Our goal is to approximate all

probabilities of all oracle queries by at most $2^{-b-a\log(n)}$. Then the difference in the probability of acceptance (Proposition 2.1 of [5]) will be of at most $2\times\alpha n^a\times 2^{-b-a\log(n)}\leq 2\alpha/(1/2-\gamma)$.

Now the proof differs slightly from the unbounded precision case. By what we said before, to obtain such an approximation of the probabilities of all oracle queries, we require $c+b+a\log(n)$ bits of z_k, for all k between 1 and $d\log(n)+e$, and also $c+b+a\log(n)+1$ bits of ϵ, where c is an integer such that $2^{-c}\leq\epsilon$. Let the auxiliary function f be defined as follows: (a) $f(0)$ is the concatenation of the first $c+b$ bits of the reals z_1 through z_e and the first $c+b+1$ bits of ϵ and (b) $f(t+1)$ is the concatenation of $f(t)$, the bits $c+b+at$ to $c+b+at+a$ of the reals z_1 to z_{e+dt}, the first $c+b+at+a$ bits of the reals z_{e+dt+1} to z_{e+dt+d}, and the bits $c+b+at+2$ to $c+b+at+a+1$ of ϵ. Observe that f can be decoded in order to find the approximations of z_k, for $1\leq k\leq e+dt$. After reading $f(t)$, a Turing machine can read $e+dt$ blocks of size a to update the approximations of z_i, $1\leq i\leq e+dt$; then it can read d blocks of size $c+b+at+a$ to get approximations of z_{e+dt+i}, $1\leq i\leq d$; finally, the machine can read a block of size a to update the approximation of ϵ. Thus $f(t)$ contains exactly the first $c+b+at$ bits of the reals z_i, $1\leq i\leq e+dt$ and the first $c+b+at+1$ bits of ϵ, and so $|f(t)|=(c+b+at)(e+dt)+c+b+at+1=\mathcal{O}(t^2)$. The advice function required is the prefix function $g(n)=f(\lceil\log(n)\rceil)$ of size $|g(n)|=\mathcal{O}((\log(n))^2)$. The advice $g(n)$ can provide approximations of all z_k, for $1\leq k\leq e+d\log(n)$, with at least $c+b+a\log(n)$ bits of precision and an approximation of ϵ with $c+b+a\log(n)+1$ bits of precision.

Consider the advice Turing machine $\mathcal{M}'$ that first writes on a tape, from the advice function g, the approximations of z_k, for $k\leq d\log(n)+e$, and of ϵ, simulates $\mathcal{M}$ for the input word and, whenever $\mathcal{M}$ is in a query state with query z, computes an approximation of $P(z)$ from the approximations of ϵ and the appropriate z_k, with an error less than $2^{-b-d\log(n)}$; finally it simulates a coin toss with such a probability, which can be done by performing $b+d\log(n)$ coin tosses. This machine induces a binary probabilistic tree in the same way as the original machine, with depth less or equal to αn^a and edge difference less or equal to $2^{-b-d\log(n)}$. As we saw, the difference in the probabilities of acceptance is then bounded by a constant less than $1/2-\gamma$. Thus, the probability that this machine gives a wrong answer is less than $\gamma+1/2-\gamma=1/2$. Since $\mathcal{M}$ runs in polynomial time and all oracle calls can be simulated in polynomial time, we conclude that the machine $\mathcal{M}'$ decides A in polynomial time. ■

8.4 *BPP*$//\log\star$ *is an upper bound for the unbounded precision case assuming explicit time*

The upper and lower bounds for the unbounded precision do not coincide. We could wonder if there is some way to improve on this bound in order to obtain a full characterization of the class of decidable sets by BBE machines. Indeed, in this section we shall see that, under some assumptions, we can derive such an upper bound.

The main step of the following proof is to devise, for given natural constants b, e and ξ, an advice function f such that $f(t)$ contains the first $b+e+2\xi t$ of the reals z_i, for $1\leq i\leq e+\xi t$, where z_i is the solution of the equation $T_{exp}(z_i,y)=T(i)$. Since $z_i>y$, we know the exact expression of T_{exp} that we write, for simplicity in what follows:

$$T_{exp}(z_i,y)^2=\eta\,\frac{z_i+y}{z_i-y},$$

where η is some constant. We can solve the equation to find that, for sufficient large i, we have $z_i=y(1+2\eta/(T(i)^2-\eta))$. Knowing approximations of y, η and $T(i)$, we can compute an approximation

of z_i. So our advice function will basically consist of enough digits of y and η. Let us suppose we want to obtain an approximation of z_i with error less than 2^{-c}. Suppose that we know the first $c+2$ digits of y and the first $c+4$ digits of η. Since we can compute in proper time the exact value of $T(i)^2-\eta$ with error less than 2^{-c-4}, we can also compute the value of $\eta/(T(i)^2-\eta)$ with error less than $2^{-c-4}+2^{-c-4}=2^{-c-3}$, bound provided by inequality (7) in the previous section. This implies that we can compute $(1+2\eta/(T(i)^2-\eta))$ with error less than 2^{-c-2}. Finally, since $y<1$ and $(1+2\eta/(T(i)^2-\eta))<2$, we can use these approximations to compute z_i with error less than $2\times 2^{-c-2}+2^{-c-2}+2^{-c-2}2^{-c-2}\leq 2^{-c}$, bound provided by inequality (6). Thus, only a linear amount of digits of y and η are required to obtained a precision of c digits on z_i.

THEOREM 8.9
If A is decided by a BBE machine $\mathcal{M}$ with unbounded precision in polynomial time and T is an exponential time schedule, then $A\in BPP//\log\star$.

PROOF. Suppose that $\mathcal{M}$ runs in polynomial time and $T(k)$ is exponential in k. By the proof of Theorem 8.6, we can specify a probabilistic Turing machine running in polynomial time that decides A helped by an advice function of size $\mathcal{O}((\log(n))^2)$. We defined the advice $g(n)=f(\lceil\log(n)\rceil)$, where $f(t)$ contains exactly the first $b+e+2\xi t$ bits of the reals z_i, $1\leq i\leq e+\xi t$, for some constants b, e and ξ. Let k_0 be such that, for $k>k_0$ we have that $z_k=y(1+2\eta/(T(k)^2-\eta))$. We define an auxiliary function $\tilde{f}$ such that (a) $\tilde{f}(0)$ is the concatenation of the first $b+e+2$ bits of y, the first $b+e+4$ bits of η and the first $b+e$ bits of the reals z_i, for $1\leq i\leq k_0$ and (b) $\tilde{f}(t+1)$ is the concatenation of $\tilde{f}(t)$, the bits $b+e+2\xi t+3$ to $b+e+2\xi t+2\xi+2$ of y, the bits $b+e+2\xi t+5$ to $b+e+2\xi t+2\xi+4$ of η and the bits $b+e+2\xi t+1$ to $b+e+2\xi t+2\xi$ of the reals z_i, for $1\leq i\leq k_0$. We can use $\tilde{f}(t)$ to obtain the first $b+e+2\xi t+2$ digits of y, the first $b+e+2\xi t+4$ digits of η and the first $b+e+2\xi t$ digits of the reals z_i, for $1\leq i\leq k_0$. By the previous discussion, we can then use the approximations of y and η to compute the first $b+e+2\xi t$ bits of the reals z_i, $k_0<i\leq e+\xi t$. We see that $|\tilde{f}(t)|=b+e+2\xi t+2+b+e+2\xi t+4+k_0(b+e+2\xi t)=\mathcal{O}(t)$. Finally, the advice function required is $\tilde{g}(n)=\tilde{f}(\lceil\log(n)\rceil)$. Observe that $\tilde{g}$ is a prefix function and that $|\tilde{g}(n)|=\mathcal{O}(\log(n))$. Furthermore, $\tilde{g}(n)$ can be used to compute $g(n)$ in polynomial time.

Now we specify a probabilistic Turing machine $\mathcal{M}''$ to decide A in polynomial time, using $\tilde{g}$ as advice. On input x, the machine $\mathcal{M}''$ starts by retrieving $g(|x|)$ from $\tilde{g}(|x|)$ and then simulates the machine $\mathcal{M}'$ on input x with advice $g(|x|)$ as in the proof of Theorem 8.6. All these operations can be achieved in polynomial time. ■

8.5 *$BPP//\log\star$ is an upper bound for the finite precision case assuming explicit time*

We will do a similar construction to the one we did in the previous section but for the finite precision case. We define an advice function $\tilde{g}$ from which we can derive the advice function g used in the proof of 8.8. We observe that g consisted on the concatenation of bits of the reals z_k and bits of ϵ. As in the proof of 8.9, we will use approximations of y and ϵ to get approximations of z_k, where y is the unknown mass and ϵ is the fixed error allowed.

THEOREM 8.10
If A is decided by a BBE machine $\mathcal{M}$ with unbounded precision in polynomial time and T is an exponential time schedule, then $A\in BPP//\log\star$.

PROOF. Suppose that $\mathcal{M}$ runs in polynomial time and $T(k)$ is exponential in k. From Theorem 8.8, we conclude that $A \in BPP//\log^2\star$, i.e. A is decided by a probabilistic Turing machine in polynomial time with advice in $\log^2\star$. Moreover, the advice function g is such that $g(n)=f(\lceil\log(n)\rceil)$, where $f(t)$ contains exactly the first $c+b+at$ bits of the reals z_i, $1 \le i \le e+dt$ and the first $c+b+at+1$ bits of ϵ, where a, b, c, d and e are integer constants. Let k_0 be such that, if $k > k_0$, then z_k is defined and given by $z_k = y(1+2\eta/(T(k)^2-\eta))$. Let $\tilde{f}$ be such that (a) $\tilde{f}(0)$ is the concatenation of the first $b+c+2$ bits of y, the first $b+c+4$ bits of η, the first $c+b+1$ bits of ϵ and the first $c+b$ bits of the reals z_i, for $1 \le i \le k_0$, and (b) $\tilde{f}(t+1)$ is the concatenation of $\tilde{f}(t)$, the bits $b+c+at+3$ to $b+c+at+a+2$ of y, the bits $b+c+at+5$ to $b+c+at+a+4$ of η, the bits $b+c+at+2$ to $b+c+at+a+1$ of ϵ and the bits $b+c+at+1$ to $b+c+at+a$ of the reals z_i, for $1 \le i \le k_0$. Then $\tilde{f}(t)$ can be used to obtain the first $b+c+at+2$ bits of y, the first $b+c+at+4$ bits of η, the first $b+c+at+1$ bits of ϵ and the first $b+c+at$ bits of the reals z_i, for $1 \le i \le k_0$. As explained in the introductory remarks of this section, we can use the bits of y and η to obtain the first $b+c+at$ bits of the reals z_i, for $k_0 < i \le e+dt$. That is, we can use $\tilde{f}(t)$ to compute $f(t)$. Furthermore, $|\tilde{f}(t)| = b+c+at+2+b+c+at+4+b+c+at+1+k_0(b+c+at) = \mathcal{O}(t)$. Finally, the advice function required is $\tilde{g}(n)=\tilde{f}(\lceil\log(n)\rceil)$. Observe that $\tilde{g}$ is a prefix function and that $|\tilde{g}(n)| \in \log\star$. Furthermore, $\tilde{g}(n)$ can be used to compute $g(n)$.

Now we specify a probabilistic Turing machine $\mathcal{M}''$ to decide A in polynomial time, using $\tilde{g}$ as advice. On input x, the machine $\mathcal{M}''$ starts by retrieving $g(|x|)$ from $\tilde{g}(|x|)$ and then simulates the machine $\mathcal{M}'$ on input x (as in the proof of Theorem 8.8) with advice $g(|x|)$. All these operations can be achieved in polynomial time. ■

Observe that Theorems 8.6, 8.8, 8.9 and 8.10 used the assumption that the time schedule is exponential. Since the experimental time is itself exponential, it makes sense to consider a time schedule that also grows exponentially (in polynomial time!). It seems that there is no way to use a polynomial time schedule to measure a polynomial number of bits of the unknown mass in polynomial time. However, there is room for improvement in these bounds.

We have used the explicit expression of physical time in order to obtain an equation relating the unknown mass y with the real numbers z_k. Even though we could use that information to get the upper bounds, this technique may not be usable on general hybrid systems. The equation concerning the broken balance is fairly easy to solve for z_k, but could we solve efficiently the equivalent equations for other systems? In general, if we only know that the experimental time has an exponential growth, but we do not know its exact expression, then there is no way to use this technique.

9 Conclusions

We have introduced methods to study the computational power of threshold experiments in which quantities can only be measured either from below or from above. We showed that Turing machines equipped with threshold oracles in polynomial time have a computational power between $P/\log\star$ and $BPP//\log^2\star$, no matter whether the precision is infinite, unbounded or fixed. First, let us reflect on the classes we have encountered in working with physical oracles.

9.1 Stability and invariance

The computational power of Turing machines in polynomial time, assisted by physical oracles, spans a wide spectrum. At the top is Post's interpretation of Turing's original idea of oracle: a machine able to answer any relevant query—formulated as a set membership problem—correctly within one time step. We have no idea how to make such oracles in non-trivial physical cases. Turing used the name

'oracle', to emphasize that such things stood outside our theory of computability. He also used the term 'devices', which strictly speaking do not stand outside our physical theories.

The class P/poly was proposed to characterize the computational power of neural nets in [22], without much physical argument. In our theory of physical oracles, P appeared as the class defined by the very simple *scatter machine with infinitely sharp point* in [2]. It is also the class given by an oracle which is the receiver of a stream of information at a constant rate (the stream being the advice function) but where the stream of information is independent of any query—an example is given in the novel *Contact* by Carl Sagan.

The physical oracles giving P/poly rely on obviously unrealistic precision in measurement—an infinitely sharp point in space or in time. In trying to repair this, by making the measurements more consistent with physical theories, in [8] we created a second physical model that as an oracle gives the class $P/\log\star$. However, this computational class as an upper bound relies on the only useful information being encoded in the single real value that the experiment is trying to measure.

In this article, we make the point that, in principle, the measuring apparatus may code sufficiently much other information to possibly exceed $P/\log\star$, and that a new lower bound may be $P/\log^2\star$. This makes the computation dependent on the exact choice of measuring apparatus and also on the Turing machine's internal clock.

All these cases assume that in some way, measurements comparing quantities can be made without error. Error may be introduced by, for example, thermal vibrations, and we can try to use probability theory to allow for this. Alternatively, we might descend to the quantum world, where quantum effects may enforce precision. However quantum measurements are intrinsically probabilistic: so, either way, we end up with probabilities. As related in various places, and in this article, the introduction of probabilities has surprisingly little effect on the classes listed above, with $P/\log\star$ becoming $BPP//\log\star$ and $P/\log^2\star$ becoming $BPP//\log^2\star$. Note that even the class $P/\log\star$ breaks the Church-Turing Barrier.

Although the variation in computational power is not dramatic, we know enough to establish there is a variation and that it is subtle. We have succeeded in axiomatizing a large class of physical oracles and characterizing their computational power in [11]. Now it remains for us to axiomatize a class of threshold oracles that would include the broken beam balance.

The exploration of the stability and invariance of the computational models is work in progress. We expect that hybrid systems in general cannot transcend such computational power and, indeed, that the computational classes $P/\log\star$ and $BPP//\log\star$ characterize hybrid systems. (Our results weaken the claims of some other classes, such as P/poly, associated with models of physical systems.) Which non-uniform classes and under what conditions we expect to be a matter for physical argument.

9.2 *Types of experiments*

In studying two-sided experiments as in the unbroken balance—and in all our previous experiments, and the axiomatic class of oracles [11])—we saw that an oracle answer such as 'LEFT' would imply that $z < y$ and an oracle answer such as 'RIGHT' would imply that $z > y$, where y is the unknown quantity and z the test quantity. In a threshold experiment, we saw that the oracle answer 'YES' would imply that $z > y$.

However, there exists yet another type of physical experiment, the *vanishing experiment*, in which the answer 'YES' implies only that $z \neq y$. An example is the determination of Brewster's angle in Optics: in the lab measurement of the critical angle of incidence of a monochromatic light ray into

the surface of separation of media such that there is a transmitted ray but no reflected ray. Vanishing experiments are new type of measurement to investigate.

9.3 Ideal nature of the models

We think that our model can capture (i) the limits of the computational power of hybrid systems; (ii) the complexity of individual measurements; and (iii) the limits of what can be measured (such as in [8]). Reactions towards our gedankenexperiments can express dissatisfaction by pointing out that such idealized devices cannot be built—even in the case of such measuring mass as in Section 2! Unfortunately, there seems to be a diffuse philosophy that considers the Turing machine an object of a different kind. Clearly, both the abstract model of a physical experiment and the Turing machine are both gedankenexperiments and non-realizable. To implement a Turing machine the engineer would need either unbounded space and an unlimited physical support structure, or unbounded precision in some finite interval to code for the contents of the tape. In fact, the experiment can be set up to some degree of precision in the same way that the Turing machine can be implemented up to some degree accuracy. Knowing that both objects, the Turing machine and the measurement equipment, are of the same ideal nature, we argue that our models allow us to study the power of feeding physical measurements as real numbers to computing devices, which is the primary characteristic of hybrid machines, and to study the limits of what can be measured.

Acknowledgements

To Bill Tantau for the use of pgf/TikZ applications. The research of José Félix Costa and Diogo Poças is supported by Fundação para a Ciência e Tecnologia, PEst – OE/MAT/UI0209/2011.

References

[1] J. A. Anderson. *An Introduction to Neural Networks*. Massachusetts Institute of Technology, 1995.

[2] E. Beggs, J. F. Costa, B. Loff, and J. V. Tucker. Computational complexity with experiments as oracles. *Proceedings of the Royal Society, Series A (Mathematical, Physical and Engineering Sciences)*, **464**, 2777–2801, 2008.

[3] E. Beggs, J. F. Costa, B. Loff, and J. V. Tucker. On the complexity of measurement in classical physics. In *Theory and Applications of Models of Computation (TAMC 2008)*, Vol. 4978 of *Lecture Notes in Computer Science*, M. Agrawal, D. Du, Z. Duan, and A. Li, eds, pp. 20–30, Springer, 2008.

[4] E. Beggs, J. F. Costa, B. Loff, and J. V. Tucker. Oracles and advice as measurements. In *Unconventional Computation (UC 2008)*, Vol. 5204 of *Lecture Notes in Computer Science*, C. S. Calude, J. F. Costa, R. Freund, M. Oswald, and G. Rozenberg, eds, pp. 33–50, Springer, 2008.

[5] E. Beggs, J. F. Costa, B. Loff, and J. V. Tucker. Computational complexity with experiments as oracles II. Upper bounds. *Proceedings of the Royal Society, Series A (Mathematical, Physical and Engineering Sciences)*, **465**, 1453–1465, 2009.

[6] E. Beggs, J. F. Costa, and J. V. Tucker. Physical experiments as oracles. *Bulletin of the European Association for Theoretical Computer Science*, **97**, 137–151, 2009.

[7] E. Beggs, J. F. Costa, and J. V. Tucker. Computational Models of Measurement and Hempel's Axiomatization. In *Causality, Meaningful Complexity and Knowledge Construction*, Vol. 46 of *Theory and Decision Library A*, A. Carsetti, ed, pp. 155–184, Springer, 2010.

[8] E. Beggs, J. F. Costa, and J. V. Tucker. Limits to measurement in experiments governed by algorithms. *Mathematical Structures in Computer Science*, **20**, 1019–1050, 2010.

[9] E. Beggs, J. F. Costa, and J. V. Tucker. Physical oracles: The Turing machine and the Wheatstone bridge. *Studia Logica*, **95**, 279–300, 2010.

[10] E. Beggs, J. F. Costa, and J. V. Tucker. The Turing machine and the uncertainty in the measurement process. In *Physics and Computation, P&C 2010*, H. Guerra, ed, pp. 62–72. CMATI – Centre for Applied Mathematics and Information Technology, University of Azores, 2010.

[11] E. Beggs, J. F. Costa, and J. V. Tucker. Axiomatising physical experiments as oracles to algorithms. *Philosophical Transactions of the Royal Society, Series A (Mathematical, Physical and Engineering Sciences)*, **370**, 3359–3384, 2012.

[12] E. Beggs, J. F. Costa, and J. V. Tucker. The impact of models of a physical oracle on computational power. *Mathematical Structures in Computer Science*, **22**, 853–879, 2012.

[13] E. Beggs, J. F. Costa, D. Poças, and J. V. Tucker. On the power of threshold measurements as oracles. In *Unconventional Computation and Natural Computation (UCNC 2013)*, Vol. 7956 of *Lecture Notes in Computer Science*, G. Mauri, A. Dennunzio, L. Manzoni, and A. E. Porreca, eds, pp. 6–18, Springer, 2013.

[14] E. Beggs, J. F. Costa, and J. V. Tucker. A natural computation model of positive relativisation. *International Journal of Unconventional Computing*, to appear, 2013.

[15] N. R. Campbell. *Foundations of Science, The Philosophy of Theory and Experiment*. Dover, 1957.

[16] R. Carnap. *Philosophical Foundations of Physics*. Basic Books, 1966.

[17] H. von Helmholtz. *H. von Helmholtz, Epistemological Writings: 72-114, Numbering and Measuring from an Epistemological Viewpoint*. Reidel, 1977.

[18] C. G. Hempel. Fundamentals of concept formation in empirical science. *International Encyclopedia of Unified Science*, **2**, 1952.

[19] S. Jain, D. N. Osherson, J. S. Royer, and A. Sharma. *Systems That Learn. An Introduction to Learning Theory*, 2nd edn. The MIT Press, 1999.

[20] D. H. Krantz, P. Suppes, R. D. Luce, and A. Tversky. *Foundations of Measurement*. Academic Press, vol. 1 (1971), vol. 2 (1989) and vol. 3 (1990).

[21] R. D. Luce and L. Narens. Fifteen problems concerning the representational theory of measurement. In *Patrick Suppes: Scientific Philosopher. Vol. 2: Philosophy of Physics, Theory Structure, Measurement Theory, Philosophy of Language, Synthese Library 234*, P. Humphreys, ed., Kluwer Academic Publisher, 1994.

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

[23] P. Suppes. A set of independent axioms for extensive quantities. *Portugaliæ Mathematica*, **10**, 163–172, 1951.

Received 25 April 2013