Complexity with costing and stochastic oracles
Extended abstract

Diogo Miguel Ferreira Poças
Instituto Superior Técnico - Universidade de Lisboa

July 13, 2013

Abstract

The purpose of this dissertation is to study the computational power obtained when combining a digital device, such as the Turing machine, with an analog device, such as a physical experiment seen as an oracle to the machine.

We define the components of such hybrid systems, which we call analog-digital Turing machine, and analyze their computational power subject to polynomial resources. We investigate three types of oracles motivated by experiments in the physical world, namely two-sided oracles, threshold oracles and vanishing oracles. The oracle provides qualitative information that helps the computation of the Turing machine.

We also consider variations of the protocol between the Turing machine and the oracle, such that queries can be translated into real numbers with infinite precision, unbounded precision or finite precision. For each case we state and prove lower and upper bounds on the complexity classes computed by such systems.

1 Introduction

The main focus of computability and complexity theories is to study the limits of what can be computed or done in a feasible way. The main tool for studying these objects is the algorithm – a sequence of instructions, executable in an unambiguous way by a machine, allowing to decide, for example, membership to a given set. A property of these algorithms is that they operate on a discrete set of data, using unbounded (yet finite) resources.

Physical experiments, on the other hand, operate on a real world, which can be seen as continuous. However, the physical experiment can also be seen as a sequence of instructions, departing from an initial set of data and materials, and reaching an experimental result. The idea that physical experimentation can be seen as algorithmic is nicely presented in [GH86], where the notion of measurable number is proposed in parallel with the notion of computable number.

In this dissertation we intend to bring together computability theory and physical experiments. We may imagine a model of computation based on a Turing machine with the possibility of performing physical experiments during a computation. We call this model the analog-digital Turing machine; the digital component is given by the Turing machine; the analog component is given by the physical experiment.

The thesis of analog computation preceded digital computation. The first analog computers were used to solve differential equations and provided a great help in gunfire control during World War II. However, digital computation emerged and improved greatly in the last half century, surpassing analog computation in present days. Despite this, analog computation is regaining interest. Several examples of recent results include the optical computer for factorization ([Bla07]), the optical computer of

Woods, Naughton ([WN05]), the dynamic and hybrid systems of Koiran, Cosnard, Garzon ([KCG94]) and Bournez ([BC96]) and also the neural networks of Siegelmann and Sontag ([Sie99, Sie95, SS95]). Most notably, in the work of Siegelmann and Sontag an analogy to the Church-Turing thesis was proposed, stating that analog devices do not extend polynomial-time computation beyond the non-uniform limit of $P/poly$.

We found that a good way to articulate the interaction between the Turing machine and the physical experiment is the theory of relative computation, i.e, computation with oracles. In modelling the physical experiment as an oracle, we are in fact implying that this dissertation could be seen as a study of the classes of sets decidable by some particular families of oracles. We could in fact make this study abstracting completely from the concept of physical experiment. However, the physical experiment is useful, not only as a motivating example of the theory, but also as a help in the understanding of the proof and techniques used. In addition, most of the results obtained are valid for a general family of oracles, including but not limited to the physical experiment that we use as example.

The oracles used for representing physical experiments possess certain features: they model certain binary relations; they possess a computational consultation cost; and they may be stochastic. The two latter features represent a shift from the usual definition of oracles. In this dissertation all the information relevant to one particular oracle may be condensed in a real number; in this way, we are basically working with a family of analog-digital devices that is parametrized by one real value.

We now present a summary of the development of the theory of analog-digital Turing machines. In [BT07], Beggs and Tucker present the Scatter Machine Experiment, an experiment in Newtonian kinematics that can measure a real number encoding the position of a triangular wedge. Later in [BCLT08, BCLT09], Costa suggested a new idea and introduced the concept of analog-digital Turing machine, coupling the Scatter Machine Experiment with a Turing machine. It was shown that these machine could decide sets in $P/poly$ or $BPP//log\star$, depending on the precision considered. For the case of infinite precision they have thus obtained the same class as Siegelmann and Sontag did with their neural networks.

In [BCT10a] a new type of experiment was studied, the Collider Machine Experiment. This experiment diverged from the previous experiment in a fundamental way. The time taken for an experiment is not constant, and increases with the number of desired digits of precision. Thus, the concept of time schedule was introduced to deal with the growing costs of the experiment. As a consequence, the lower bounds of decidable sets dropped from $P/poly$ to $P/log\star$ and $BPP//log\star$, depending on the precision considered. In further papers [BCT12b] new experiences were studied with the same results.

The above experiments could be characterized as two-sided, in the sense that we could approach the unknown value either by above or below. For this reason the techniques used were essentialy the same. However, in [BCT10c] two new kinds of experiments came across, namely threshold experiments (exemplified by the Rutherford Experiment) and vanishing experiments (exemplified by the Brewster Angle Experiment). It was hinted that these experiments could boost the computational power of a Turing machine to $P/log\star$ and $BPP//log\star$, but for these experiments the previous techniques were not applicable.

The main contributions from this work are as follows. First, a general form of the techniques for proving lower bounds is presented. The general formalism can be applied to every type of experiment that was studied so far. Second, more robust upper bounds are proved for the threshold experiments, using a new technique based on a different choice of advice. The new technique generalizes for two-sided experiment, for which the former techniques were only applicable for polynomial cost experiments (such as the Sharp Scatter Machine) and for a very specific case of exponential cost experiments (seen in [BCT12a]). Third, the case of vanishing experiments is analised and the respective protocols are defined with rigour, with an additional degree of (im)precision, the time tolerance. For the case of upper bounds yet another technique is used that differs from the previous types.

2 Defining analog-digital machines

Our main object of study will be what we can define as an *analog-digital Turing machine*, or AD machine. According to [BCT12a], there are three important components of an analog-digital Turing machine:

analog-digital Turing machine = Turing machine + physical experiment + interface.

The Turing machines that we consider are equipped with one *input tape* for input, several *work tapes* for performing calculations, and for the case of AD machines, one *query tape* for the purpose of instructing the physical experiment, and a finite amount of states to interact with the physical experiment, such as the *query state* and an outcome state for each possible outcome of the experiment.

2.1 Physical experiments

We take a look at a lot of different physical experiments. There are many features that they have in common, namely: some initial conditions that can be tuned to some specific values; a physical process, depending on the initial conditions, which takes a (possibly infinite) amount of time; and a finite set of possible results or outcomes. We characterize experiments as two-sided, threshold or vanishing, depending on the type of comparisons that can be made.

In two-sided experiments we can compare two values z and a and test the inequalities $z < a$ and $a < z$. The canonical example is the balance scale experiment, abbreviated by BSE, mentioned for the first time in [BCT09]. It consists of a balance scale with two pans (see Figure 1). In the right pan we have some body with an unknown mass a.

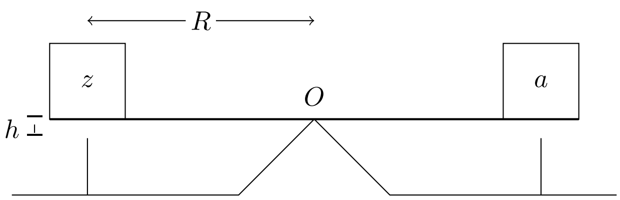

Figure 1: Schematic depiction of the balance scale experiment.

To measure a we place test masses z on the left pan of the balance: if $z < a$, then the right pan of the scale will move down; if $z > a$, then the left pan of the scale will move down; if $z = a$, then we assume that the scale will not move since it is in equilibrium.

In threshold experiments we can only test one of the inequalities regarding the comparison between z and a; for example, we could only test if $a < z$. As a canonical example we introduce the broken balance experiment[1], mentioned for the first time in [BCT09] (abbreviated by BBE). The difference from the two-sided variant is that we place a rigid block under the right pan of the balance (see Figure 2).

To measure a we place test masses z on the left pan of the balance: if $z < a$, then the scale will not move since the rigid block prevents the right pan from moving down; if $z > a$, then the left pan of the scale will move down, which will be detected in some way; if $z = a$, then we assume that the scale will not move since it is in equilibrium.

[1] suggested by Prof. Manuel João Morais.

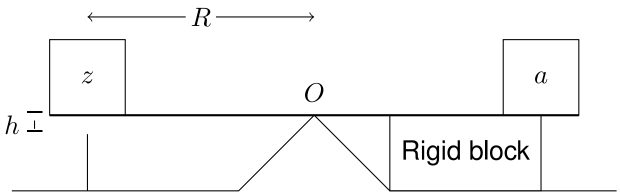

Figure 2: Schematic depiction of the broken balance experiment.

In vanishing experiments we can only test, for masses z and a, if $z \neq a$. The canonical example in this case is the vanishing balance experiment, abbreviated by VBE (Figure 3).

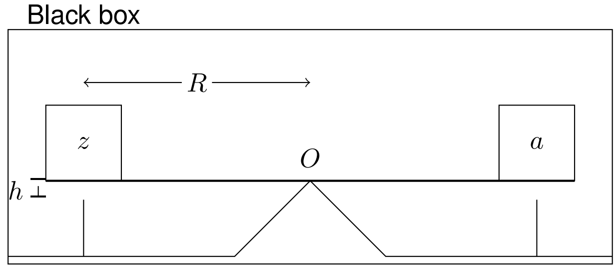

Figure 3: Schematic depiction of the vanishing balance experiment

The main difference from the two-sided balance is that both pressure sticks are connected to the same instrument, so we only get one possible signal, and that the entire experiment is contained inside an opaque box. The behaviour is also different: (a) if $z = a$, then the scale will not move since the system is in equilibrium; (b) if $z \neq a$, then one of the pans will move down, which will be detected by one of the pressure sticks. However, there is *no information about which of the pans sank*, only that one of them did.

2.2 Cost and time schedule

Our next step is to compute the cost (experimental time) of the experiments. 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 a, we can compute the cost of the experiment, which is the time taken for the left pan of the balance to touch the pressure stick, denoted by $T_{exp}(z, a)$:

$$T_{exp}(z,a) = \tau\sqrt{\frac{z+a}{|z-a|}} \text{ in the two-sided and vanishing versions, if } z \neq a;$$

$$T_{exp}(z,a) = \tau\sqrt{\frac{z+a}{z-a}} \text{ in the threshold version, if } z > a;$$

$$T_{exp}(z,a) = \infty \text{ in the threshold version, if } z \leq a.$$

Since the physical experiment has a consultation cases, it makes sense to implement a feature for interrupting a physical experiment that has been going on for too much time. For that end we introduce the concept of a time schedule (mentioned in [BCT10a]), that specifies the time allowed for the experiment. For us, a time schedule is simply a time-constructible function $T : \mathbb{N} \to \mathbb{N}$.

With the notion of time schedule we are able to devise protocols, which are basically a sequence of instructions pertinent to the experiment. For example, one may define the sequence of instructions for a given instance of the broken balance experiment, as in Figure 4.

PROTOCOL "MASS(IP)"

1. **Input** dyadic rational z (possibly padded with 0s);
2. **Place** a mass z in the left pan;
3. **Wait** $T(|z|)$ units of time;
4. **If** the pressure stick sent a signal, **Then Return** 'YES', **Else Return** 'TIMEOUT'.

Figure 4: Procedure that describes the BBE with infinite precision, for some unknown mass a and some time schedule T.

2.3 Precision and decidability

We can see that a query word corresponds to a dyadic rational in $[0, 1]$ that represents a possible initial condition of the experiment. Thus all protocols will attempt to perform the experiment with test position z, where z is the dyadic rational corresponding to the query word. However, we may allow some imprecision in the initial conditions; for instance, we could attempt to perform the experiment with initial condition z but a different z' is used instead. This is our way to incorporate imprecision in the system. As such, there are three types of precision that we will study in these experiments:

- *infinite precision*: when z is read in the query tape, we place a test mass z in the left pan;
- *unbounded precision*: when z is read in the query tape, we place a test mass z' in the left pan, where z' is randomly and uniformly sampled in $\left(z - 2^{-|z|}, z + 2^{-|z|}\right)$;
- *fixed precision* ϵ, for some real valued $\epsilon > 0$: when z is read in the query tape, we place a test mass z' in the left pan, where z' is randomly and uniformly sampled in $(z - \epsilon, z + \epsilon)$;

With all the above concepts, we can define the notion of analog-digital Turing machine, or AD machine, as well as the notion of decidability by an AD machine. An AD machine is specified by an oracle Turing machine M, a class of experiments E (in this case, one of the three balances), a time schedule T and a type of precision P (either infinite or unbounded or fixed). We denote the corresponding AD machine by $M_{E,T,P}$. Thus, for example, $M_{BBE(a),T,IP}$ could denote an AD machine coupled with the broken balance experiment with unknown mass a and infinite precision.

Definition 1. We say that a set A is decidable by an AD machine in polynomial time if there is an oracle Turing machine M, an unknown mass a, a time schedule T and some $0 < \gamma < 1/2$ such that, for any input word w, if $w \in A$, then M accepts w with probability at least $1 - \gamma$, and if $w \notin A$, then M rejects w with probability at least $1 - \gamma$.

3 Techniques

The main goal of the thesis is to characterize the complexity classes decidable in polynomial time by such AD machines. To that end, we prove lower bounds (finding a class $\mathcal{C}$ such that every set in $\mathcal{C}$ is decidable in polynomial time by an AD machine) and upper bounds (finding a class $\mathcal{C}$ such that every set decidable by an AD machine in polynomial time is in $\mathcal{C}$). We now present several techniques that we found useful in proving such results.

Measurement algorithms: In proving lower bounds, we try to encode some advice as a real number (for the unknown mass) and then use the AD machine to obtain aproximations of that number. That is, we specify measurement algorithms such as the one in Figure 3. In what follows the suffix operation $\downharpoonright_n$ on a word w, $w\downharpoonright_n$, denotes the prefix sized n of the ω-word $w0^\omega$, no matter the size of w.

ALGORITHM "BINARYSEARCH(IP)"

1. **Input** a natural number ℓ — number of places to the right of the left leading 0;
2. $x_0 := 0; m := 0, x_1 := 1;$
3. **While** $x_1 - x_0 > 2^{-\ell}$ **Do Begin**
   3.1. $m := \dfrac{x_0 + x_1}{2};$
   3.2. $s :=$ Mass$(IP)\,(m\lfloor_\ell);$ `%Recall that this step takes` $T(\ell)$ `units of time`
   3.3. **If** $s =$ 'YES', **Then** $x_1 = m$, **Else** $x_0 = m$;
4. **End While**;
5. **Output** the dyadic rational denoted by x_0.

Figure 5: Procedure to obtain an approximation of an unknown mass a in the right pan of the Broken Balance. As we see, the protocol is called in instruction 3.2 with test mass $m\lfloor_\ell$.

Biased coin toss: In the cases of fixed precision, we use estimations to produce approximations of increasing precision. We can place the unknown mass in such a way that the oracle call Mass$(FP, \epsilon)(1\lfloor_\ell)$, i.e., the call with test mass $z = 0.5$ on the left pan and waiting $T(\ell)$ units of time, works as a biased coin whose probability of success converges to a real number s as ℓ grows to infinity. Thus we can guess an approximation for s by performing Mass$(FP, \epsilon)(1\lfloor_\ell)$ a large number of times and returning the relative frequency of those that produced result 'YES'.

ALGORITHM "FREQCOUNT(FP, ϵ, h)"

1. **Input** a natural number ℓ — number of places to the right of the left leading 0;
2. $c := 0; \zeta := 2^{2\ell+h};$
3. **Repeat** ζ times
   3.1. $s :=$ Mass$(FP, \epsilon)\,(1\lfloor_\ell);$ `%Recall that this step takes` $T(\ell)$ `units of time`
   3.2. **If** $s =$ 'YES', **Then** $c := c + 1;$
4. **End Repeat**;
5. **Output** the dyadic rational denoted by c/ζ.

Figure 6: Procedure to obtain an approximation of a real number s; assume fixed precision ϵ and unknown mass $a = 1/2 - \epsilon + 2s\epsilon$; h is an integer number used to bound the probability of error.

Fair coin tosses: Finally, in the cases of unbounded and fixed precision, the experiment becomes probabilistic and we can use it to simulate independent coin tosses and to produce random strings.

Lemma 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 $1 - \delta$.*

The previous and the next lemmas are a variation of those relative to the two-sided oracles (see [von56, BCLT08]). After fixing the dyadic rational provided by Lemma 1 relative to some unknown mass a, 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'. We then have:

Lemma 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.*

Boundary and section numbers: For proving upper bounds, we desire to simulate the execution of an AD machine using a (probabilistic) Turing machine that uses some advice. For the advice we introduce a sequence of numbers, called the boundary numers (or section numbers in the case of vanishing experiments) that are defined in terms of the experimental time $T_{exp}(z, a)$ and the time schedule T.

Definition 2. Let $a \in (0, 1)$ be the unknown mass and T a time schedule. Then, for all $k \in \mathbb{N}$, we define l_k and r_k as the real numbers in $(0, 1)$ such that $T_{exp}(l_k, y) = t_{exp}(r_k, a) = T(k)$ and $l_k < a < r_k$.

Definition 3. Let $a \in (0, 1)$ be the unknown mass. Then, for every natural number k, we define the section numbers l'_k and r'_k as the real numbers in $(0, 1)$ such that $T_{exp}(l'_k, a) = T_{exp}(r'_k, a) = k$ and $l'_k < a < r'_k$.

The following argument shows the utility of these numbers. Consider the two-sided experiment. For any oracle query z of size k, (a) if $z < l_k\downharpoonleft_k$,[2] then the result of the experiment is 'LEFT'; (b) if $z > r_k\downharpoonleft_k$, then the result of the experiment is 'RIGHT' and (c) if $l_k\downharpoonleft_k \leq z \leq r_k\downharpoonleft_k$, then the result of the experiment is 'TIMEOUT'. Thus, by knowing approximations of l_k and r_k, we can obtain the result of any experiment of size k (in the infinite precision case) without having to perform it. A similar reasoning can be made for the threshold and vanishing experiments.

Simulation algorithms: We have seen that we can simulate a call to the oracle without using it. This means, in other words, that it is possible to specify simulation algorithms (using advice) and use them to replace oracle queries. One of these algorithms is presented in Figure 3; note however that this is for a very simple case, and for other choices of experiment and precision the simulation algorithm can be quite complicated.

ALGORITHM "SIMULATE(IP)"

1. **Input** z — dyadic rational of size k;
2. **Advice** r'_k — approximation of r_k;
3. **If** $z \geq r'_k$, **Then Return** 'YES';
4. **If** $z < r'_k$, **Then Return** 'TIMEOUT';

Figure 7: Simulation for the two-sided experiment with infinite concept precision.

4 Summary of results

We have studied several types of oracles and characterized several classes of complexity. We begin by emphasizing that the role of the physical experiment is only motivational. Most of the results are proven with respect to a particular physical experiment (the BSE in the two-sided case; the BBE in the threshold case; the VBE in the vanishing case). However, *most of these results can be equally stated and proven assuming that the computation is carried with an oracle* $\mathcal{O}$ such that: $\mathcal{O}$ receives a dyadic rational or a pair of dyadic rationals and returns one of a finite number of results; $\mathcal{O}$ may be deterministic or stochastic; $\mathcal{O}$ has a cost of consultation. Each oracle type has its specification, that is,

- two-sided oracles are of the form $\mathcal{O}_y(z) =$'LEFT' if $z < y$ and $\mathcal{O}_y(z) =$'RIGHT' if $z > y$;
- threshold oracles are of the form $\mathcal{O}_y(z) =$'YES' if $z > y$;
- vanishing oracles are of the form $\mathcal{O}_y(z) =$'YES' if $z \neq y$,

[2]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.

where y is an unknown value. The determinism or stochasticity of the oracle is given by the notion of precision considered and, in the vanishing case, also by the chosen time tolerance. The consultation cost is given by the experimental time function of the chosen experiment. Our results are summarized in the following table:

Type of Oracle		Infinite	Unbounded	Finite
Two-sided	lower bound	$P/log\star$	$BPP//log\star$	$BPP//log\star$
	upper bound	$P/poly$	$P/poly$	$P/poly$
	upper bound (w/ exponential T)	$--$	$--$	$--$
Threshold	lower bound	$P/log\star$	$BPP//log\star$	$BPP//log\star$
	upper bound	$--$	$--$	$--$
	upper bound (w/ exponential T)	$P/log\star$	$BPP//log\star$	$BPP//log\star$
Vanishing1	lower bound	$P/poly$	$P/poly$	$BPP//log\star$
	upper bound	$P/poly$	$P/poly$	$BPP//log\star$
	upper bound (w/ exponential T)	$--$	$--$	$--$
Vanishing2	lower bound	$P/log\star$	$BPP//log\star$	$BPP//log\star$
	upper bound	$P/poly$	$P/poly$	$BPP//log\star$
	upper bound (w/ exponential T)	$--$	$BPP//log\star$	$--$

Naturally, each experiment considered has an associated physical time; we now provide a list of all assumptions on the computational cost to prove lower bounds:

- t_{exp} is bounded by $C/|z-y|^d$ for some constants C and d;
- t_{exp} increases as $|z-y|$ decreases;
- t_{exp} is differentiable and t'_{exp} fits between $C/|z-y|^d$ and $D/|z-y|^d$ for some constants C, D and d.

And here is a list of all assumptions on the computational cost to prove upper bounds:

- t_{exp} increases as $|z-y|$ decreases;
- there is a procedure to simulate in polynomial time queries of size k using $\mathcal{O}(k)$ bits of advice;
- there is a procedure to compute in polynomial time the first k bits of any boundary or section number using $\mathcal{O}(k)$ bits of advice.

It can be seen that the previous assumptions are all satisfied for many common choices of experimental time functions, and we highlight two: functions of the form $t_{exp}(z) = C/|z-a|^d$, and $t_{exp}(z) = \tau\frac{(z+a)^b}{|z-a|^d}$. Additional assumptions are used in some proofs, namely that T is exponential, that $g \in o\left(\lambda n.2^{n/2}\right)$ and that $\lambda z.g(|z|) \in PF$.

We now observe that the bounds for the two-sided and threshold oracles are essentially the same. In fact, we can state that the power of Turing machines, when coupled with either the two-sided or threshold oracle, is boosted to a class between $P/log\star$ ($BPP//log\star$ using non-infinite precision) and $P/poly$; the class is exactly $P/log\star$ ($BPP//log\star$ using non-infinite precision) if we further assume that the time schedule is exponential. This bound is weaker than other bounds presented in literature: for example, [SS95] studied an analog model that boosted the computational power (using polynomial resources) to $P/poly$. There is a reason for the difference in the classes obtained: in the neural networks, it is possible to extract a polynomial amount of information (that is, the bits of the real weights in the network) in polynomial time; however, in our model, since the experimental time functions are exponential, it seems that we can only extract a logarithmic amount of information in polynomial time. There is evidence to state that exponential cost is the cause for the upper bounds of logarithmic advice. However, in all of the physical experiments that we considered the experimental time function was seen to increase exponentially as the test value approaches the unknown value. We are led to question if this is common to all physical experiments; that is, we can formulate the following conjecture, also seen in [BCT10b]:

Conjecture. All experiments of measurement in nature have an exponential cost of measurement. That is, in polynomial time it is only possible to extract a logarithmic amount of information from a physical experiment. That is, the class of sets decided in polynomial time by AD machines using oracles arising from physical experiments is contained in $BPP//log\star$.

However, when we consider vanishing oracles we come across evidence that appears to contradict the previous conjecture. Using oracles in the first implementation, the computational power is boosted to $P/poly$, and we can make the lower and upper bounds coincide. This means that vanishing oracles with the first implementation have the same power as neural networks, [Sie99]. But we must not forget that the first implementation requires a very strong assumption, namely that, given two events, we can decide which of these occurs first, no matter how small is the difference in times. The reader can consult the full thesis for details on these implementations.

There is also a remarkable conclusion that we must not forget to mention. When giving examples of experiments at the beginning of each chapter we have observed that the balance scale can be adapted to each class of oracles. That is, the balance scale can be seen as the canonical model of physical experiment: the original balance scale is a two-sided oracle; replacing a pressure stick by a rigid block produces a threshold oracle; and placing the entire balance inside a 'black box' gives rise to a vanishing oracle. Thus the balance scale is a very positive experimental setting that unifies all classes of oracles.

To conclude, we adress some open problems that are left to future research. In the two-sided and threshold oracles, to work on the gap between the lower bounds and the upper bounds without assumptions on T. That is, to show that either the upper bound falls to $P/log\star$ ($BPP//log\star$ in non-infinite precision) or that there is a set decided in polynomial time by an AD machine that is not in $BPP//log\star$. We believe that this can be done for the fixed precision with some techniques used for the vanishing oracles. In the two-sided and threshold oracles, to see what happens for different choices of cost functions. We have only proven upper bounds when t_{exp} has a well-known expression. Could we prove the same upper bounds knowing only, for example, that t_{exp} fits between two exponentials? In the second implementation of the vanishing oracle, in addition to the same problems as before, to state and prove the upper bound for infinite precision and exponential time schedule, for which the techniques introduced were not capable of producing a desirable result. We hope that in tackling these problems new insights can be acquried about the influence of oracles in the computational power of AD machines.

References

[BC96] Olivier Bournez and Michel Cosnard. On the computational power of dynamical systems and hybrid systems. *Theoretical Computer Science*, 168(2):417–459, 1996.

[BCLT08] 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.

[BCLT09] 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.

[BCT09] Edwin Beggs, José Félix Costa, and John V. Tucker. Physical oracles, 2009. Technical Report, Department of Computer Science, University of Swansea.

[BCT10a] 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. Special issue on Quantum Algorithms, Editor Salvador Elías Venegas-Andraca.

[BCT10b] 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. Special issue on Contributions of Logic to the Foundations of Physics, Editors D. Aerts, S. Smets & J. P. Van Bendegem.

[BCT10c] Edwin Beggs, José Félix Costa, and John V. Tucker. The Turing machine and the uncertainty in the measurement process. In Hélia Guerra, editor, *Physics and Computation, P&C 2010*, pages 62–72. CMATI – Centre for Applied Mathematics and Information Technology, University of Azores, 2010.

[BCT12a] Edwin Beggs, José Félix Costa, and John V. Tucker. 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.

[BCT12b] 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. Special issue on Computability of the Physical, Editors Cristian S. Calude and S. Barry Cooper.

[Bla07] Ed Blakey. An analogue solution to the problem of factorization. Technical Report RR-07-04, Oxford University Computing Laboratory, July 2007.

[BT07] Edwin Beggs and John V. Tucker. Experimental computation of real numbers by Newtonian machines. *Proceedings of the Royal Society, Series A (Mathematical, Physical and Engineering Sciences)*, 463(2082):1541–1561, 2007.

[GH86] Robert Geroch and James B. Hartle. Computability and physical theories. *Foundations of Physics*, 16(6):533–550, 1986.

[KCG94] Pascal Koiran, Michel Cosnard, and Max Garzon. Computability with low-dimensional dynamical systems. *Theoretical Computer Science*, 132:113–128, 1994.

[Sie95] Hava T. Siegelmann. Computation beyond the Turing limit. *Science*, pages 545–548, 1995.

[Sie99] Hava T. Siegelmann. *Neural Networks and Analog Computation: Beyond the Turing Limit*. Birkhäuser, 1999.

[SS95] Hava T. Siegelmann and Eduardo D. Sontag. On the computational power of neural networks. *J. Comp. Syst. Sciences*, 50(1):132–150, 1995.

[von56] John von Neumann. Probabilistic logics and the synthesis of reliable organisms from unreliable components. In *Automata Studies*, pages 43–98. Princeton University Press, 1956.

[WN05] Damien Woods and Thomas J. Naughton. An optical model of computation. *Theoretical Computer Science*, 334(1-3):227–258, 2005.