Inner structure of $BPP//\log\star$
Counting calls to an oracle

Edwin Beggs[1], Pedro Cortez, José Félix Costa[⋆2,3], and John V Tucker[1]

[1] College of Science, Swansea University, Singleton Park, Swansea, SA3 2HN, Wales, United Kingdom
e.j.beggs@swansea.ac.uk, j.v.tucker@swansea.ac.uk
[2] Department of Mathematics, Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal
pedro.cortez.91@gmail.com,fgc@math.ist.utl.pt
[3] Centro de Filosofia das Ciências da Universidade de Lisboa

Abstract. We introduce the random walk on the line as oracle to a Turing machine and prove that the number of calls to the oracle correlate with inner non-collapsing levels of $BPP//\log\star$, thus identifying meaningful hierarchies.

1 Introduction

Consider the classical model of a Turing machine with an oracle. However, the oracle is not a one step external consultation device, but it might have a time cost dependent on the size of the query. The oracle may contain either non-computable information, or computable information provided just to boost the computations of the Turing machine. Essentially, the oracle we will be considering is a real number accessed by the Turing machine through a measurement procedure such like a computer controlling a physical system.

The computational power of Turing machines having access to real numbers in polynomial time seems to have fallen from P/poly as originally proposed in [23] to $BPP//\log\star$ as in [14], even for deterministic machines, as a result of detailed investigation on protocols between analogue and digital components of computation devices (see [12,18]). The probabilistic format of $BPP//\log\star$ is due to the stochasticity of the oracle that is consulted by finite precision methods only, while infinite precision measurements bounds the power to $P/\log\star$ in deterministic polynomial time. This reduction of power in deterministic time is due to the fact that measurement takes time in non-linear systems, while in [23] the systems considered are piecewise linear.

We will consider the abstract experimenter (e.g. the experimental physicist) as a Turing machine and the abstract experiment of measuring a physical quantity (using a specified physical apparatus) as an oracle to the Turing machine (see [5,7,10,11,12,15] inter alia). The measurement algorithm running in the machine abstracts the experimental method chosen by the experimenter and proceeds by approximations.

Inspired in the work developed in [23], in [26] the authors specify hardware presumably capable of hypercomputation, namely electronic components assembled to compute supposedly a non-decidable fragment $BPP//\log\star$. In our view such systems will not support programming, since

⋆ Corresponding author

programming in such a context turns to be the settlement of a real number into the system with unbounded precision. Eventually, such systems will be capable of emergent computation due to arbitrary unknown reals (if real numbers exist in Nature) specifying their components. Emergent computational activities might well be relevant in learning tasks.

What do we intend to measure with Turing machines? Considering [26], it can be a electrical resistance, or a capacitance, or an impedance, etc (see [9]). Note that measurable numbers were first considered a scientific enterprise by Geroch and Hartle in their now famous paper [24], where they introduce the concept: *We propose, in parallel with the notion of a computable number in mathematics, that of a measurable number in a physical theory*, and measurement is a scientific activity supported by a full theory developed since the beginning of the last century as a chapter of mathematical logic, which is unexpectedly similar to oracle consultation but exhibiting new features in complexity theory (see [25]).

In our paper, we discuss how, while measuring a physical magnitude, a slight amount of bits of a real, yet unbounded for arbitrary long inputs to the machine, on the size of the input, can originate hypercomputation. The amount of bits translates each time, for inputs of size n, into a small number of calls to the oracle, which is sublogarithmic in n. We explore one of the two directions discussed in [20]. The hierarchies we will be establishing start arbitrarily close to BPP and stretch towards $BPP//\log\star$ in polynomial time.

2 Contributions of this paper

Consider the random walk experiment (RWE) of having a particle moving along an axis. The particle is sent from position $x = 0$ to position $x = 1$. Then, at each positive integer coordinate, the particle moves right, with probability σ, or left, with probability $1 - \sigma$, as outlined in Figure 1. If the particle ever returns to its initial position $x = 0$, then it is absorbed. In this process, the particle takes steps of one unit, at time intervals also of one unit, postulated to be the time step of a Turing machine transition (see [1]).

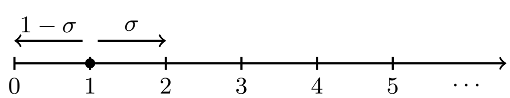

Fig. 1. Random walk on the line with absorption at $x = 0$.

We are interested in the probability that the particle is absorbed (see [21]). Let p_i be the probability of absorption when the particle is at $x = i$. In our model, the particle is launched from $x = 0$ but it only starts its random walk at $x = 1$. It is easy to see that $p_1 = (1 - \sigma) + \sigma p_2$. From $x = 2$, to be absorbed, the particle must initially move from $x = 2$ to $x = 1$ (not necessarily in one step), and then from $x = 1$ to $x = 0$ (again, not necessarily in one step). Both movements are made, independently, with probability p_1, thus, p_2 is just p_1^2. More generally, we have $p_k = p_1^k$. Therefore, the equation for the unidimensional random walk with absorption at $x = 0$ is given by the equation

$$p_1 = (1 - \sigma) + \sigma p_1^2 \ ,$$

with solutions $p_1 = 1$ and $p_1 = \frac{1-\sigma}{\sigma}$. For $\sigma = \frac{1}{2}$, the solutions coincide and $p_1 = 1$. For $\sigma < \frac{1}{2}$, the second solution is impossible, because $\frac{1-\sigma}{\sigma} > 1$, so, we must have $p_1 = 1$. For $\sigma = 1$, the particle always moves to the right, so $p_1 = 0$. Thus, for the sake of continuity of p_1, for $\sigma > \frac{1}{2}$, we must choose $p_1 = \frac{1-\sigma}{\sigma}$. Consequently, we get

$$p_1 = \begin{cases} 1 & \text{if } \sigma \leq \frac{1}{2} \\ \frac{1-\sigma}{\sigma} & \text{if } \sigma > \frac{1}{2} \end{cases} .$$

So, if $\sigma \leq \frac{1}{2}$, with probability 1 the particle always returns, but the number of steps is unbounded. In Figure 2, we illustrate this situation, for the case $\sigma = 1/4$, giving the possible locations of the particle, and the respective probabilities, after the first steps.

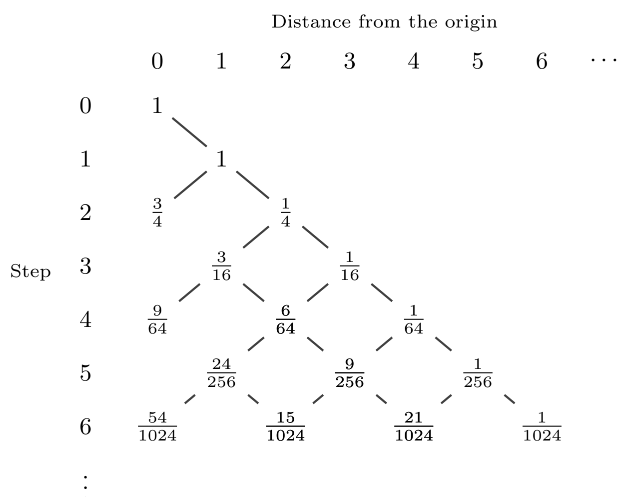

Fig. 2. Diagram showing probabilities of the particle being at various distances from the origin, for the case of $\sigma = 1/4$.

Now, let us consider a Turing machine coupled with a random walk experiment, the RW machine, as introduced in [20]. To use the RWE as an oracle, we admit that the probability σ that the particle moves forward, encodes some advice. Unlike scatter machine experiments in [5,11,16], the RWE does not need any parameters to be initialized, i.e. the Turing machine does not provide the oracle with any dyadic rational, it just "pulls the trigger" to start the experiment.

Denoting by $\log^{(k)}$ the class of advice functions f such that $|f(n)| \in O(\log^{(k)}(n))$, we prove the following:

Theorem 1. *The class of sets decidable in polynomial time by deterministic RW machines that can make up to $O(\log^{(k)}(n))$ calls to the oracle, for inputs of size n, is exactly $BPP//\log^{(k+1)}\star$.*

Bounding the number of calls to the oracle, we find a hierarchy of complexity classes within $BPP//\log\star$. Denoting the class of sets decided by deterministic RW machines that can make up to $O(\log^{(k)}(n))$ calls to the oracle by $BPP[O(\log^{(k)}(n)]$ we prove that

Theorem 2. *For every $k, k' \in \mathbb{N}$, if $k < k'$, then $BPP[O(\log^{(k)}(n)] \subseteq BPP[O(\log^{(k')}(n)]$.*

3 RW machines

For every unknown $\sigma \in (0,1)$, the time that a particle takes to be absorbed is unbounded. We introduce a constant time schedule to bound the oracle consultation time. If the particle is absorbed during that time, the finite control of the Turing machine changes to the 'yes' state, otherwise, the finite control changes to the 'no' state. The experiment has two possible outcomes and a constant time schedule.

We analyse the probability of 'yes'.

A path of the random walk is a possible sequence of moves that the particle makes until it is absorbed. Note that all such paths are made of an even number of steps. Paths of the random walk along the positive x-axis with absorption at $x = 0$ are isomorphic to a specific set of well-formed sequences of parentheses. For instance, in a random walk of length 6, the particle could behave as ((())) or (()()), where a movement to the right is represented by "(" and a movement to the left is represented by ")". The first opening parenthesis corresponds to the first move of the particle from $x = 0$ to $x = 1$. The probability of answer in 6 steps is the sum of two probabilities corresponding to the two possible paths. All paths of a certain length have the same probability; namely, for every even number n, the probability of each path of length n is

$$\sigma^{\frac{n}{2}-1}(1-\sigma)^{\frac{n}{2}} \ .$$

Therefore, we only need to know the number of possible paths for each length, i.e. the number of well-formed sequences of parentheses satisfying some properties. In [4], the authors generalize the Catalan numbers and prove the following interesting result:

Proposition 1 (Blass and Braun [4]). *For every $\ell, w \in \mathbb{Z}$, $\ell \geq w \geq 0$, let X be the number of strings consisting of ℓ left and ℓ right parentheses, starting with w consecutive left parentheses, and having the property that every nonempty, proper, initial segment has strictly more left than right parentheses. Then*

$$X = \frac{w}{2\ell - w} \binom{2\ell - w}{\ell}$$

Note that when $w = \ell = 0$, the undefined fraction $w/(2\ell - w)$ is to be interpreted as 1, since this gives the correct value $X = 1$, corresponding to the empty string of parentheses. From this proposition, we derive the probability $q(t)$ that the particle is absorbed in even time $t+1$, for $t \geq 1$. It suffices to take $\ell = (t+1)/2$ and $w = 1$:

$$q(t) = \frac{1}{t}\binom{t}{\frac{t+1}{2}}(1-\sigma)^{\frac{t+1}{2}}\sigma^{\frac{t+1}{2}-1} \ .$$

Therefore, the probability that the particle is absorbed during the time schedule T is given by

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

This is the probability of getting the outcome 'yes' from the oracle. Figure 3 allows us to understand the behaviour of the probability $F(\sigma, T)$ as a function of σ. We see that, as T increases,

$F(\sigma, T)$ increases as well, since the longer the machine waits, the more likely it is that a particle is absorbed. We can also see that as T approaches infinity, $F(\sigma, T)$ approaches the probability p_1 that the particle is absorbed, which makes sense, since p_1 represents a probability of absorption with unbounded time. For analytical reasons, we will consider only $\sigma \in [\frac{1}{2}, 1]$, corresponding to a variation of p_1 from 1 to 0. Note that we could consider any interval contained in $[0, 1]$. For every T, this probability is a function of σ that satisfies the following conditions: (a) $F(\bullet, T) \in C^1([\frac{1}{2}, 1])$, (b) for every $\sigma \in [\frac{1}{2}, 1]$, $F'(\sigma, T) \neq 0$ and (c) n bits of $F(\bullet, T)$ are computable in time $O(2^n)$. Once this conditions are verified the following theorem holds (see proof in the forthcoming paper [17]):

Theorem 3. *For every set A, $A \in BPP//\log\star$ if and only if it is decidable by a RW machine in polynomial time.*

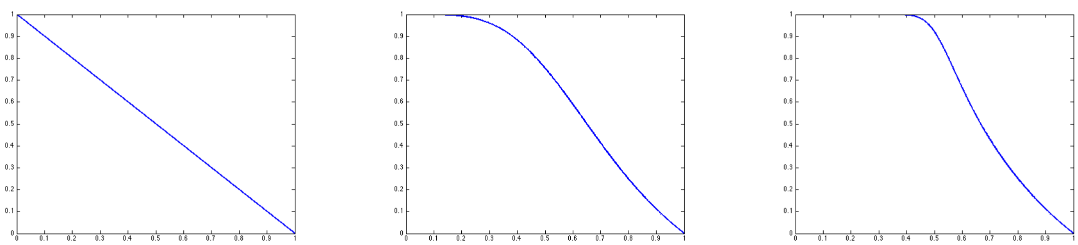

Fig. 3. Graphs of $F(\sigma, T)$ for $T = 2$, $T = 10$ and $T = 100$.

4 Computational resources

Consider that we have a limiting number of particles that the RW machine can launch, i.e. a bound in the number of oracle calls that the machine can make. We study now how the precision in the measurement of σ depends on the number of oracle calls.

Theorem 4. *A RW machine that can make up to $\xi(n)$ calls to the oracle, on input w of size $|w| = n$, can read $O(\log(\xi(n)))$ bits of the unknown parameter σ in polynomial time in n.*

Proof. We know that each particle has probability of absorption $F(\sigma, T)$ in time T. Thus, if we make $\xi(n)$ oracle calls on an input of size n, the number of times α that the experiment returns 'yes' is a random variable with binomial distribution. Let us consider $X = \alpha/\xi(n)$, the random variable that represents the relative frequency of absorption ('yes'). We have the expected value $\mathbb{E}[X] = \mathbb{E}[\alpha]/\xi(n) = \xi(n)F(\sigma, T)/\xi(n) = F(\sigma, T)$ and the variance $\mathbb{V}[X] = \mathbb{V}[\alpha]/\xi(n)^2 = \xi(n)F(\sigma, T)(1 - F(\sigma, T))/\xi(n)^2 = F(\sigma, T)(1 - F(\sigma, T))/\xi(n)$. Chebyshev's inequality states that, for every $\delta > 0$,

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

Let k be the number of bits of σ to be read.[4] By setting $\delta = 2^{-k-5}$, we get

$$P(|X - F(\sigma, T)| > 2^{-k-5}) \leq \frac{2^{2k+10}F(\sigma, T)}{\xi(n)}$$

[4] It is proved in [12,16] that, for every $\sigma \in \mathcal{C}_3$ and for every dyadic rational z, if $|\sigma - z| \leq 2^{-k-5}$, then the binary expansions of x and z coincide on the first k bits.

and if we want an error probability of at most γ, we set

$$\frac{2^{2k+10}F(\sigma,T)}{\xi(n)} \leq \gamma$$

Applying logarithms, we get

$$2k + 10 + \log(F(\sigma,T)) - \log(\xi(n)) \leq \log(\gamma) \ ,$$

therefore,

$$k \leq \frac{\log(\xi(n)) + \overbrace{\log\left(\frac{1}{F(\sigma,T)}\right) - (10 + \frac{1}{\log(\gamma)})}^{\text{constant value}}}{2} \ .$$

For every σ, $F(\sigma,T)$ increases with T and the term $\log(1/F(\sigma,T))$ decreases; contrary to what one might expect, for every input word w of size n, the longer we wait for the particles to return, the less precision we can obtain for σ.[5] We establish that in every oracle call the machine will wait exactly two time steps for the particle to return ($T = 2$). Therefore, $F(\sigma,2) = (1-\sigma)$. Now, with $k \in O(\log(\xi(n)))$, we have

$$P(|(1-X) - \sigma| = P(|X - (1-\sigma)| > 2^{-k-5}) \leq \gamma \ .$$

With value $1 - X$ we can estimate σ. □

This result suggests a non-collapsing hierarchy of classes defined by the magnitude of the number of queries to the oracle.

5 Lower and upper bounds

We encode advice functions in order to compare RW machines with Turing machines with advice. We define the iterated logarithmic functions $\log^{(k)}(n)$:

- $\log^{(0)}(n) = n$;
- $\log^{(k+1)}(n) = \log(\log^{(k)}(n))$.

Similarly, we define the iterated exponential $\exp^{(k)}(n)$:

- $\exp^{(0)}(n) = n$;
- $\exp^{(k+1)}(n) = 2^{\exp^{(k)}(n)}$.

The iterated exponential is a well known bound on the number of computation steps of elementary functions (e.g. see [22]). For every $k \in \mathbb{N}$, the functions $\log^{(k)}$ and $\exp^{(k)}$ are inverse of each other. Let $\log^{(k)}$ also denote the class of advice functions f such that $|f(n)| \in O(\log^{(k)}(n))$.

Let $c(w)$ be the encoding of a single word w. We define the encoding $y(f) = \lim y(f)(n)$ for an advice function $f \in \log^{(k)}\star$ in the following way:

[5] This statement makes sense, since, if we wait too long, then we will lose information about the absorption time of the particle.

- $y(f)(0) = 0.c(f(0))$;
- if $f(n+1) = f(n)s$, then

$$y(f)(n+1) = \begin{cases} y(f)(n)c(s) & \text{if } n+1 \text{ is not of the form } \exp^{(k)}(m) \\ y(f)(n)c(s)001 & \text{if } n+1 \text{ is of the form } \exp^{(k)}(m) \end{cases}$$

So, for example, if we want to encode a function $f \in \log\log\star$, we just have to place the separator 001 when $n+1$ is of the form 2^{2^m}, for some $m \in \mathbb{N}$.

For every k and for every $f \in \log^{(k)}\star$, we have that $y(f) \in \mathcal{C}_3$. Also, for every n, in order to extract the value of $f(n)$, we only need to find the number $m \in \mathbb{N}$ such that $\exp^{(k)}(m-1) < n \leq \exp^{(k)}(m)$ and then read $y(f)$ in triplets, until we find the $(m+1)$-th separator. Then, it is only needed to ignore the separators and replace each 100 triplet by 0 and each 010 triplet by 1. Since $f \in \log^{(k)}\star$, we know that $|f(\exp^{(k)}(m))| = O(\log^{(k)}(\exp^{(k)}(m))) = O(m)$. we conclude that $3O(m) + 3(m+1) = O(m)$ bits are enough to get the value of $f(\exp^{(k)}(m))$ and, consequently, $O(\log^{(k)}(n))$ bits to get the value of $f(n)$.

Theorem 5. *[Lower bounds] For every k, every set in $BPP//\log^{(k+1)}\star$ is decidable in polynomial time by a RW machine that can make up to $\xi(n) = O(\log^{(k)}(n))$ oracle calls on inputs of size n.*

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

Let $\mathcal{M}'$ be a RW machine with unknown parameter $y(f)$, the encoding of f, and let $\gamma_2 \in \mathbb{R}$ be such that $\gamma_1 + \gamma_2 < 1/2$. Let w be a word such that $|w| \leq n$. Theorem 4 assures that $\mathcal{M}'$ can estimate $O(\log(\xi(n))) = O(\log(\log^{(k)}(n))) = O(\log^{(k+1)}(n))$ bits of $y(f)$, and, thus, $\mathcal{M}'$ can read $f(n)$ in scheduled protocol time $T = 2$ and in machine polynomial time, with an error probability bounded by γ_2. We have that $P(\text{'}yes\text{'}) = 1 - \sigma$ and $P(\text{'}no\text{'}) = \sigma$. Let γ_3 be such that $\gamma_1 + \gamma_2 + \gamma_3 < 1/2$. Using calls to the same oracle, $\mathcal{M}'$ can produce a sequence of fair coin tosses. In fact, with probability at least $1 - \gamma_3$, we can use a sequence of independent biased coin tosses of length

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

to produce a sequence of s independent fair coin tosses (see the proof in [16,5]). Therefore, $\mathcal{M}'$ can decide A in polynomial time, with error probability bounded by $\gamma_1 + \gamma_2 + \gamma_3 < 1/2$. □

In order to state and prove upper bounds, we need the following auxiliary lemma:

Theorem 6. *Let A be the set decided by RW machine $\mathcal{M}$ with unknown parameter σ that can make up to $\xi(n)$ calls to the oracle, for inputs of size n, with error probability bounded by $\gamma < 1/4$. If $\mathcal{M}'$ is an identical AD machine, with unknown parameter $\tilde{\sigma}$ and the probability of absorption $\tilde{F}$, such that*

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

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

Proof. We know that $\mathcal{M}$ and $\mathcal{M}'$ make at most $\xi(n)$ calls to the oracle, in such a way that the query tree $\mathcal{T}$ associated to both, has maximum depth $\xi(n)$. Let w be of size not greater than n. Let D be the assignment of probabilities to the edges of $\mathcal{T}$ corresponding to the unknown parameter σ and 'yes' probability $F(\sigma, T)$ and D' be the assignment of probabilities given by the unknown parameter $\tilde{\sigma}$ and 'yes' probability $\tilde{F}(\tilde{\sigma}, T)$. Since $|F(\sigma, T) - \tilde{F}(\tilde{\sigma}, T)| < 1/8\xi(n)$, the difference between any particular probability is at most

$$\kappa = \frac{1}{8\xi(n)} \ .$$

Invoking Proposition 11 of [16], we have two different cases:

- $w \notin A$: In this case, an incorrect result corresponds to $\mathcal{M}'$ accepting w. The probability of acceptance $P_A(\mathcal{T}, D')$ for $\mathcal{M}'$ is

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

- $w \in A$: In this case, an incorrect result corresponds to $\mathcal{M}'$ rejecting w. The probability of rejection $P_R(\mathcal{T}, D')$ for $\mathcal{M}'$ is

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

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

Let $F(\sigma, T)\lfloor_m$ denote the first m bits of the probability $F(\sigma, T)$. The next theorem is a corollary of the previous:

Theorem 7. *Let A be the set decided by RW machine $\mathcal{M}$ with unknown parameter σ that can make up to $\xi(n)$ calls to the oracle, for inputs of size n, with error probability bounded by $\gamma < 1/4$. If $\mathcal{M}'$ is an identical analogue-digital machine, with unknown parameter $\tilde{\sigma}$, but with the exception that the probability that the oracle returns 'yes' is given by $F(\sigma, T)\lfloor_{\log \xi(n)+3}$, then $\mathcal{M}_n$ decides the same set as $\mathcal{M}$, also in time $t(n)$, but with error probability bounded by $3/8$.*

Now we state and prove upper bounds.

Theorem 8. *[Upper bounds] For every k, every set decided in polynomial time by a RW machine that can make up to $\xi(n) = O(\log^{(k)}(n))$ calls to the oracle, where n is the size of the input, is in $BPP//\log^{(k+1)}\star$.*

Proof. Let A be a set decided in polynomial time $p(n)$ and with error probability bounded by $1/4$ by a RW machine $\mathcal{M}$ with unknown parameter σ that can make up to $\xi(n) = O(\log^{(k)}(n))$ calls to the oracle. We specify a probabilistic Turing machine $\mathcal{M}'$ with advice $f(n) = F(\sigma, T)\lfloor_{\log \xi(n)+3}$ to decide A. We have $f \in \log^{(k+1)}\star$.

By Theorem 7, we know that an RW machine with 'yes' probability $f(n)$ decides the same than $\mathcal{M}$ for words of size $\leq n$, but with error probability $\leq 3/8$. The value $f(n) = F(\sigma)\lfloor_{\log \xi(n)+3}$ is a dyadic rational with denominator $2^{\log \xi(n)+3}$. Thus, $m = 2^{\log \xi(n)+3} f(n) \in [0, 2^{\log \xi(n)+3})]$ is an integer. Consider $\kappa = \log \xi(n)+3$ fair coin tosses, interpreted as a sequence of bits. The machine $\mathcal{M}'$ then tests if $\tau_1\tau_2 \ldots \tau_k < m$, where $\tau_1\tau_2 \ldots \tau_k$ is now interpreted as an integer. If the test is true, the machine returns 'yes', otherwise it returns 'no'. The probability of returning 'yes' is $m/2^k = f(n)$, as required. The time taken is polynomial in n. □

6 The hierarchy

From Theorem 4 and Theorem 8, we get the following corollary:

Theorem 9. *The class of sets decidable in polynomial time by RW machines that can make up to $O(\log^{(k)}(n))$ calls to the oracle, for inputs of size n, is exactly $BPP//\log^{(k+1)}\star$.*

As we want the RW machines to run in polynomial time, the maximum number of oracle calls that we can allow is polynomial. For that bound, the corresponding class is $BPP//\log\star$. Thus, if we restrict more and more the number of queries to the oracle, we can obtain a fine structure of $BPP//\log\star$. Observe that if k is a very large number, the machine is allowed to make only few calls to the oracle, but the advice is smaller, so the number of bits that the machine needs to read is also smaller.

We explore some properties of advice classes (see [3], [23], and [6]).

If $f : \mathbb{N} \to \Sigma^*$ is an advice function, then we use $|f|$ to denote its size, i.e., the function $|f| : \mathbb{N} \to \mathbb{N}$ such that $|f|(n) = |f(n)|$, for every $n \in \mathbb{N}$. For a class of functions, $\mathcal{F}$, $|\mathcal{F}| = \{|f| : f \in \mathcal{F}\}$.

Definition 1. *A class of advice functions is said to be a class of reasonable advice functions if:*

1. *for every $f \in \mathcal{F}$, $|f|$ is computable in polynomial time;*
2. *for every $f \in \mathcal{F}$, $|f|$ is bounded by a polynomial;*
3. *for every $f \in \mathcal{F}$, $|f|$ is increasing;*
4. *$|\mathcal{F}|$ is closed under addition and multiplication by positive integers;*
5. *for every polynomial p of positive integer coefficients and every $f \in \mathcal{F}$, there exists $g \in \mathcal{F}$ such that $|f| \circ p \leq |g|$.*

Definition 2. *Let r and s be two total functions. We say that $r \prec s$ if $r \in o(s)$. Let $\mathcal{F}$ and $\mathcal{G}$ be classes of advice functions. We say that $\mathcal{F} \prec \mathcal{G}$ if there exists a function $g \in \mathcal{G}$ such that, for every $f \in \mathcal{F}$, $|f| \prec |g|$.*

We have $\log^{(k+1)} \prec \log^{(k)}$, for all $k \geq 0$. Now, we just need to know the relation between the non-uniform complexity classes of BPP, induced by the relation $\prec$ in the advice classes. Remember that a set is said to be tally if it is a language over an alphabet of a single symbol (e.g. $\{0\}$). Now, consider the set of finite sequences over the alphabet Σ ordered first by size and then alphabetically. The characteristic function of a set $A \subseteq \Sigma^*$ is the unique infinite sequence $\chi_A : \mathbb{N} \to \{0, 1\}$ such that, for every n, $\chi_A(n)$ is 1 if, and only if, the n-th word in that order is in A. The characteristic function of a tally set A is a sequence where the i-th bit is 1 if, and only if, the word 0^i is in A. The following theorem generalizes the related theorem of [3], [23] and [6], where it is proved for the deterministic case.

Theorem 10. *If $\mathcal{F}$ and $\mathcal{G}$ are two classes of reasonable sublinear advice functions* [6] *such that $\mathcal{F} \prec \mathcal{G}$, then $BPP//\mathcal{F} \subsetneq BPP//\mathcal{G}$.*

Proof. Trivially, $BPP//\mathcal{F} \subseteq BPP//\mathcal{G}$. Let *linear* be the set of advice functions of size linear in the size of the input and η.*linear* be the class of advice functions of size ηn, where n is the size of the input and η is a number such that $0 < \eta < 1$. There is an infinite sequence γ whose set of prefixes is in $BPP//linear$ but not in $BPP//\eta.linear$ for some η sufficiently small.[7]. Let $g \in \mathcal{G}$ be a function such that, for every $f \in \mathcal{F}$, $|f| \prec |g|$. We prove that there is a set in $BPP//g$ that does not belong to $BPP//f$, for any $f \in \mathcal{F}$.

A tally set T is defined in the following way: for each $n \geq 1$,

$$\beta_n = \begin{cases} \gamma\lfloor_{|g|(n)}\ 0^{n-|g|(n)} & \text{if } |g|(n) \leq n \\ 0^n & \text{otherwise} \end{cases} .$$

T is the tally set with characteristic string $\beta_1\beta_2\beta_3\cdots$. With advice $\gamma_{|g|(n)}$, it is easy to decide T, since we can reconstruct the sequence $\beta_1\beta_2\cdots\beta_n$, with $(n^2+n)/2$ bits, and then we just have to check if its n-th bit is 1 or 0. We conclude that $T \in P/g \subseteq BPP//g$.

We prove that the same set does not belong to $BPP//f$. Suppose that some probabilistic Turing machine $\mathcal{M}$ with advice f, running in polynomial time, decides T with probability of error bounded by [8]

$$2^{-\log(4|g|(n))} = \frac{1}{4|g|(n)}$$

Since $|f| \in o(|g|)$, then, for all but finitely many n, $|f|(n) < \eta|g|(n)$, for arbitrarily small η, meaning that we can compute, for all but finitely many n, $|g|(n)$ bits of γ using an advice of length $\eta.|g|(n)$, contradicting the fact that the set of prefixes of γ is not in $BPP//\eta.linear$. The reconstruction of the binary sequence $\gamma\lfloor_{|g|(n)}$ is provided by the following procedure:

Procedure
  Begin
    Input n;
    $x := \lambda$;
    Compute $|g|(n)$;
    For $i := \frac{n^2-n}{2}$ **To** $\frac{n^2-n}{2} + |g|(n)$ **Do Begin**
      Query 0^i to T by running machine $\mathcal{M}$ with advice $f(i)$;
      If "YES" **Then** $x := x1$ **Else** $x := x0$;
    End For;
    Output x
  End.

[6] $\mathcal{F}$ is a class of reasonable sublinear advice functions if it is a class of reasonable advice functions such that, for every $f \in \mathcal{F}$, $|f| \in o(n)$.

[7] We can take for γ the Chaitin Omega number, Ω

[8] E.g. see Proposition 6.17 in [2]. The probability of error of a given probabilistic machine that decides T in polynomial time can be reduced below any fixed value just by iteration.

The queries are made simulating machine $\mathcal{M}$ which is a probabilistic Turing machine with error probability bounded by $2^{-\log(4|g|(n))} = \frac{1}{4|g|(n)}$. Thus, the probability of error of $\mathcal{M}'$ is bounded by

$$\frac{1}{4|g|(\frac{n^2-n}{2})} + \cdots + \frac{1}{4|g|(\frac{n^2-n}{2} + |g|(n))} \; .$$

As $|g|$ is increasing, the error probability is bounded by

$$\frac{1}{4|g|(\frac{n^2-n}{2})} \times |g|(n),$$

which, for $n \geq 3$, is bounded by

$$\frac{1}{4|g|(n)} \times |g|(n) = \frac{1}{4} \; .$$

□

As we are considering prefix advice classes, it is useful to derive the following corollary:

Theorem 11. *If $\mathcal{F}$ and $\mathcal{G}$ are two classes of reasonable sublinear advice functions such that $\mathcal{F} \prec \mathcal{G}$, then $BPP//\mathcal{F}\star \subsetneq BPP//\mathcal{G}\star$.*

Proof. The proof of 10 is also a proof that $BPP//\mathcal{F} \subsetneq BPP//\mathcal{G}\star$, because the advice function used is $\gamma\lfloor_{|g|(n)}$, which is a prefix advice function. Since $BPP//\mathcal{F}\star \subseteq BPP//\mathcal{F}$, the statement follows. □

We have already seen that, for all $k \geq 0$, $\log^{(k+1)} \prec \log^{(k)}$. In particular, this is true for $k \geq 1$ and we have the following infinite descending chain

$$\cdots \prec \log^{(4)} \prec \log^{(3)} \prec \log^{(2)} \prec \log \, .$$

Therefore, by Theorem 11, we have also the descending chain of sets

$$\cdots \subsetneq BPP//\log^{(4)}\star \subsetneq BPP//\log^{(3)}\star \subsetneq BPP//\log^{(2)}\star \subsetneq BPP//\log\star \; ,$$

that, according with Theorem 9, coincide with the classes of sets decided by RW machines that can make up to

$$\cdots \subsetneq O(\log^{(3)}(n)) \subsetneq O(\log^{(2)}(n)) \subsetneq O(\log(n)) \subsetneq O(n)$$

calls to the oracle, respectively.

Note that, although we have defined $\log^{(0)}(n) = n$, the non-uniform complexity class $BPP//\log\star$ corresponds to the class of sets decided by RW machines with a polynomial number of oracle calls, and not only $O(n)$ calls.

7 Conclusion

We introduced RW machines specified as Turing machines having access to a random walk experiment on a line. We then proved that the class of sets decidable in polynomial time by RW machines that can make up to $O(\log^{(k)}(n))$ calls to the oracle is exactly $BPP//\log^{(k+1)}\star$, where $\log^{(k)}$ is the class of advice functions f such that $|f(n)| \in O(\log^{(k)}(n))$.

We proved that, if $\mathcal{F}$ and $\mathcal{G}$ are two classes of reasonable sublinear advice functions such that $\mathcal{F} \prec \mathcal{G}$, then $BPP//\mathcal{F} \subsetneq BPP//\mathcal{G}$. Although this result was already discussed for the deterministic case in [3,6,23], the probabilistic case seems not to have been considered.

Then, we presented a fine structure of $BPP//\log\star$ based on counting oracle calls:

$$\cdots \subsetneq BPP//\log^{(4)}\star \subsetneq BPP//\log^{(3)}\star \subsetneq BPP//\log^{(2)}\star \subsetneq BPP//\log\star,$$

that coincide with the structure of classes of sets decided by RW machines that can make up to

$$\cdots \subsetneq O(\log^{(3)}(n)) \subsetneq O(\log^{(2)}(n)) \subsetneq O(\log(n)) \subsetneq O(n)$$

calls to the oracle, respectively.

8 Open problem

Together with the transfinite chain of advice classes presented in [6] and [19], we also have a transfinite chain of non-uniform probabilistic classes:

$$\cdots \subsetneq BPP//\log^{(2\omega)}\star \subsetneq \cdots \subsetneq BPP//\log^{(\omega)}\star \subsetneq \cdots \subsetneq BPP//\log^{(2)}\star \subsetneq BPP//\log\star.$$

In fact, the chain of non-uniform classes can be continued, where $\log^{(\omega)} = \bigcap_{k\in\mathbb{N}} \log^{(k)}$ is a non-empty class (as shown in [6,19] for diverse transfinite classes). However, we do not know if there is a correspondence between these complexity classes and the classes decided by RW machines with bounded number of oracle calls, since we only proved such a correspondence for advice classes of the form $\log^{(k)}$, with $k \in \mathbb{N}$. At present, we do not know how to encode a function $f \in \log^{(\omega)}\star$ into a real number.

Acknowledgements. The research of José Félix Costa is supported by Fundação para a Ciência e Tecnologia, projeto FCT I.P.:UID/FIL/00678/2013.

References

1. Salvador Elias Venegas-Andraca. *Quantum Walks for Computer Scientists.* Morgan and Claypool Publishers, 2008.
2. José Luis Balcázar, Josep Días, and Joaquim Gabarró. *Structural Complexity I.* Springer-Verlag, 2nd edition, 1988, 1995.
3. José Luis Balcázar, Ricard Gavaldà, and Hava T. Siegelmann. Computational power of neural networks: a characterization in terms of kolmogorov complexity. *IEEE Transactions on Information Theory*, 43(4):1175–1183, July 1997.
4. Andreas Blass and Gábor Braun. Random orders and gambler's ruin. *Electr. J. Comb.*, 12:R23, 2005.
5. Edwin Beggs, José Félix Costa, Bruno Loff, and John V. Tucker. Computational complexity with experiments as oracles. *Proceedings of the Royal Society, Series A (Mathematical, Physical and Engineering Sciences)*, 464(2098):2777–2801, 2008.
6. Edwin Beggs, José Félix Costa, Bruno Loff, and John V. Tucker. Oracles and advice as measurements. In Cristian S. Calude, José Félix Costa, Rudolf Freund, Marion Oswald, and Grzegorz Rozenberg, editors, *Unconventional Computation (UC 2008)*, volume 5204 of *Lecture Notes in Computer Science*, pages 33–50, Springer-Verlag, 2008.

7. Edwin Beggs, José Félix Costa, Bruno Loff, and John V. Tucker. Computational complexity with experiments as oracles II. Upper bounds. *Proceedings of the Royal Society, Series A (Mathematical, Physical and Engineering Sciences)*, 465(2105):1453–1465, 2009.
8. Edwin Beggs, José Félix Costa, and John V. Tucker. Computational models of measurement and Hempel's axiomatization. In A. Carsetti, editor, *Causality, Meaningful Complexity and Embodied Cognition*, volume 46 of *Theory and Decision Library A*, pages 155–183. Springer, 2010.
9. Edwin Beggs, José Félix Costa, and John V. Tucker. Physical oracles: The Turing machine and the Wheatstone bridge. *Studia Logica*, 95(1-2):279–300, 2010.
10. Edwin Beggs, José Félix Costa, and John V. Tucker. Limits to measurement in experiments governed by algorithms. *Mathematical Structures in Computer Science*, 20(06):1019–1050, 2010.
11. Edwin Beggs, José Félix Costa, and John V. Tucker. The impact of models of a physical oracle on computational power. *Mathematical Structures in Computer Science*, 22(5):853–879, 2012.
12. Edwin Beggs, José Félix Costa, Diogo Poças, and John V. Tucker. Oracles that measure thresholds: the Turing machine and the broken balance. *Journal of Logic and Computation*, 23(6):1155–1181, 2013.
13. Edwin Beggs, José Félix Costa and John V. Tucker. A natural computation model of positive relativisation. *International Journal of Unconventional Computing*, 10(1–2):111–141, 2013.
14. Edwin Beggs, José Félix Costa, Diogo Poças, and John V. Tucker. An analogue-digital Church-Turing thesis. *Int. J. Found. Comput. Sci.*, 25(4):373–390, 2014.
15. Edwin Beggs, José Félix Costa, and John V. Tucker. Three forms of physical measurement and their computability. *The Review of Symbolic Logic*, 7:618–646, 12 2014.
16. Tânia Ambaram, Edwin Beggs, José Félix Costa, Diogo Poças, and John V. Tucker. An analogue-digital model of computation: Turing machines with physical oracles. In Andrew Adamatzky (editor): *Advances in Unconventional Computing*, volume 1 (theory), 38pp, Springer, September 2016, to appear.
17. Edwin Beggs, Pedro Cortez, José Félix Costa, and John V. Tucker. Classifying the computational power of stochastic physical oracles. Submitted, May 2016.
18. Edwin Beggs, José Félix Costa, Diogo Poç as, and John V. Tucker. Computations with oracles that measure vanishing quantities. *Mathematical Structures in Computer Science*, in print.
19. José Félix Costa. Incomputability at the foundations of physics (A study in the philosophy of science). *J. Log. Comput.*, 23(6):1225–1248, 2013.
20. José Félix Costa. Uncertainty in time. *Parallel Processing Letters*, 25(1), 2015.
21. Frederick Mosteller. *Fifty Challenging Problems in Probability with Solutions*. Dover Publications, 1987.
22. Piergiorgio Odifreddi. *Classical Recursion Theory II*. Studies in Logic and the Foundations of Mathematics. North Holland, 1999.
23. Hava T. Siegelmann. *Neural Networks and Analog Computation: Beyond the Turing Limit*. Birkhäuser, 1999.
24. Geroch, R. and Hartle, J. B. (1986). Computability and Physical Theories. *Foundations of Physics*, 16(6):533–550, 1986.
25. David H. Krantz, Patrick Suppes, R. Duncan Luce, and Amos Tversky. *Foundations of Measurement*. Academic Press, vol. 1 (1971), vol. 2 (1989) and vol. 3 (1990).
26. Arthur Steven Younger, Emmett Redd, and Hava T. Siegelmann. Development of physical superturing analog hardware. In *Unconventional Computation and Natural Computation - 13th International Conference, UCNC 2014, London, ON, Canada, July 14-18, 2014, Proceedings*, pages 379–391, 2014.

Index