On the Power of Threshold Measurements as Oracles

Edwin Beggs[1], José Félix Costa[2,3], Diogo Poças[2,3,⋆], and John V. Tucker[1]

[1] College of Science, Swansea University, Singleton Park, Swansea SA2 8PP, Wales, United Kingdom
[2] Department of Mathematics, Instituto Superior Técnico, UTL, Lisbon
[3] Centro de Matemática e Aplicações Fundamentais, University of Lisbon
diogopocas1991@gmail.com

Abstract. We consider the measurement of physical quantities that are thresholds. We use hybrid computing systems modelled by Turing machines having as an oracle physical equipment that measures thresholds. The Turing machines compute with the help of qualitative information provided by the oracle. The queries are governed by timing protocols and provide the equipment with numerical data with (a) infinite precision, (b) unbounded precision, or (c) finite precision. We classify the computational power in polynomial time of a canonical example of a threshold oracle using non-uniform complexity classes.

1 Introduction

Computation and measurement are intimately connected in all sorts of ways. Measurement is a scientific activity supported by a comprehensive axiomatic theory developed in the 20th Century using the methods of mathematical logic (see [9,11]). We are developing a new theory of measurement processes from an algorithmic perspective, in a series of papers (see [2,3,4,5,6,7,8]). At the heart of our theory is the idea that an experimenter measures a physical quantity by applying an algorithmic procedure to equipment, and that we can model *the experimenter as a Turing machine and the equipment as a device connected to the Turing machine as a physical oracle*. The Turing machine abstracts the experimental procedure, encoding the experimental actions as a program. The physical oracle model is rather versatile: for example, it accommodates using the measurements in subsequent computations and, indeed, arbitrary interactions with equipment. Some implications for the axiomatic theory have been considered in [5].[1] Case studies shape the development of the theory. The standard oracle to a Turing machine is a set that contains information to boost the power and efficiency of computation: a query is a set membership question that is answered in one time step. Experiments require queries based upon rational numbers (dyadic rationals denoted by finite binary strings).

⋆ Corresponding author.

[1] Scientific activity seen as a Turing machine can be found in computational learning theory (see [10]).

G. Mauri et al. (Eds.): UCNC 2013, LNCS 7956, pp. 6–18, 2013.

The measurement of distance taught us that oracles involve information with possible error (see [2,4,8]). The measurement of a mass taught us that oracles may take considerable time to consult (such an experiment is fully analysed in [6,8]). An important difference is the need of a cost function T of the size of the query (e.g., a timer) as part of the Turing machine. To measure the value of a physical quantity, i.e., a real number μ, 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, in a series of experimental values that converges. We call such experiments *two-sided*. Two-sided measurement is the focus of our previous work (see [8]), and that of axiomatic measurement theory (see [9]). However, not all measurements can be made this way: some quantities by their nature, or by the nature of the equipment, 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. We study threshold experiments in this paper, which are complex and are not yet addressed in the literature. We prove the following new theorems that indicate the computational power of threshold oracles and reveal differences with the less complex two-sided case:

Theorem 1.1. *(1) If a set A is decided in polynomial time by an oracle Turing machine coupled with a threshold oracle of infinite precision, then $A \in P/\log^2\star$. 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] *(2) If a set A is decided by an oracle Turing machine coupled with a threshold oracle of unbounded or fixed precision, then $A \in BPP//\log^2\star$. If a set A is in $BPP//\log\star$, then A is decided in polynomial time by a oracle Turing machine coupled with a threshold oracle.*[3,4]

Threshold oracles have not yet been considered in the literature (e.g., in [4]) and the results about two-sided oracles do not apply to these systems. Moreover, the upper bound known so far for the two-sided oracles with non-infinite precision is P/poly (except for particular types of two-sided oracles considered in [4] and [7] for which the upper bounds are P/poly and $BPP//\log\star$, respectively).

[2] 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 than or equal to n, $w \in A$ if and only if $\langle w, f(n)\rangle \in B$. The role of advices in computation theory is fully discussed e.g. in [1], Chapter 5. We use $\log^2$ to denote the class of advice functions such that $|f(n)| \in \mathcal{O}((\log(n))^2)$.

[3] $BPP//\mathcal{F}\star$ 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 $w \notin A$.

[4] Note that in experiments where the lower/upper bounds are P/poly for the infinite precision case, the unbounded comes together because $BPP//poly = P/\text{poly}$. In the threshold experiments, however, the unbounded and finite precision cases display identical power.

2 Threshold Experiments

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

2.1 The Squid Giant Motor Neurone

This first threshold experiment is inspired in the spiking neurone, such as the squid giant motor neurone, 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. 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 characterised 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.

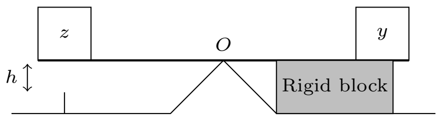

Fig. 1. Schematic representation of the *broken balance experiment*

2.3 The BBE: Broken Balance Experiment

The experiment consists of a balance scale with two pans (see Figure 1). 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 $[0, 1]$, and (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. In the context of classical, pure Newtonian mechanics of the rigid body, in the perfect Platonic world, once we assume that the test mass weighs z and the unknown mass weighs y, the cost of the experiment, $T_{exp}(z, y)$, which is the time taken for the left pan of the balance to touch the pressure stick, is given by:

$$[T_{exp}(z, y) = \tau \times \sqrt{\frac{z + y}{\max(0, z - y)}} \text{ for all } y, z \in \mathbb{R}.^5] \tag{1}$$

3 The BBE Machine as a Means to Measure Real Numbers

In what follows the suffix operation $\rfloor_n$ on a word w, $w\rfloor_n$, denotes the prefix sized n of the ω-word $w0^\omega$, no matter the size of w. Letters such as a, b, c, ..., denote constants. To the oracle Turing machine model $\mathcal{M}$ we associate a schedule $T : \mathbb{N} \to \mathbb{N}$, a time constructible function such that $T(\ell)$ steps of busy waiting of $\mathcal{M}$ are needed for the oracle to provide an answer 'YES' or 'TIMEOUT', resulting in a transition of $\mathcal{M}$ to the state q_{YES} or q_{TIMEOUT}, respectively. Everything about setting an oracle Turing machine coupled with a physical experiment as oracle can be found in [2,6].

A larger variety of experiments could have been mentioned (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., [2,6,7]), we will consider different types of precision, i.e., different communication protocols between the experimenter/Turing machine and the oracle/analogue device: (a) *infinite precision*: when the dyadic z is read in the query tape, a test mass 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 $z - 2^{-|z|} \leq z' \leq z + 2^{-|z|}$ and (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 $z - \epsilon \leq z' \leq z + \epsilon$. In the last two cases, z' is assumed to be an independent random variable, with a uniform distribution on the error interval. In what follows, $\text{Mass}(m\rfloor_\ell)$ denotes the action that triggers the BBE experiment with mass $m\rfloor_\ell$. Depending on the context, the experiment is performed either with infinite, unbounded or finite precision, as explained above.

[5] This expression for the time, specifically 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. The constant τ depends on the geometry of the lever, the value of h and the acceleration of gravity g.

Definition 3.1. *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 $1-\gamma$.*

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**
        $m := (x_0 + x_1)/2$;
        $s := \text{Mass}(m\downharpoonright_\ell)$; `% Proc. Mass is either deter. or stoch. and takes time` $T(\ell)$
        If $s =$ 'YES' **Then** $x_1 := m$ **Else** $x_0 := m$;
    End While;
Output x_0.

Fig. 2. The experimental procedure Mass takes the scheduled time $T(\ell)$, where ℓ is the size of the query and T an arbitrary time constructible function

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
        $s := \text{Mass}(1\downharpoonright_\ell)$; `% Recall that this step takes T(`$\ell$`) units of time`
        If $s =$ 'YES' **Then** $c := c + 1$;
    End Repeat;
Output c/ζ.

Fig. 3. The experimental procedure Mass is stochastic for the fixed precision case

Proposition 3.1. **(1)** *Let s be the result of Mass(m) in Figures 2 and 3, 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 $y > m - (\tau/T(|m|))^2$. In the unbounded precision scenario, (a) if the oracle Mass(m) answers 'YES', then $y < m + 2^{-|m|}$ and (b) if the oracle Mass(m) answers 'TIMEOUT', then $y > m - 2^{-|m|} - (\tau/T(|m|))^2$.* **(2)** *For any unknown mass y and any time schedule T, (a) the time complexity of algorithm of Figure 2, both in the infinite and unbounded precision scenarios, for input ℓ, is $\mathcal{O}(\ell T(\ell))$, (b1) in the infinite precision case, for all $k \in \mathbb{N}$, there exists $\ell \in \mathbb{N}$ such that $T(\ell) \geq \tau 2^{k/2}$ and the* *output is a dyadic rational m such that $|y-m| < 2^{-k}$, (b2) in the unbounded precision case, for all $k \in \mathbb{N}$, there exists $\ell \in \mathbb{N}$ such that $\ell \geq k+1$ and $T(\ell) \geq \tau 2^{(k+1)/2}$ and the output is a dyadic rational m such that $|y-m| < 2^{-k}$, and (c) moreover, both in the infinite and unbounded precision scenarios, ℓ is at most exponential in k and, if $T(k)$ is exponential in k, then the value of ℓ witnessing (b) can be taken to be linear in k.* **(3)** *For all $s \in (0,1)$, $\epsilon \in (0,1/2)$, $h \in \mathbb{N}$, and time schedule T, (a) the time complexity of algorithm of Figure 3 for input ℓ is $\mathcal{O}(2^{2\ell}T(\ell))$, (b) for all $k \in \mathbb{N}$ there exists $\ell \in \mathbb{N}$ such that $T(\ell) > \tau 2^{(k+1)/2}/\sqrt{2\epsilon}$ and thus, with probability of error 2^{-h}, the output of the algorithm is a dyadic rational m such that $|s-m| < 2^{-\ell}$, and (c) if $T(k)$ is exponential in k, then the value of ℓ witnessing the above proposition is linear in k.*

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. We can state (see [13,2]):

Proposition 3.2. **(1)** *For all unknown mass y and all time schedule 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 $1-\delta$.* **(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.*

4 Lower Bounds on the BBE Machine

We encode an advice function $(f : \mathbb{N} \to \{0,1\}^\star) \in$ *log*$\star$ into a real number $\mu(f) \in (0,1)$ by replacing every 0 by 100, every 1 by 010 and adding 001 at the end of the codes for $f(2^k)$, with $k \in \mathbb{N}$ (see Section 6(c) of [2]). These numbers belong to the Cantor set $\mathcal{C}_3$.

Theorem 4.1. *If $A \in P/\log\star$, then A is decidable by a BBE machine with infinite precision in polynomial time.*

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 $|w| \leq n$, $w \in A$ iff $\mathcal{M}'$ accepts $\langle w, f(n)\rangle$. Let $y = \mu(f)$ and T 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 3.1 (2) (b1) and (c), there is a value of ℓ, linear in k_n (and thus linear in $\lceil\log(n)\rceil$), such that $T(\ell) > \tau 2^{k_n/2}$ and so the result of running the algorithm of Figure 2 for input ℓ is a dyadic rational m such that $|y-m| < 2^{-k_n}$. Then, by Cantor $\mathcal{C}_3$ properties,[6] m and y coincide in the first

[6] 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) $|x-z| > 1/2^{m+10}$.

$k_n - 5 = 3(a+1)\lceil \log(n) \rceil + 3(b+1)$ 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 3.1 (2) (a) and the fact that $A \in P/\log\star$, the time complexity of these activities is polynomial in n. $\square$

Theorem 4.2. *If $A \in BPP//\log\star$, then* **(a)** *A is decidable by a BBE machine with unbounded precision in polynomial time and* **(b)** *A is decidable by a BBE machine with fixed precision $\epsilon \in (0, 1/2)$ in polynomial time.*

Proof: Herein we prove (a) and leave (b) to the full paper. 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 Proposition 3.2 (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 Proposition 3.2 (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 3.1 (2) (b2) and (c), there is ℓ, linear in k_n, such that the result of the algorithm of Figure 2 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, 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. 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 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 3.1 (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 $\mathcal{O}(p_1 + p_2 + p_3)$. $\square$

5 Upper Bounds on the BBE Machine

5.1 $P/\log\star$ is an Upper Bound for the Infinite Precision Case

We introduce a sequence of real numbers called *boundary numbers.* These are defined in terms of the time $T_{exp}(z, y)$ taken by the experiment for test value z and unknown value y (see the timing Equation (1) in 2.3 for an example of this), and the time schedule $T : \mathbb{N} \to \mathbb{N}$ which used to determine the output 'TIMEOUT'.

Definition 5.1. *Let $y \in (0, 1)$ be the unknown mass and T a time schedule. Then, for all $k \in \mathbb{N}$, we define $w_k \in (0, 1)$ as the number such that $T_{exp}(w_k, y) = T(k)$. We also define z_k as $z_k = w_k\downharpoonleft_k$.*

For any oracle query z of size k, (a) if $z \leq z_k$,[7] 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 is precisely the result of the algorithm for input k and as such, by knowing z_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 5.1. *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 logarithmically 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 than or equal to $a\lceil\log(n)\rceil + b$. Consider the advice function f such that $f(n)$ encodes the word $z_1\#z_2\#\cdots\#z_t$, 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. □

Observe that $w_k \searrow y$, 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} can be obtained by adding to the word z_k a very few bits of information, shortening the encoding to $\mathcal{O}(\log(n))$ bits.

[7] 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.

Theorem 5.2. *If A is a set decidable by a BBE machine with infinite precision in polynomial time and the chosen time schedule is $T(k) \in \Omega(2^{k/2})$,*[8] *then $A \in P/\log\star$.*

Proof: Since $T(k) \in \Omega(2^{k/2})$, it follows that there exist constants $\sigma, k_0 \in \mathbb{N}$ such that $T(k) \geq \sigma 2^{k/2}$, for $k \geq k_0$. By Proposition 3.1 (2) (b1) and (c), we can ensure that the value of the boundary number w_k is such that $y < w_k < y + 2^{-k+c}$, 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 as the concatenation of two strings, $z_k = x_k \cdot y_k$, where y_k has size c and x_k has size $k - c$. Note that $w_k - 2^{-k+c} < x_k < w_k$, 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. We define the function $f(n)$ as follows: (a) if $n < k_0$, then $f(n) = z_1\#z_2\# \cdots \ \#z_n n$, (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, 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 logarithmically 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 than 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 than or equal to n. Then, as in the proof of Theorem 5.1, we can devise a Turing machine that decides A in polynomial time using g as advice, witnessing that $A \in P/\log\star$. □

5.2 $BPP//\log^2\star$ 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 of size n and (b) the size of $f(n)$ grows as slowly as we can accomplish.

[8] 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 $n \geq p$, $f(n) \geq rg(n)$.

In the previous section, we made the observation that a dyadic rational z_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's. Given a threshold oracle (that is, an oracle with two possible random answers), we can depict the sequence of the answers in a binary tree, where each path is labelled 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 t and there is a real number ρ that bounds the difference in the probabilities labelling all pairs of corresponding edges in the two trees. Proposition 2.1 of [4] states that the difference in the probabilities of acceptance of the two trees is at most $2t\rho$. (Motivation from the automata theory comes from von Neumann's article [13].)

Recall the sequence of real numbers w_k such that w_k is the solution to the equation $T_{exp}(w_k, y) = T(k)$. This means that w_k is the mass in which the time taken for the experiment to return a value equals the time scheduled for an experiment with a query of size k. The numbers w_k verify two important properties. First, if we round down w_k to the first k bits, we get z_k. That is, $z_k \leq w_k < z_k + 2^{-k}$. Remember that z_k is the result of the algorithm of Figure 2 for input k, or alternatively, is the biggest dyadic rational of size k for which the result of the experiment is 'TIMEOUT' (see Proposition 3.1 (1)). The second property is that, when performing the experiment Mass(z_k) in Figure 2, since the mass z' is uniformly sampled from the interval $(z_k - 2^{-k}, z_k + 2^{-k})$, the probability of obtaining result 'YES' is precisely $(z_k + 2^{-k} - w_k)/(2 \times 2^{-k}) = 1/2 - (w_k - z_k)/(2 \times 2^{-k})$. From these facts we can conclude that, if we know the first $k + d$ bits of w_k, then we can obtain an approximation of the probability of answer 'YES' when performing experiment Mass(z_k) with an error of at most 2^{-d}. The same reasoning can be made for the experiment Mass($z_k + 2^{-k}$), 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 $1 - (w_k - z_k)/(2 \times 2^{-k})$, and this value can also be approximated by knowing the first $k + d$ bits of w_k, with an error of at most 2^{-d}. We state without proof the theorem of this section:[9]

Theorem 5.3. *If A is a set decided by a BBE machine in polynomial time with unbounded precision and exponential time schedule T, then $A \in BPP//\log^2\star$.*

5.3 $BPP//\log^2\star$ 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 5.4 has a proof that follows

[9] The proof of this theorem and of Theorem 5.4 can be found in the full paper on threshold oracles, available on demand.

the same lines of the proof of Theorem 5.3. We discuss now how the bits of the probability distribution can be computed. The numbers w_k are defined as in the beginning of Section 5.2. The following proposition is straightforward:

Proposition 5.1. *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.*

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

Our advice function will contain dyadic rational approximations of w_k and ϵ that will be used to compute approximations to $P(z)$ up to 2^{-e}, for some $e \in \mathbb{N}$ and for any dyadic rational z of size k. Let d be an integer such that $2^{-d} \leq \epsilon$, and let w'_k and ϵ' be w_k and ϵ rounded up to the first $d + e$ and $d + e + 1$ bits, respectively. We can then compute $(z - w_k)/2$ with precision 2^{-d-e-1} and $P(z)$ 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 than or equal to that of z, only a logarithmic amount of bits of $P(z)$ is required. Again, we state without proof:

Theorem 5.4. *If A is a set decided by a BBE machine in polynomial time with fixed precision and an exponential time schedule, then $A \in BPP// \log^2\star$.*

6 Conclusions

We have introduced methods to study the computational power of threshold systems, such as the neurone or photoelectric cells, for 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. We expect that hybrid systems in general cannot transcend such computational power and that this computational power stands to hybrid systems as the Church-Turing thesis is to Turing machines. Our result weakens the claims of other classes associated with models of physical systems (see, e.g., $P/$poly in [12]). In studying two-sided experiments (as in [6]), 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 mass and z the test mass. 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 two media such that there is a transmitted ray but no reflected ray. Vanishing experiments are a new type of measurement to investigate. We think that our model captures (i) the complexity of measurement and the limits to computational power of hybrid systems and (ii) the limits of what can be measured (such as in [6]). Reactions towards a gedankenexperiment, such as measuring mass as in Section 2, as an oracle can express dissatisfaction as such idealized devices cannot be built. Unfortunately, there seems to be a diffuse philosophy that considers the Turing machine an object of a different kind. Clearly, both the abstract physical experiment and the Turing machine are 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. However, the experiment can be set up to some degree of precison 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 device, are of the same ideal nature, we argue that the models allow us to study the power of adding real numbers to computing devices, characteristic of hybrid machines, and the limits of what can be measured.

Acknowledgements. The research of José Félix Costa and Diogo Poças is supported by FCT PEst – OE/MAT/UI0209/2011.

References

1. Balcázar, J.L., Días, J., Gabarró, J.: Structural Complexity I, 2nd edn. Springer (1988, 1995)
2. Beggs, E., Costa, J.F., Loff, B., Tucker, J.V.: Computational complexity with experiments as oracles. Proceedings of the Royal Society, Series A (Mathematical, Physical and Engineering Sciences) 464(2098), 2777–2801 (2008)
3. Beggs, E., Costa, J.F., Loff, B., Tucker, J.V.: On the complexity of measurement in classical physics. In: Agrawal, M., Du, D.-Z., Duan, Z., Li, A. (eds.) TAMC 2008. LNCS, vol. 4978, pp. 20–30. Springer, Heidelberg (2008)
4. Beggs, E., Costa, J.F., Loff, B., Tucker, J.V.: 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)
5. Beggs, E., Costa, J.F., Tucker, J.V.: Computational Models of Measurement and Hempel's Axiomatization. In: Carsetti, A. (ed.) Causality, Meaningful Complexity and Knowledge Construction. Theory and Decision Library A, vol. 46, pp. 155–184. Springer (2010)
6. Beggs, E., Costa, J.F., Tucker, J.V.: Limits to measurement in experiments governed by algorithms. Mathematical Structures in Computer Science 20(06), 1019–1050 (2010) Special issue on Quantum Algorithms, Venegas-Andraca, S.E. (ed.)
7. Beggs, E., Costa, J.F., Tucker, J.V.: Axiomatising physical experiments as oracles to algorithms. Philosophical Transactions of the Royal Society, Series A (Mathematical, Physical and Engineering Sciences) 370(12), 3359–3384 (2012)
8. Beggs, E., Costa, J.F., Tucker, J.V.: The impact of models of a physical oracle on computational power. Mathematical Structures in Computer Science 22(5), 853–879 (2012), Special issue on Computability of the Physical, Calude, C.S., Barry Cooper, S. (eds.)

9. Hempel, C.G.: Fundamentals of concept formation in empirical science. International Encyclopedia of Unified Science 2(7) (1952)
10. Jain, S., Osherson, D.N., Royer, J.S., Sharma, A.: Systems That Learn. An Introduction to Learning Theory, 2nd edn. The MIT Press (1999)
11. Krantz, D.H., Suppes, P., Duncan Luce, R., Tversky, A.: Foundations of Measurement. Dover (2009)
12. Siegelmann, H.T.: Neural Networks and Analog Computation: Beyond the Turing Limit. Birkhäuser (1999)
13. von Neumann, J.: Probabilistic logics and the synthesis of reliable organisms from unreliable components. In: Automata Studies, pp. 43–98. Princeton University Press (1956)